

# Пространственные облачные вычисления

Практический подход



# Пространственные облачные вычисления

## Практический подход

Чаовой Ян

Цюньин Хуан

В сотрудничестве с

Чжэньлун Ли

Чэнь Сюй

Кай Лю



CRC Press

Taylor & Francis Group  
Бока Ратон Лондон Нью-Йорк

---

CRC Press - это импринг информационной  
компания Taylor & Francis Group.

CRC Press  
Taylor & Francis Group  
6000 Broken Sound Parkway NW, Номер 300  
Бока-Ратон, FL 33487-2742

Впервые издано в мягкой обложке 2017

© 2014 Taylor & Francis Group, LLC  
CRC Press - это импринт информационной компании Taylor & Francis Group

Претензии к оригинальным работам правительства США отсутствуют

Дата версии: 20131227

ISBN 13: 978-1-138-07555-9 (pbk)

ISBN 13: 978-1-4665-9316-9 (hbk)

Данная книга содержит информацию, полученную из достоверных и авторитетных источников. Были приложены разумные усилия для публикации достоверных данных и информации, однако автор и издатель не несут ответственность за достоверность всех материалов или последствия их использования. Авторы и издатели предприняли попытку определить владельцев авторских прав на все материалы, воспроизведенные в данной публикации, и принести извинения владельцам авторских прав, если разрешение на публикацию в надлежащей форме не было получено. Если какой-либо материал, защищенный авторским правом, не был признан, пожалуйста, напишите нам и дайте знать, чтобы мы могли исправить это при дальнейшем издании книги.

За исключением случаев, разрешенных Законом США об авторском праве, никакая часть настоящей книги не может быть перепечатана, воспроизведена, передана или использована в любой форме любыми электронными, печатными или другими средствами, известными в настоящее время или изобретенными в будущем, включая фотокопирование, микрофильмирование и запись, или в любой системе хранения или поиска информации, без письменного разрешения издателей.

Для получения разрешения на фотокопирование или использование материалов в электронном виде, пожалуйста, перейдите по адресу [www.copyright.com](http://www.copyright.com) (<http://www.copyright.com/>) или свяжитесь с Центром по оформлению авторских прав, Inc. (CCC), 222 Rosewood Drive, Danvers, MA 01923, 978-750-8400. CCC - некоммерческая организация, предоставляющая лицензии и регистрацию для различных пользователей. Для организаций, которым CCC предоставила лицензию на фотокопирование, установлена отдельная система оплаты.

**Обозначение товарного знака на издании:** Продукт или фирменные наименования могут быть торговыми марками или зарегистрированными торговыми марками и используются только для идентификации и объяснения без намерения нарушения таких прав.

---

### Библиографическая запись Библиотеки Конгресса США

---

Ян, Чаовей.

Пространственная облачная обработка данных: практический подход / авторы, Чаовей Ян, Цюньинь Хуанг.  
страницы см

Краткое изложение: "Облачная обработка данных - это набор сетевых услуг, обеспечивающих масштабируемые, недорогие вычислительные платформы по требованию, доступ к которым можно получить простым способом. Данная книга помогает читателям понять процесс развертывания и настройки геопространственных приложений в облаке, а также то, как оптимизировать облако, чтобы они лучше поддерживали геопространственные приложения. Здесь также обсуждаются и представлены стратегии настройки различных типов приложений для более эффективного использования облачных возможностей, таких как обслуживание в режиме "вопрос-ответ" - Предоставлено издателем.

Содержит библиографические ссылки и указатель.

ISBN 978-1-4665-9316-9 (твёрдый переплёт)

1. Геопространственные данные. 2. Облачные вычисления. И. Хуанг, Цунинг. II. Название.

G70.217.G46Y36 2014

004.67'82--dc23

2013037852

---

См. веб-сайт Taylor & Francis  
<http://www.taylorandfrancis.com>  
и веб-сайт CRC Press <http://www.crcpress.com>

---

# Содержание

---

<i>Введение</i>	xvii
<i>Благодарность</i>	xxiii

## **ЧАСТЬ I**

<b>Введение в облачную обработку данных для геонаук</b>	<b>1</b>
---	----------

<b>1 Проблемы применения геонаучных приложений для вычислительной инфраструктуры</b>	<b>3</b>
--	----------

1.1 Проблемы и возможности приложений геонаук в 21 веке	3
---	---

1.1.1 Энергия	3
---------------	---

1.1.2 Реагирования на чрезвычайные ситуации	4
---	---

1.1.3 Изменение климата	5
-------------------------	---

1.1.4 Устойчивое развитие	6
---------------------------	---

1.2 Потребности новой вычислительной инфраструктуры	8
---	---

1.2.1 Обеспечение достаточной вычислительной мощности	8
---	---

1.2.2 Реагирование в реальном времени	9
---------------------------------------	---

1.2.3 Энергосбережение	9
------------------------	---

1.2.4 Экономия бюджета	10
------------------------	----

1.2.5 Улучшение доступности	10
-----------------------------	----

1.3 Зарождение облачной обработки данных	10
--	----

1.3.1 Распределенные вычисления	10
---------------------------------	----

1.3.2 Сервисы по запросу	11
--------------------------	----

1.3.3 Совместное использование вычислений и экономия средств	12
--	----

1.3.4 Надежность	12
------------------	----

1.3.5 Появление облачной обработки данных	12
---	----

1.4 Преимущества и недостатки облачной обработки данных для геонаучных приложений	14
---	----

vi Содержание

---

- 1.4.1 *Преимущества облачной обработки данных*
- 14 1.4.2 *Проблемы* 14
- 1.5 *Краткое изложение* 15
- 1.6 *Проблемы* 15
- Список литературы* 16

## **2 Архитектура, концепции и характеристики облачной обработки данных** 19

- 2.1 *Концепты* 19
- 2.2 *Архитектура облачной обработки данных* 20
- 2.3 *Характеристики* 22
- 2.4 *Модели сервисов* 23
- 2.5 *Развертывание модели и типы облаков* 25
- 2.6 *Обзор сервисов облачной обработки данных* 27
  - 2.6.1 *Коммерческие облака* 27
  - 2.6.2 *Облачные решения с открытым исходным кодом* 29
- 2.7 *Краткое изложение* 30
- 2.8 *Проблемы* 30
- Список литературы* 31

## **3 Технологии реализации**

33

- 3.1 *Аппаратные достижения* 33
  - 3.1.1 *Многоядерные технологии* 33
  - 3.1.2 *Сеть* 34
  - 3.1.3 *Хранение* 34
  - 3.1.4 *Умные устройства* 35
- 3.2 *Вычислительные технологии* 35
  - 3.2.1 *Парадигма распределенных вычислений* 35
  - 3.2.2 *Модель вычислительной архитектуры* 36
- 3.3 *Виртуализация* 36
  - 3.3.1 *Реализация виртуализации* 37
  - 3.3.2 *Решения виртуализации* 38
- 3.4 *Распределенная файловая система* 39
  - 3.4.1 *Введение в распределенную файловую систему* 40
  - 3.4.2 *Файловая система Google* 40
  - 3.4.3 *Распределенная файловая система Apache Hadoop* 41
- 3.5 *Web x.0* 42
  - 3.5.1 *Веб-сервисы* 43
  - 3.5.2 *Сервис-Ориентированная Архитектура* 44
- 3.6 *Заключение* 46

- 3.7 *Краткое изложение* 46
- 3.8 *Проблемы* 47
- Список литературы* 47

## **ЧАСТЬ II**

### **Развертывание приложений и облачных сервисов** 49

#### **4 Как применять облачную обработку данных** 51

- 4.1 *Популярные облачные сервисы* 51
  - 4.1.1 *Введение* 51
  - 4.1.2 *Amazon AWS и Windows Azure* 52
- 4.2 *Способ внедрения: Простое веб-приложение* 53
  - 4.2.1 *HTML-дизайн для веб-приложения HelloCloud* 53
  - 4.2.2 *Веб-серверы* 54
- 4.3 *Развертывание веб-приложения в облачных сервисах* 55
  - 4.3.1 *Amazon Web Services* 55
  - 4.3.2 *Windows Azure* 66
- 4.4 *Заключение и обсуждение* 70
- 4.5 *Краткое изложение* 72
- 4.6 *Проблемы* 72
- Список литературы* 73

#### **5 Облачные приложения для геонаук** 75

- 5.1 *Общие компоненты для приложений геонаук* 75
  - 5.1.1 *Серверное программирование* 75
  - 5.1.2 *База данных* 76
  - 5.1.3 *Высокопроизводительные вычисления* 76
- 5.2 *Приложения для геофизических исследований в облаке* 77
- 5.3 *Способы внедрения* 78
  - 5.3.1 *Веб-приложения, управляемые базами данных* 78
  - 5.3.2 *Типичные приложения для высокопроизводительных вычислений* 84
- 5.4 *Краткое изложение* 90
- 5.5 *Проблемы* 90
- Список литературы* 90

#### **6 Как выбрать облачный сервис: Относительно модели стоимости облачного сервиса** 93

- 6.1 *Важность и проблемы выбора облачных сервисов* 93
- 6.2 *Факторы, влияющие на выбор облачной службы* 94

viii Содержание

---

6.2.1	Предоставление и измерение емкости облачных услуг	94
6.2.2	Правила ценообразования облачной платформы	96
6.2.3	Особенности и требования приложения	97
6.3	Выбор облачных сервисов с использованием Консультативного инструмента по внедрению облака Earth Science Information Partners (ESIP) в качестве примера	98
6.3.1	Архитектура консультативного инструмента	99
6.3.2	Общий рабочий процесс выбора облачного сервиса	99
6.3.3	Способ внедрения	102
6.4	Подробное рассмотрение выбора облачных сервисов и разработки консультативных инструментов	104
6.4.1	Правильность и точность оценочных моделей	105
6.4.2	Актуальная информация об облачных сервисах	106
6.4.3	Интерактивные функции и функции визуализации консультативного инструмента	106
6.5	Краткое изложение	106
6.6	Проблемы	107
	Список литературы	107

### **ЧАСТЬ III**

## **Облачные проекты по геонауке 109**

### **7 ArcGIS в облаке 111**

7.1	Введение	111
7.1.1	Зачем географической информационной системе облако	111
7.1.2	Примеры GIS, которым нужно облако	112
7.2	ArcGIS в облаке	112
7.2.1	ArcGIS Online	112
	❗ Функциональные возможности	113
7.2.2	ArcGIS для сервера	114
	❗ Функциональные возможности	114
7.2.3	ПО GIS как сервис	115
	❗ Функциональные возможности	115
7.2.4	Мобильный GIS сервис	116
7.2.5	Краткое изложение раздела	116
7.3	Способы внедрения	117
7.3.1	Региональный анализ штата Орегон с использованием ArcGIS Online	117

- 7.3.2 Способы внедрения ArcGIS для сервера 120
  - ☒ Общий рабочий рисунок 120 для наводнения городского совета Брисбена
  - ☒ Программа просмотра парков штата Пенсильвания 121
- 7.3.3 Краткое изложение раздела 122
- 7.4 Краткое изложение 122
- 7.5 Проблемы 123
- Список литературы 124

## **8 Информационный сервис GEOSS с поддержкой облака 125**

- 8.1 Информационный сервис GEOSS: исходные данные и сложности 125
  - 8.1.1 Исходные данные 125
  - 8.1.2 Сложности 126
- 8.2 Развертывание и оптимизация 127
  - 8.2.1 Основная последовательность развертывания 127
  - 8.2.2 Особые обсуждения 130
    - ☒ Резервное копирование данных 130
    - ☒ Балансировка нагрузки 132
    - ☒ Автоматическое масштабирование 132
  - 8.2.3 Отличия от общих шагов в главе 5 134
- 8.3 Демонстрация системы 135
  - 8.3.1 Локальный поиск 135
  - 8.3.2 Удаленный поиск 136
- 8.4 Заключение 136
  - 8.4.1 Экономические преимущества 137
  - 8.4.2 Технические преимущества 138
- 8.5 Краткое изложение 138
- 8.6 Проблемы 138
- Приложение 8.1 Шаблон для создания функции автомасштабирования 138
- Список литературы 141

## **9 Climate @ Home с облачной поддержкой 143**

- 9.1 Climate @ Home: исходные данные и сложности 143
  - 9.1.1 Исходные данные 143
  - 9.1.2 Сложности 146
- 9.2 Развертывание и оптимизация 147
  - 9.2.1 Основная последовательность развертывания 147
    - ☒ Развертывание пространственного веб-портала 147

## х Содержание

---

9.2	<i>Развертывание сервераBOINC</i>	149
9.2.2	<i>Особые обсуждения</i>	151
9.2.3	<i>Отличия от общих шагов в главе 5</i>	152
9.3	<i>Демонстрация системы</i>	153
9.3.1	<i>Обзор пространственного веб-портала</i>	153
9.3.2	<i>Гео визуальный аналитический портлет</i>	154
9.3.3	<i>Портлет управления ресурсами</i>	155
9.4	<i>Заключение</i>	155
9.5	<i>Краткое изложение</i>	157
9.6	<i>Проблемы</i>	158
	<i>Приложение 9.1 E4M20a_000040_wu.xml</i>	158
	<i>Приложение 9.2 Climateathome_re.xml</i>	159
	<i>Список литературы</i>	159
<b>10</b>	<b>Прогнозирование пыльных бурь с облачной поддержкой</b>	<b>161</b>
10.1	<i>Моделирование пыльной бури: исходные данные и сложности</i>	161
10.1.1	<i>Исходные данные</i>	161
10.1.2	<i>Сложности</i>	162
10.2	<i>Развертывание и оптимизация</i>	164
10.2.1	<i>Основная последовательность выполнения</i>	164
10.2.2	<i>Особые обсуждения</i>	167
10.2.3	<i>Краткое изложение разницы в основных шагах с главой 5</i>	169
10.3	<i>Демонстрация</i>	169
10.3.1	<i>Пыльная буря в Финиксе</i>	169
10.3.2	<i>Результат моделирования</i>	170
10.3.3	<i>Производительность</i>	170
10.3.4	<i>Эффективность затрат</i>	173
10.4	<i>Заключение</i>	174
10.5	<i>Краткое изложение</i>	175
10.6	<i>Проблемы</i>	175
	<i>Список литературы</i>	175
<b>ЧАСТЬ IV</b>		
	<b>Статус и готовность облачной обработки данных</b>	<b>177</b>
<b>11</b>	<b>Облачные сервисы</b>	<b>179</b>
11.1	<i>Введение в облачные сервисы</i>	179
11.1.1	<i>Географическое присутствие</i>	179

---

11.1.2	Пользовательские интерфейсы и доступ к серверам	180
11.1.3	Автоматическое масштабирование и балансировка нагрузки	180
11.1.4	Соглашение об уровне обслуживания (SLA)	180
11.2	Amazon Web Services (AWS)	181
11.2.1	Архитектура	181
11.2.2	Основные характеристики EC2	182
11.2.2.1	Масштабируемость	182
11.2.2.2	Совместимость	183
11.2.2.3	Развертывание и интерфейс	183
11.2.2.4	Гипервизоры	183
11.2.2.5	Надежность	183
11.2.2.6	ОС поддержка	184
11.2.2.7	Стоимость	184
11.2.3	Основные пользователи и общие комментарии	185
11.2.3.1	Список основных клиентов	185
11.2.3.2	Завершение	186
11.2.3.3	Отзывы сообщества	186
11.2.4	Сложность использования	187
11.3	Windows Azure	188
11.3.1	Архитектура	188
11.3.2	Общие характеристики Windows Azure	189
11.3.2.1	Масштабируемость	189
11.3.2.2	Совместимость	190
11.3.2.3	Развертывание и интерфейс	191
11.3.2.4	Гипервизоры	191
11.3.2.5	Надежность	191
11.3.2.6	ОС поддержка	191
11.3.2.7	Стоимость	191
11.3.3	Основные пользователи и общие комментарии	192
11.3.3.1	Список основных клиентов	192
11.3.3.2	Завершение	193
11.3.3.3	Отзывы сообщества	193
11.3.3.4	Сложность использования	193
11.4	Nebula	194
11.4.1	Архитектура	194
11.4.2	Основные характеристики Nebula	195
11.4.2.1	Масштабируемость	195
11.4.2.2	Совместимость	195
11.4.2.3	Развертывание и интерфейс	195
11.4.2.4	Гипервизоры	195

## xii Содержание

---

11.4.2.5	Надежность	195
11.4.2.6	Поддержка ОС	195
11.4.2.7	Стоимость	195
11.4.3	Основные пользователи и общие комментарии	196
11.4.3.1	Список основных клиентов	196
11.4.3.2	Завершение	196
11.4.3.3	Отзывы сообщества	196
11.4.3.4	Сложность использования	196
11.5	Заключение	196
11.6	Краткое изложение	197
11.7	Проблемы	198
	Список литературы	198
<b>12</b>	<b>Как проверить готовность облачных сервисов</b>	<b>201</b>
12.1	Введение	201
12.2	Среда тестирования	201
12.2.1	Сеть	201
12.2.2	Конфигурация вычислительного сервиса	202
12.3	Одновременный тест интенсивности с использованием Информационного сервиса GEOSS (CLH)	202
12.3.1	Требования к информационному центру для компьютерных услуг	202
12.3.2	Проектирование тестов	203
12.3.3	Рабочий процесс тестирования	204
12.3.4	Анализ результатов испытаний	206
12.4	Тест данных и вычислений с использованием Climate @ Home	207
12.4.1	Требования к вычислению Climate @ Home	207
12.4.2	Проектирование теста	208
12.4.3	Рабочий процесс тестирования	208
12.4.4	Анализ результатов испытаний	211
12.5	Облачный тест с использованием прогноза пыльной бури	211
12.5.1	Требования к вычислениям для прогнозирования пыльных бурь	211
12.5.2	Проектирование теста	212
12.5.3	Рабочий процесс тестирования	213
12.5.4	Анализ результатов испытаний	215
12.6	Краткое изложение	216
12.7	Проблемы	217
	Приложение 12.1: Пример GetRecords для поиска в метаданных	218
	Приложение 12.2: Пример плана тестирования Jmeter	218
	Список литературы	221

<b>13 Решения для облачной обработки данных с открытым исходным кодом</b>	<b>223</b>
13.1 Введение в решения для облачной обработки данных с открытым исходным кодом	223
13.1.1 <i>CloudStack</i>	224
13.1.2 <i>Eucalyptus</i>	224
13.1.3 <i>Nimbus</i>	225
13.1.4 <i>OpenNebula</i>	225
13.2 <i>CloudStack</i>	225
13.2.1 Архитектура	225
13.2.2 Общие характеристики <i>CloudStack</i>	226
13.2.3 Основные пользователи и общие комментарии к решению <i>CloudStack</i>	227
13.3 <i>Eucalyptus</i>	228
13.3.1 Архитектура	228
13.3.2 Общие характеристики <i>Eucalyptus</i>	229
13.3.3 Основные пользователи и общие комментарии к <i>Eucalyptus</i>	230
13.4 <i>OpenNebula</i>	231
13.4.1 Архитектура	231
13.4.2 Общие характеристики <i>OpenNebula</i>	231
13.4.3 Основные пользователи и общие комментарии к <i>OpenNebula</i>	233
13.5 <i>Nimbus</i>	234
13.5.1 Архитектура	234
13.5.2 Общие характеристики <i>Nimbus</i>	235
13.5.3 Основные пользователи и общие комментарии к <i>Nimbus</i>	235
13.6 Рекомендации по тестированию с открытым исходным кодом	236
13.7 Заключение	236
13.8 Краткое изложение	236
13.9 Проблемы	239
Список литературы	239
<b>14 Как проверить готовность решений облачной обработки данных открытым исходным кодом</b>	<b>241</b>
14.1 Введение	241
14.2 Среда тестирования	241
14.3 Тесты облачных операций	243
14.4 Тесты виртуальных вычислительных ресурсов	244
14.4.1 Краткое введение	244
14.4.2 Проектирование теста	245
14.4.3 Рабочий процесс тестирования	246
14.4.4 Анализ результатов испытаний	249
14.5 Тесты общих приложений	251

#### xiv Содержание

---

- 14.5.1 *Краткое введение в аспекты тестирования* 251
- 14.5.2 *Проектирование теста* 251
- 14.5.3 *Рабочий процесс тестирования* 252
- 14.5.4 *Анализ результатов испытаний* 253
- 14.6 *Проверка готовности GEOSS Clearinghouse к облаку* 254
  - 14.6.1 *Требования к вычислениям в информационном сервисе* 254
  - 14.6.2 *Проектирование теста, рабочий процесс и анализ* 254
- 14.7 *Проверка готовности облаков для прогнозирования пыльных бурь* 255
  - 14.7.1 *Требования к вычислениям для прогнозирования пыльных бурь* 255
  - 14.7.2 *Проектирование теста* 255
  - 14.7.3 *Рабочий процесс тестирования* 256
  - 14.7.4 *Анализ результатов испытаний* 256
- 14.8 *Краткое изложение* 256
- 14.9 *Проблемы* 257
- Приложение 14.1 Патч-файл Ubench (Ubench-Patch.Txt)* 257
- Список литературы* 258

### **15 Инициатива GeoCloud**

261

- 15.1 *Введение* 261
- 15.2 *Архитектура GeoCloud* 262
- 15.3 *Действия в GeoCloud* 264
  - *Создание прототипов платформ* 265
  - *Подтверждение с помощью приложений агентства* 267
  - *Документ и обнародование* 267
- 15.4 *Безопасность GeoCloud* 268
  - *Безопасность Amazon Web Services (AWS)* 268
  - *Операция безопасности GeoCloud* 269
- 15.5 *Операционные расходы в облаке* 269
- 15.6 *Обсуждение* 270
- 15.7 *Краткое изложение* 271
- 15.8 *Проблемы* 271
- Список литературы* 271

### **Часть V**

#### **Будущие направления**

273

### **16 Обработка больших объемов данных, вычислений, одновременного доступа и пространственно-временных шаблонов** 275

- 16.1 *Введение* 275
- 16.2 *Большие данные* 276

<b>И</b>	<i>Введение</i>	276
<b>П</b>	<i>Пример с Climate @ Home</i>	277
<b>В</b>	<i>Решения</i>	277
<b>И</b>	<i>Сохраняющиеся проблемы и будущие исследования</i>	278
<b>16.3</b>	<i>Интенсивность вычислений</i>	<b>280</b>
<b>И</b>	<i>Введение</i>	280
<b>П</b>	<i>Пример с интерполяцией цифровой модели оценки</i>	280
<b>В</b>	<i>Решения</i>	281
	<i>16.3.3.1 Облачные вычисления на ЦП</i>	281
	<i>16.3.3.2 Облачные вычисления на GPU</i>	282
<b>И</b>	<i>Сохраняющиеся проблемы и будущие исследования</i>	283
<b>16.4</b>	<i>Параллельная интенсивность</i>	<b>284</b>
<b>И</b>	<i>Введение</i>	284
<b>П</b>	<i>Пример с Информационным сервисом GEOSS</i>	285
<b>В</b>	<i>Решение</i>	286
	<i>16.4.3.1 Глобальная доставка контента</i>	286
	<i>16.4.3.2 Гибкость</i>	287
	<i>16.4.3.3 Пространственно-временная индексация</i>	287
<b>И</b>	<i>Сохраняющиеся проблемы и будущие исследования</i>	288
<b>16.5</b>	<i>Пространственно-временная интенсивность</i>	<b>289</b>
<b>16.6</b>	<i>Краткое изложение</i>	<b>290</b>
<b>16.7</b>	<i>Проблемы</i>	<b>290</b>
	<i>Список литературы</i>	<b>291</b>

## **17 Исследования облачной обработки данных для геонаук и приложений 295**

### **17.1 Развитие видения 21 века для приложений геонаук 295**

<b>И</b>	<i>Запросы фундаментальной геопространственной науки</i>	295
<b>П</b>	<i>Интеграция геонаук с другими областями науки для новых открытий</i>	296
<b>В</b>	<i>Видение приложения</i>	296
<b>17.2</b>	<i>Технологический прогресс</i>	<b>297</b>
<b>И</b>	<i>Оценка и выбор облаков</i>	297
<b>П</b>	<i>Управление ресурсами облачных сервисов</i>	298
<b>В</b>	<i>Резервное копирование и синхронизация данных</i>	299
<b>И</b>	<i>Функциональная совместимость</i>	299
<b>В</b>	<i>Новые системы визуализации и интерактивные</i>	301
<b>В</b>	<i>Надежность и доступность</i>	302
<b>В</b>	<i>Моделирование в реальном времени и доступ</i>	302

xvi Содержание

---

*17.3 Синергетический прогресс социальных наук и облачной  
обработки данных 303*

- И** *Управление облаком 303*
- И** *Облачный охват 303*
- В** *Безопасность и правила 304*
- И** *Глобальное сотрудничество 305*

*17.4 Краткое изложение 305*

*17.5 Проблемы 306*

*Список литературы 306*

*Индекс*

# Введение

---

## ПОЧЕМУ МЫ РЕШИЛИ НАПИСАТЬ ДАННУЮ КНИГУ?

Есть несколько причин, по которые натолкнули нас на мысль о написании данной книги. Мы начали использовать облачную обработку данных для геонаучных приложений примерно в 2008 году, когда облачная обработка данных только начинала формироваться. В последние годы вышло множество книг касательно облачной обработки данных в области компьютерных наук. Но мы не нашли такой книги, в которой бы подробно описывались различные аспекты того, как геонаучное сообщество может использовать облачную обработку данных. Первая причина для написания данной книги - заполнение такого пробела, в целях оказания помощи геонаучному сообществу охватить различные аспекты того, почему и как использовать облачную обработку данных в геонауках (Части I и II).

Вторая причина - хорошо цитируемая публикация Международного журнала "Цифровая Земля" за 2011 год, посвященная пространственной облачной обработке данных. В докладе были представлены общие концепции и преимущества, которые облачная обработка данных может принести геонаучным исследованиям и прикладным разработкам. Мы также получили вопросы о том, как добиться таких преимуществ и как использовать облачную обработку данных в педагогических целях. Данная книга, с одной стороны, является ответом на просьбы, содержащиеся в Частях II и III о том, как поэтапно применять прикладные геонаучные технологии, основанные на технологиях облачной обработки данных.

Мы провели ряд научно-исследовательских инициатив по использованию облачной обработки данных для применения в геонауках. Проекты, подробно описанные в Частях II, III и IV, варьируются от перемещения веб-портала на облачный сервис до исследования готовности облачной обработки данных для геонаук с использованием как коммерческих облачных сервисов, так и решений с открытым исходным кодом. Мы также сочли, что опыт, как говорится из первых рук, был бы весьма полезен, если бы геологи и разработчики прикладных программ в области геонаук систематически документировали их для оценки, отбора, планирования и осуществления облачных операций для своих прикладных программ. Это была третья причина, которая побудила нас приняться за написание данной книги.

Мы объединили наш опыт, накопленный за последние шесть лет, чтобы написать эту систематически прогрессивную книгу для демонстрации того, как геонаучные сообщества могут использовать облачную обработку данных для перехода от концепций (Часть I),

## xviii Введение

---

перемещения приложений к облачным сервисам (Часть II), приложений для геонаук с поддержкой облачной обработки данных (Часть III), тестов готовности облачных сервисов и федеральных подходов к адаптации облачной обработки данных (Часть IV), а также будущего направления исследований в области облачной обработки данных в геонауке (Часть V). Мы считаем, что данная книга предоставит систематические знания читателям, которые хотят получить представление о пространственной облачной обработке данных, применить облачную обработку данных для своих приложений или провести дальнейшие исследования в данной области.

## КАК ПРОХОДИЛ ПРОЦЕСС НАПИСАНИЯ ДАННОЙ КНИГИ?

В 2012 году издательство CRC Press/Taylor & Francis (Ирма Бриттон) разглядело необходимость в публикации книги по облачной обработке данных для геонаучных сообществ и согласилось с коллективом авторов воплотить данную инициативу в жизнь. За прошедший год мы предприняли 13 важных шагов, чтобы предоставить качественно написанную и структурированную книгу для нашей аудитории: (1) Доктор Чаовэй Ян, Цюньинь Хуан, Чэнь Сюй, а также г-н Чжэньлун Ли и г-н Кай Лю работали над определением структуры и содержания книги с каждым из авторов глав, которые являются разработчиками и исследователями соответствующих проектов. (2) Каждая глава была изначально написана авторами при участии одного или нескольких редакторов. (3) Для того чтобы содержание каждой главы соответствовало общему оформлению книги, Ян отвечал за рецензирование каждой главы в частях I, II и V; Сюй отвечал за рецензирование части IV; Ли отвечал за рецензирование глав 4, 5 и части III. (4) Авторам каждой главы были предоставлены комментарии касательно структуры и содержания, чтобы обеспечить единое оформление. (5) Авторы каждой главы самостоятельно пересмотрели и проанализировали содержащуюся информацию. (6) В целях обеспечения единого связующего смысла каждой главы был проведен внутренний обзор авторами других соответствующих глав. (7) Авторы глав по мере необходимости пересматривали и перестраивали содержащуюся информацию. (8) Каждая глава была разослана на рецензирование 2-4 внешним рецензентам. (9) Авторы каждой главы и редакторы (части) осуществляли совместную работу над комментариями внешних рецензентов. (10) Ян, Сюй и Ли провели заключительный рецензирование и оформление глав. (11) Главы и вся книга были доработаны редакторами Taylor & Francis после завершающего форматирования Наньинь Чжоу. (12) Хуан, Ян, Ли, Сюй и Лю осуществляли совместную работу над созданием онлайн-контента, включая слайды лекций для каждой главы и онлайн-код, скрипты, образы виртуальных машин, видео и документы, чтобы читатели могли с легкостью повторить развертывание облака и процессы перемещения, описанные в данной книге. (13) Онлайн содержание книги опубликовано на веб-сайте Taylor & Francis. Книга написана авторами, которые имеют личный опыт работы, что позволило им более точно составить содержание книги. Содержание книги было также проверено на предмет оформления в виде единого тома, и руководители группы проекта (все редакторы) и главный исследователь (Ян) проверили и одобрили все пункты содержания.

## О ЧЕМ ДАННАЯ КНИГА?

Данная книга всесторонне охватывает знания в области пространственной облачной обработки данных на практических примерах в 17 главах из 5 аспектов, в том числе: (а) Каковы основные концепции облачной обработки данных и зачем геонаукам необходима облачная обработка данных? (b) Каким образом простые прикладные программы в области геонаук могут быть переведены на облачную обработку данных? с) Каким образом сложные прикладные программы в области геонаук могут быть основаны на облачной обработке данных? d) Каким образом можно проверить готовность облачных сервисов к применению прикладных геонаучных технологий? e) Какие вопросы научных исследований требуют дальнейшего изучения?

Первая часть представляет требования геонаук к облачной обработке данных в главе 1, в главе 2 - архитектуру, характеристики и концепции облачной обработки данных, а в главе 3 - технологии, способствующие развитию облачной обработки данных.

В части II излагаются общие процедуры и решения, связанные с переносом геонаучных приложений на облачные сервисы. В четвертой главе рассказывается о том, как использовать облачные сервисы путем развертывания простого веб-приложения на двух популярных облачных сервисах: Amazon EC2 и Windows Azure. В пятой главе рассказывается об общих процедурах развертывания общих геонаучных приложений на облачных платформах, где требуется написание сценариев на стороне сервера, настройка баз данных и высокопроизводительные вычисления. В главе 6 обсуждается выбор облачных сервисов на основе общих критериев оценки облачной обработки данных и моделей стоимости облачной обработки данных.

В части III рассказывается о способах внедрения различных геонаучных приложений в облачных сервисах. В главе 7 рассказывается о том, как пользователи могут взаимодействовать с облачными сервисами на примере ArcGIS. В трех других главах показано, как пользователи могут использовать три различных сложных геонаучных приложения для облачной обработки данных: 1) базы данных, содержащие информацию об облачных сервисах, пространственные индексы и технологии веб-порталов для поддержки Центра обмена информацией GEOSS, 2) моделирование автономных моделей, содержащих информацию об облачных сервисах, и визуализация результатов моделирования для приложения Climate@Home и 3) использование эластичных облачных ресурсов для прогнозирования дестабилизирующих явлений (например, пылевые бури).

В части IV рассматривается готовность облачной обработки данных для поддержания геонаучных приложений с использованием программных решений с открытым исходным кодом для облачной обработки данных и коммерческих облачных сервисов. В главе 11 представлены и сравниваются три коммерческих облачных сервиса: Amazon EC2, Windows Azure и Nebula. В главе 12 проверяется готовность данных облачных сервисов с помощью трех приложений, описанных в части III. В 13-й главе рассказывается о четырех основных облачных вычислительных решениях с открытым исходным кодом, включая CloudStack, Eucalyptus, Nimbus и OpenNebula; их производительность и готовность проверяется и сравнивается в 14-й главе. В главе 15 рассказывается об истории создания, проектировании архитектуры, подходе и координации GeoCloud, представляющей собой межведомственную инициативу по определению общих операционных систем и программных пакетов для приложений в области геонаук.

## xx Введение

---

Наконец, в части V рассматриваются будущие исследования и разработки в области облачной обработки данных в главах 16 и 17. В главе 16 представлены данные, вычисления, параллелизм и пространственно-временная интенсивность геонаук, а также методы использования облачных сервисов для решения таких задач. Глава 17 знакомит с направлениями исследований с точки зрения технологий, видения и социальных аспектов.

## ОНЛАЙН СОДЕРЖАНИЕ КНИГИ

В целях оказания помощи читателям лучше использовать данную книгу с различными дополнениями, см.: <http://www.crcpress.com/product/isbn/9781466593169> размещено следующее содержание в сети: <http://www.crcpress.com/product/isbn/9781466593169>.

- *Информационные слайды для каждой главы* - применяются в образовательных целях, данная книга предоставляет слайды для инструкторов, в целях оказания помощи при изучении содержания книги. Слайды тесно связаны с содержанием глав книги.
- *Ключевые вопросы* - от пяти до десяти вопросов, доступны после каждой главы в целях закрепления изученного материала. Ответы на данные вопросы можно найти в контексте глав в виде обзора основного содержания.
- *Образы виртуальных машин примерами использования*, представленные в данной книге. Главы 4, 5, 7, 8, 9 и 10 включают в себя различные уровни тестирования - от простого веб-приложения до сложных приложений в области геонаук, таких как информационный центр GEOSS (глава 8), Climate@Home (глава 9) и прогнозирование пылевых бурь (глава 10). Образы содержат исходный код и данные для тех примеров, которые доступны для Amazon EC2. Поэтому аудитория может напрямую запускать виртуальные машины облачной обработки данных из этих образов и тестировать такие примеры.
- *Примеры руководства по развертыванию приложений* - Приведены подробные сведения о развертывании приложений рабочего процесса в облачных сервисах (главы 4, 5, 7, 8, 9 и 10). Кроме того, главы 12 и 14 также включают в себя детальное описание рабочего процесса для тестирования облачных сервисов.
- *Скрипты для установки и настройки примеров приложений и облачных сервисов.*
- *Видеоролики, показывающие пошаговое развертывание примеров приложений.*

## КТО ЯВЛЯЕТСЯ ЦЕЛЕВОЙ АУДИТОРИЕЙ?

Для обеспечения полного понимания пространственной облачной обработки данных, особенно для поддержки вычислительных потребностей приложений в области геонаук, мы написали данную книгу, основываясь на нашем десятилетнем исследовании многих проектов в сотрудничестве с различными агентствами и компаниями для решения вычислительных задач в области геонаук. В процессе чтения книги должен наблюдаться прогресс

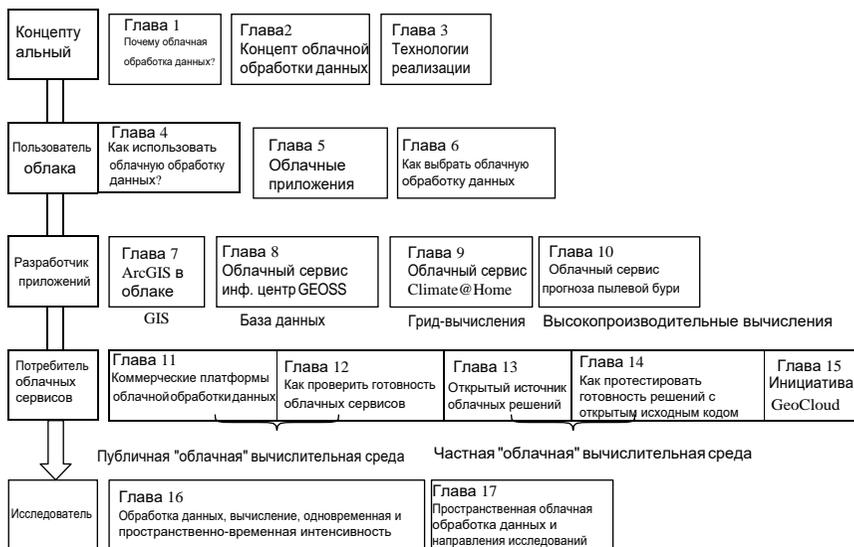


Рисунок Р.1 Руководство по прочтению книги.

в последовательности частей и глав книг. Но некоторые из них можно опустить, основываясь на интересе. На рисунке Р.1 показан процесс работы с главами для читателя в последовательности получения знаний.

Данную книгу можно использовать следующим образом:

(1) В качестве учебника преподавателями и студентами, которые планируют изучать различные аспекты облачной обработки данных с помощью сочетания онлайн-слайдов и примеров для лекций. Каждая глава включает в себя информационные слайды и может служить самостоятельным содержанием лекций. Главы Части со II по IV включают в себя подробные примеры, исходный код и данные, которые могут быть использованы для преподавательской практики, предоставляя студентам практический опыт использования и развертывания облачной обработки данных. Данные примеры также могут быть использованы в качестве домашней работы, чтобы закрепить то, что студенты изучили на лекции. Кроме того, примеры тщательно отобраны и рассмотрены в диапазоне от простого до сложного, так что студенты с различными уровнями подготовки могут заниматься вместе. Для каждой главы предусмотрено от пяти до десяти вопросов, чтобы помочь студентам подытожить основное содержание соответствующей главы.

(2) Руководство для разработчиков облачных приложений, содержащее руководящие принципы, постепенно приводится в частях II, III и IV. Сначала в данной книге дается общее руководство по развертыванию приложений в облачных сервисах (Глава 4). Затем на основе такого руководства вводится общий рабочий процесс развертывания геоаучных приложений на облачных сервисах (Глава 5). На основе предоставленного общего рабочего процесса используются три практических примера для демонстрации а) того, как можно использовать три различных типа

## xxii Введение

---

геонаучных приложений (базы данных, сетевые вычисления и высокопроизводительные вычисления [HPC]) для работы в облаке, и b) того, как учитывать особые требования различных приложений (Главы 8, 9 и 10). Помимо демонстрации того, как использовать облачные сервисы, в данной книге также приводятся рекомендации по выбору подходящих облачных сервисов (Глава 6) и тестированию облачных сервисов (Главы 12 и 14).

(3) Справочник для геоученых. В книге представлены различные аспекты облачной обработки данных: от требований к движущей силе (Глава 1), концепций (Глава 2) и технологий (Глава 3) до приложений (главы 8, 9 и 10), от выбора провайдера облачной обработки данных (Глава 6) до тестирования (Главы 12 и 14), от коммерческих облачных сервисов (Главы 4, 5, 11 и 12) до открытых облачных решений (Главы 13 и 14), а также от использования облачной обработки данных для решения проблем современных исследований и прикладных задач (Глава 16) до будущих тем для исследований (Глава 17). Ученые-геологи, обладающие знаниями в области исследований и научной специализации в различных областях, могут легко выявить необходимые знания в области облачной обработки данных, которые будут соответствовать их требованиям.

(4) Справочник для профессионалов в области ИТ и лиц, принимающих решения. В данной книге приводятся ссылки на концепции, технические детали и руководство по эксплуатации облачной обработки данных. В первых 15 главах даются поэтапные описания различных аспектов облачной обработки данных. Главы 4, 5 и с 7 по 15 тесно связаны с рутинной деятельностью в области ИТ. Лица, принимающие решения, могут использовать Главы с 1 по 3 для построения фундаментального понимания облачной обработки; затем перейдите к Главе 6 для рассмотрения вопросов, связанных с выбором облачных сервисов; и ознакомьтесь с полезной информацией в Главах 11, 13 и 15, которые охватывают как коммерческие, так и частные представления и оценки облачной обработки данных, наиболее важные для принятия решений.

---

## Благодарность

---

Авторы наработали опыт, участвуя в проектах на общую сумму более 10 миллионов долларов, финансируемых различными агентствами и компаниями, включая ОСЮ Национального управления по аэронавтике и исследованию космического пространства (НАСА), программу NASA MAP, программу NASA Applied Sciences, программу NASA High End Computing Программа EarthCube Национального научного фонда (NSF), Программа NSF I/UCRC, Программа NSF CNF и полярная программа NSF, Программа SilvaCarbon Геологической службы США (USGS), Федерация информационных партнеров по наукам о Земле (FGDC), Microsoft Research, межправительственная группа по Земле Observation (GEO), ESIP, Ассоциация американских географов (AAG), Cyberinfrastructure Speciality Group (CISG), ISPRS WG II/IV, Международная ассоциация китайских профессионалов в области географических информационных наук (CPGIS) и Amazon Inc. В число соавторов проекта, не являющихся прямыми авторами, входят Мирко Албани, Джон Антун, Жанетт Аркетто, Джефф де ла Божардьер, Дон Бадрок, Селеста Банааг, Дэвид Буркхальтер, Роберт Кахалан, Сонцин Чен, Ллойд Кларк, Гвидо Коланджели, Кори Дакин, Хазем Элдакбрехт, Джон Энгель Д. Эванс, Дэниел Фэй, Билл Финк, Пол Фукухара, Андриан Гарднер, Пэт Гэри, Том Гиффен, Стив Гленденинг, Йонантан Гойтом, Лон Гоуэн, Сью Харцлет, Мохаммед Хассан, Томас Хуанг, Хайсам Идо, Махендран Каданнапалли, Кен Кейзер, Пол Лэнг, Венвен Ли, Мэтью Линтон, Линда Липтрап, Майкл А. Литтл, Стивен Лоу, Ричард Мартин, Джефф Марц, Рой Мендельсон, Дэйв Меррилла, Лизи Мяо, Марк Миллер, Ник Мистри, Мэттью Моррис, Дуг Манчони, Аруна Маппалла, Стив Наус, Слободан Никович, Эрик У. Ноубл, Роберт Патт-Корнер, Горан Пеянович, Пит Поллергер, Крис Русановски, Тодд Сандерс, Гэвин А. Шмидт, Алан Сеттелл, Хавайя С. Шамс, Брайан Дж. Смит, Уильям Спригг, Майк Стефанелли, Джо Стивенс, Никола Трочино, Тиффани Вэнс, Арчи Варнок, Майк Уайтинг, Пол Визе, Лиза Вулфиш, Хуайи Ву, Ян Сюй и Авраам Т. Зейоханнис.

Главы были проверены внешними экспертами, в том числе Майклом Петерсоном из Университета Небраски в Омахе, Яном Труславом из Национального центра данных по морскому льду, Чуанронг Чжаном из Университета Коннектикута, Стефаном Фальке из Northrop Grumman,

## xxiv Благодарность

---

Сюань Ши из Университета Арканзаса, Лонг Фам из НАСА Годдард, Цзян Чен из Университета Луизианы, Цзинь Син из Университета Макгилла, Маршалл Ма из Политехнического института Ренсселера, Томас Хуанг из Лаборатории реактивного движения НАСА, Крис Бадур из Государственного университета Аппалачии, Дуг Неберт из FGDC, Рик Ким из Национального университета Сингапура, Руи Ли из Уханьского университета и Алисия Джефферс из Государственного университета Нью-Йорка в Дженесео. Наньинь Чжоу из Университета Джорджа Мейсона потратил много времени на форматирование книги. Питер Лостритто из Университета Джорджа Мейсона утвердили несколько глав.

Также благодарим редактора CRC Press / Taylor & Francis Ирму Бриттон, ее помощника и координатора производства Арлин Копелофф, Джозелин Бэнкс-Кайд, которые помогли обеспечить форматирование рукописи в соответствии со стандартами, принятыми CRC Press / Taylor & Francis. Они предоставили проницательные комментарии и проявили терпение при работе с нами. Мы очень благодарны студентам, которые работали над проектами и участвовали в написании глав.

Наконец, мы хотели бы поблагодарить членов нашей семьи за их терпение и понимание, когда нам приходилось жертвовать временем, которые мы могли провести с семьей.

Чаовой Ян хотел бы поблагодарить свою жену Янь Сян, своих детей.

Эндрю Ян, Кристофер Х. Ян и Ханна Ян.

Цюньин Хуан хотела бы поблагодарить своего мужа Юньфэн Цзян.

Чжэньлун Ли благодарит свою жену Вейли Сю и сына Мэйсона Дж. Ли.

Самая глубокая благодарность Чэнь Сюй его любимой жене, Цзяньпин Хань, за ее понимание, поддержку и терпение.

Кай Лю выражает благодарность своей жене Хуэйфэн Ван.

## Часть I

---

# Введение в облачную обработку данных для геонаук

---

Облачная обработка данных - это вычислительная парадигма нового поколения для совместного использования и объединения вычислительных ресурсов для удовлетворения динамических требований к вычислительным ресурсам, возникающих в результате многих проблем 21 века. В данной части представлены основы облачной обработки данных с нескольких аспектов: требования из области геонаук (Глава 1), концепции облачной обработки данных, архитектура и статус (Глава 2), а также технологии, которые сделали возможным облачную обработку данных (Глава 3).



## Глава 1

---

# Проблемы применения геофизических приложений для вычислительной инфраструктуры

*Чаовой Ян и Чен Сюй*

---

В данной главе рассказывается о необходимости новой вычислительной инфраструктуры, такой как облачная обработка данных, для решения нескольких сложных проблем, включая стихийные бедствия, нехватку энергии, изменение климата и устойчивость в 21 веке.

## 1.1 ЗАДАЧИ И ВОЗМОЖНОСТИ В ОБЛАСТИ ПРИМЕНЕНИЯ ГЕОНАУК В 21 ВЕКЕ

Геонаука сталкивается с серьезными проблемами при решении многих глобальных или региональных проблем, которые сильно влияют на нашу повседневную жизнь. Такие проблемы варьируются от истории зарождения планеты Земля до качества воздуха, которым мы дышим (NRC 2012a, b). В данном разделе рассматриваются проблемы энергетики, реагирования на чрезвычайные ситуации, климата и устойчивости.

### 1.1.1 Энергетика

В настоящее время около 80% мирового спроса на энергию удовлетворяется за счет ископаемого топлива (IEA 2010). Однако зависимость от ископаемого топлива неустойчива из-за двух фундаментальных проблем: во-первых, ископаемое топливо невозобновляемо и в конечном итоге такие ресурсы будут исчерпаны, и, во-вторых, потребление ископаемого топлива вызывает серьезные экологические и социальные проблемы, такие как изменение климата и конфликты, связанные с природными ресурсами. Администрация по энергетической информации США (EIA) и Международное энергетическое агентство (IEA) прогнозируют, что глобальное потребление энергии будет продолжать расти на 2% каждый год, при этом к 2040 году уровень потребления энергии вырастет вдвое, по сравнению с 2007 г. (IEA 2010). При ограниченных достижениях в развитии более устойчивых альтернативных источников энергии большая часть прогнозируемого возрастания потребления энергии будет приходиться на ископаемое топливо. Следовательно, мы приближаемся к истощению ископаемых видов топлива и производим больше парниковых газов. Задача удержать повышение средней глобальной температуры

## 4 Чаоуей Ян и Чен Сюй

---

ниже 2 градусов Цельсия по сравнению с доиндустриальными уровнями становится слишком оптимистичной, поскольку фундаментальные преобразования в энергопотреблении постоянно уклоняются (Петерс 2012).

IEA считает, что для достижения безопасного энергетического будущего прозрачность глобального энергетического рынка, обеспечиваемая анализом энергетических данных и глобальным сотрудничеством в области энергетических технологий, является важнейшей стратегией, которую необходимо предпринять. Одним из способов реализации расширенного потребления энергии на основе информации является интеллектуальная сеть, в которой используются цифровые технологии для динамического согласования производства энергии совместно с потребностями пользователей. Комплексные данные, совместные датчики и интеллектуальное управление энергопотреблением поставили огромную задачу перед передовыми вычислениями для развития возможностей обработки больших данных, поддержки динамического сотрудничества и включения интеллекта в энергосистему для интеллектуального управления энергопотреблением (IEA 2010).

### 1.1.2 Противоаварийные мероприятия

Природные и антропогенные катастрофы становятся все более частыми и серьезными в 21 веке из-за изменения климата, прироста населения и инфраструктуры. Например, птичий грипп 2003 г. распространился по всем континентам всего за несколько недель из-за международного импорта и перевозки людей (Ли и др., 2004).

Наводнения, вызванные ураганами, цунами и проливными дождями, ежегодно уносят десятки тысяч жизней людей по всему миру. Лесные пожары в сезон засухи приводят к потере активов на миллиарды долларов. Утечка опасных материалов, таких как ядерные материалы и ядовитый газ, также ежегодно уносит жизни сотен людей (IAPII 2009). Быстрое реагирование на стихийные бедствия и катастрофы, вызванные деятельностью человека, является важной задачей для спасения жизней и сокращения потерь имущества. Поддержка принятия решений при реагировании на чрезвычайные ситуации может быть наилучшим образом реализована только при своевременной интеграции большого объема геопространственной информации.

Например, на рис. 1.1 показана карта наводнения трагедии Нового Орлеана 2005 года, когда бушующий ураган Катрина затопил весь город. На карте показано место и глубина затопления от 0 до 14 футов. Карта могла бы стать отличным инструментом поддержки принятия решений, если бы ее можно было подготовить через несколько часов после урагана как для первых респондентов, так и для жителей Нового Орлеана, чтобы они могли решить, эвакуироваться или нет и где проводить поиск и спасение. К сожалению, для создания такой карты требуется несколько недель, чтобы собрать все геопространственные, метеорологические, строительные и другие наборы данных (Кертис, Миллс и Лейтнер 2006). Создание такой карты за несколько часов сопряжено с двумя трудностями. Во-первых, данные распределяются по разным агентствам и компаниям, и определение и интеграция наборов данных занимает относительно много времени. Во-вторых, требуются от десятков до сотен компьютеров, чтобы моделирование и расчеты затопления могли быть выполнены за несколько часов. Как только карта создана, вычислительные ресурсы могут быть высвобождены. Для этого требуется гибкая система обработки данных,

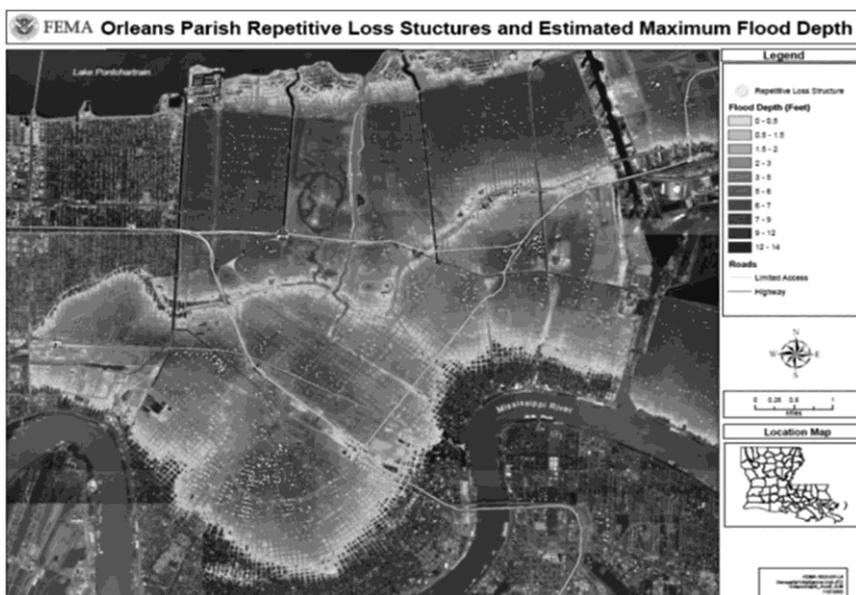


Рисунок 1.1 (См. Цветную вставку.) Карта глубины затопления Нового Орлеана после урагана Катрина. (Предоставлено Федеральным агентством по чрезвычайным ситуациям [FEMA], Министерство внутренней безопасности.)

которую можно выделить для обработки геопространственных данных за несколько минут с небольшим вмешательством человека или без него. Аналогичная вычислительная эластичность и быстрая интеграция также требуются для других аварийных мер, таких как лесные пожары, цунами, землетрясения и даже в большей степени для выбросов ядовитых и радиоактивных материалов.

### 1.1.3 Изменение климата

Глобальное изменение климата - одна из самых серьезных проблем, с которыми мы сталкиваемся в 21 веке. Изменение климата требует научных исследований для определения факторов, которые приводят к ответу на вопрос, как и почему меняется климат. Национальный исследовательский совет рекомендует провести необходимы дальнейшие исследования для понимания изменения климата по трем различным аспектам: (1) продвижение понимания изменения климата (NRC 2010a), (2) ограничение масштабов будущего изменения климата (NRC 2011) и (3) адаптация к воздействиям изменения климата (NRC 2010b).

Хотя факторы, влияющие на изменение климата, можно просто разделить на внутренние факторы (например, распределение воды и энергии) и внешние факторы (например, извержение вулкана, солнечная радиация и деятельность человека), для глубокого понимания изменения климата потребуются сотни параметров,

## 6 Чаовой Ян и Чен Сюй

---

которые были выявлены и исследованы, чтобы понять, как работает сложная система Земли и как эти параметры влияют на климатическую систему. Для ученых - сложная задача - построить множество моделей для количественной оценки влияния параметров, запустить множество различных конфигураций моделей и сравнить их с наблюдениями для получения научных знаний. При ограниченных вычислительных ресурсах становится важным использовать простаивающие вычислительные ресурсы в виде сетевых вычислений, таких как прогнозирование климата Великобритании или проекты NASA Climate @ Home (Stainforth и др., 2002). Управление вычислительной инфраструктурой потребует разрушительного хранения, связи, обработки и других вычислительных возможностей для координации данных и вычислений между компьютерами.

Ограничение масштабов будущего изменения климата зависит от научных достижений в области изменения климата и, в то же время, от хорошо продуманного гипотетического сценария, моделирующего возможности изменения климата с учетом ключевых параметров антропогенного воздействия, таких как двуокись углерода и другие экологические факторы, парниковые газы. Такое моделирование требует значительного количества вычислительных ресурсов, которые будут готовы к использованию в относительно короткий период времени для подготовки к международным и национальным переговорам; например, поддержка принятия решений по контролю выбросов углерода в политическом процессе (IEA 2010).

Адаптация к воздействиям изменения климата потребует от нас проведения множества много масштабных симуляций (Henderson-Sellers and McGuffie 2012), включая: (1) моделирование глобального масштаба для поддержки принятия решений, связанных с международной политикой, и (2) поддержку принятия решений на региональном уровне с помощью мезо - моделирования климатической модели в масштабе, основанное, например, на повышении уровня моря и смягчении воздействия на прибрежные города. Моделирование с более высоким разрешением может потребоваться для прогнозов, связанных с недвижимостью, из-за изменения климата; например, для поддержки разработки страховых полисов. Принимая во внимание влияние климата в ближайшем будущем, такие симуляции будут становиться все более частыми и могут быть запрошены как вычислительная услуга, когда запрашивается стоимость страховки для дома. Каждый из запросов может задействовать значительные вычислительные ресурсы за относительно короткий период времени для принятия решений в режиме реального времени, поэтому предъявляются пиковые требования к вычислительной инфраструктуре.

### 1.1.4 Устойчивое развитие

Устойчивость, возникшая в 1970-х и 1980-х годах (Kates and Clark, 2000), в значительной степени выигрывает от развития компьютерных наук и информационных технологий, которые предоставляют наборы инструментов для сбора данных, управления данными, вычислительного моделирования и многих других функций. Устойчивость как междисциплинарное исследование касается сложных взаимосвязей между тремя областями: природная среда, экономическая жизнеспособность и здоровые сообщества (Millett and Estrin 2012). Выживание людей зависит от наличия пресной воды на Земле. Поскольку общее количество пресной воды составляет лишь небольшой процент всех водных ресурсов на Земле, и ее доступность для людей сокращается из-за роста населения, нехватка водных ресурсов вызывает все больше региональных конфликтов (Kukk and Deese 1996).

## Проблемы применения геонаук в вычислительных инфраструктурах 7

---

Пресная вода является неотъемлемой частью производства многих продуктов жизнедеятельности человека. Она попадает в мировые товарные круги, что затрудняет планирование устойчивого использования водных ресурсов. Устойчивое планирование использования водных ресурсов требует комплексного моделирования потребления воды в различных масштабах - от местного до глобального. Данный процесс доказал свою информативность (Maier and Dandy 2000). По мере того, как Земля становится все более урбанизированной, возникают только более серьезные проблемы, которые ставят под угрозу общее состояние окружающей среды обитания. К таким проблемам относятся разрастание городов, городские острова тепла, связанное с санитарией бремя для здоровья и загрязнение окружающей среды, а также негабаритные энергетические и грунтовые воды. Мы остро нуждаемся в ограничениях на разрастание городов, дезинфекции городской среды и строительстве более пригодных для жизни городских пространств (Gökçekus, Türker, и LaMoreaux 2011). Требуется комплексное городское планирование. Такое планирование требует сбора, обработки и интеграции огромного объема данных для принятия решений. Этот процесс выигрывает от повышения доступности новых вычислительных ресурсов.

По прогнозам, к 2050 году население мира составит 9,3 миллиарда человек (Livi-Bacci 2012). В условиях, когда огромное население будет жить за счет неустойчивого потребления энергии на основе ископаемого топлива и сниженной индивидуальной доли пресной воды, а растущее число населения будет жить в постепенно ухудшающейся городской среде, устойчивость человеческого общества находится под угрозой. Чтобы обратить вспять такую опасную тенденцию, необходимо осуществить систематические и смелые политические изменения, которые должны быть подкреплены основательными научными исследованиями. По мере того, как в различных областях научных исследований происходят вычислительные повороты, исследовательские процессы кардинально меняются под влиянием цифровых технологий (Franchette 2011).

Проблемы устойчивого развития часто имеют общие черты масштаба (например, потребление энергии, которое может быть достигнуто на местном, региональном или глобальном уровнях) и гетерогенности (например, различные факторы способствуют наличию пресной воды, и существует множество решений этой проблемы). Наилучшим решением является оптимизация компромиссов между конкурирующими целями, что делает процесс интенсивным как с точки зрения данных, так и с точки зрения вычислительной техники. Например, мы находимся на ранней стадии проекта технологии "Интернет вещей" (Atzori, Iera, и Morabito 2010), характеризующегося устройствами, которые имеют встроенные датчики, исполнительные механизмы и процессоры обработки данных. Такие устройства способны обнаруживать, собирать, хранить и обрабатывать данные в режиме реального времени, например, из отчетов по мониторингу окружающей среды, фондовых рынков или персональных гаджетов. Количество собранных данных огромно. К концу 2011 года их количество составило 1,8 ЗБ, а к 2020 году - 35 ЗБ (Krishna 2011). Доступность больших данных дала новую парадигму принятия решений, основанную на данных. Для эффективной обработки больших данных необходимы компьютерные инновации.

## 1.2 ПОТРЕБНОСТЬ В СОЗДАНИИ НОВЫХ ВЫЧИСЛИТЕЛЬНЫХ ИНФРАСТРУКТУР

Проблемы, описанные в разделе 1.1, требуют создания вычислительной инфраструктуры, которая могла бы помочь в проведении соответствующих вычислений и обработки данных с учетом таких характеристик, как достаточные вычислительные возможности, минимальные затраты на энергию, быстрое реагирование на потребности, связанные со всплесками вычислительной техники, и, при необходимости, широкая доступность для общественности.

### 1.2.1 Обеспечение достаточной вычислительной мощности

Несмотря на то, что технологии вычислительного оборудования, включая центральный процессор (CPU), сеть, хранилище, оперативную память и графический процессор (GPU), были значительно усовершенствованы за последние десятилетия, многие вычислительные требования для решения научных и прикладных задач выходят за рамки существующих вычислительных возможностей. Высокопроизводительные вычисления (HPC), размещенные в вычислительных центрах, используются для научных исследований. Вычислительные возможности, предлагаемые высокопроизводительными вычислительными центрами, часто не соответствуют растущим требованиям к таким научным симуляциям и вычислениям в реальном времени. Вычисления для граждан - это один из подходов, который был принят научным сообществом для использования вычислительных циклов в целях решения данной проблемы. Например, в проекте SETI@Home используются компьютеры граждан для обработки сигналов из космоса с целью обнаружения возможной связи с пришельцами (Anderson et al. 2002). Из-за большого объема сигналов и возможных алгоритмов обработки таких сигналов требуются миллиарды запусков через данную комбинацию. Потребность в вычислительных ресурсах возрастет, когда большее количество сигналов будет улавливаться более совершенными датчиками и будут разработаны более сложные алгоритмы для более эффективной обработки сигналов. Гражданские вычисления становятся хорошим подходом для подобного способа обработки, так как нет никаких предсказуемых временных рамок, установленных для обнаружения внеземной жизни. Другим примером является проект Climate@Home, в рамках которого гражданские компьютеры тысячи раз помогают климатологам улучшить прогнозирование климатических моделей (Sun и др. 2012). Данный подход также использовался для помощи биологам в решении биологических проблем, таких как Folding@Home (Beberg и др., 2009).

Другая потребность в компьютерных технологиях для решения проблем заключается в том, чтобы заставить людей помочь в их решении. Мы часто решаем эту задачу, допуская проблему в общественное пользование или используя краудсорсинг. Такой тип формирующейся вычислительной модели помогает решать проблемы, например, разрабатывать планы городского транспорта или утверждать патентную заявку<sup>1</sup> для населения в определенный период времени. Награда обычно присуждается лучшему участнику, а интеллектуальная собственность принадлежит лицу, представившему заявку.

---

<sup>1</sup> См. раздел "Одноранговый патент" на <http://peertopatent.org/>.

## Проблемы применения геонаук в вычислительных инфраструктурах 9

---

Как гражданские вычисления, так и краудсорсинг являются эффективным решением некоторых проблем 21 века, как это было описано в предыдущих примерах. Однако ни то, ни другое не является надлежащей гарантией своевременности. Многие проблемы требуют получения вычислительной мощности в разумные сроки. Необходима новая вычислительная инфраструктура для своевременного решения таких задач.

### 1.2.2 Реагирование в реальном времени

В реагировании на чрезвычайные ситуации большинство оперативных систем будут требовать предоставления информации в режиме реального времени для поддержки процесса принятия решений, а односекундное раннее предупреждение или тревога может помочь спасти больше жизней (Asimakopoulou и Bessis 2010). Такой способ предоставления информации в режиме реального времени также широко распространена и в других средах поддержки принятия решений, например, в области координации и планирования вычислительной инфраструктуры. Требование в реальном времени предполагает быстрое выделение большого количества вычислительных ресурсов и быстрое высвобождение вычислительных ресурсов, когда аварийное реагирование завершено.

Аналогичным образом, в Интернете и социальных сетях существует спрос на свободный доступ к вычислительным ресурсам, что расширяет доступ к вычислительным ресурсам для общественности. Как только произойдет крупное событие, такое как президентские выборы в США в 2012 году или сильное землетрясение, в относительно короткие сроки потребуется значительное количество вычислительных ресурсов для реагирования общественности. Такой всплеск доступа может также существовать при пространственно-временном распределении. Для экзаменов по всему миру, когда Соединенные Штаты находятся в дневное время, будет существовать всплеск доступа к вычислительной инфраструктуре для общественной информации или повседневной работы. В то же время Азия, на другом конце Земли, будет иметь наименьший доступ к своей вычислительной инфраструктуре, и наоборот. Как использовать вычислительные мощности для такой глобальной пространственно-временной вычислительной потребности также является проблемой.

### 1.2.3 Экономия энергии

В последние десятилетия наблюдался быстрый рост скорости работы процессоров. Напротив, потребление энергии для процессоров также сократилось в геометрической прогрессии. Энергопотребление всей компьютерной системы также значительно снизилось за последние десятилетия. В дополнение к аппаратной части, программная часть управления вычислениями, планирования заданий и управления аппаратной частью также движется в сторону экономии энергии для получения оптимального количества ватт на гигабайт (миллиарды) операций с плавающей запятой (GFLOP) и оптимальное количества ватт на пропускную способность компьютера. Такие достижения были успешно использованы в одном компьютере и высокопроизводительных вычислительных системах (HPC). Однако стремительный рост использования вычислительных ресурсов (персональных компьютеров, серверов и других интеллектуальных терминалов, таких как планшетные компьютеры и мобильные устройства) резко увеличивает глобальное потребление энергии.

## 10 Чаовой Ян и Чен Сюй

---

Чтобы снизить глобальное потребление энергии, особенно для распределенной вычислительной системы, которая требуется для решения проблем 21 века, нам нужна система управления, оказывающая помощь в объединении и совместном использовании вычислительных ресурсов в географически удаленных регионах.

### 1.2.4 Экономия средств

Современные исследования или приложения, поддерживаемые вычислениями, обычно имеют очень разрушительное воздействие на использование вычислений. Например, при проведении научных исследований или учёбы в академической среде, нам может потребоваться использовать значительное количество вычислительных ресурсов за относительно короткий промежуток времени, например, полдня или три часа лекций в неделю. Несмотря на то, что требования к вычислительному ресурсу велики, мы не используем его на постоянной основе или менее чем на 10%. Поэтому размещение максимального количества частных вычислительных ресурсов либо не будет экономически эффективным, либо покупка таких вычислительных ресурсов должна производиться на основе полезной модели, или так называемых "утилит". Вычислительная инфраструктура идеально подходила бы для оплаты только той части, которая используется.

### 1.2.5 Повышение доступности

Популяризация Интернета и смарт-устройств, таких как планшеты и смартфоны, обеспечивает нам вездесущую вычислительную мощность, сопоставимую с более ранними настольными компьютерами. Многие приложения развертываются на смарт-устройствах и становятся общедоступными. Например, 85% владельцев телефонов используют свои устройства для отправки текстовых сообщений, а 23% используют их для серфинга в Интернете (Judge 2011). Большинство приложений также предоставляются с клиентами смарт-устройств для широкой публики. Такая широкая доступность вычислений с наших ежедневных смарт-устройств обеспечивает спрос и возможность доступа к распределенным вычислениям на гораздо более широкой основе, чем ранние сетевые настольные компьютеры или ноутбуки. И это становится идеальным и естественным требованием для обеспечения широкого доступа к вычислительным ресурсам с помощью смарт-устройств.

## 1.3 ЗАРОЖДЕНИЕ ОБЛАЧНОЙ ОБРАБОТКИ ДАННЫХ

### 1.3.1 Распределённая обработка данных

Постоянно растущий спрос на вычислительные мощности побуждает ученых и инженеров, занимающихся вычислительной техникой, исследовать новые парадигмы для экономически эффективного снабжения вычислительной техникой. Обмен вычислительными ресурсами берет свое начало с обмена крупномасштабными мейнфреймами, которые использовались учеными и компаниями в 1950-х годах, когда стоимость мейнфрейма была чрезвычайно высока (Richards и Lyn, 1990).

Для достижения наибольшей отдачи от инвестиций мейнфрейма были разработаны стратегии, позволяющие нескольким терминалам подключаться к мейнфрейму и работать за счет совместного использования. В то время совместное использование вычислительных ресурсов было локализовано там, где позиционировались мейнфреймы. Совместное использование мейнфреймов считалось самой ранней моделью облачной обработки данных (Voas и Zhang 2009). Модель совместного использования вычислительных ресурсов очень похожа на модель, которую используют коммунальные компании, такие как электрокомпании, для обслуживания клиентов. Джон Маккарти в своей речи в Массачусетском технологическом институте (MIT) впервые публично предположил, что вычисления будут предоставляться как общий утилит, концепция, которая впоследствии была тщательно изучена Дуласом Паркхиллом (1966). Херб Грош (Ryan, Merchant, и Falvey 2011) даже смело предсказал, что глобальные требования к вычислениям могут быть выполнены 15 крупными центрами обработки данных. Концепция была принята крупными компаниями, такими как IBM, для создания бизнес-модели совместного использования вычислительных ресурсов на основе технологии разделения времени

В 1990-х годах телекоммуникационные компании создали новый механизм повышения эффективности и безопасности удаленной передачи данных путем создания виртуальной частной сети (VPN) вместо изначального физического соединения данных, передаваемых только между двумя пунктами. Механизм VPN позволил повысить эффективность использования пропускной способности сети. Таким образом, концепция "облачной" обработки данных была расширена за счет совместного использования коммуникационной инфраструктуры, а изначально локализованная модель "облачной" обработки данных могла поддерживать географически рассредоточенных пользователей, чтобы они могли извлекать выгоду из мейнфреймов за счет использования расширенных возможностей связи через Интернет.

Идея облачной обработки данных возникла еще на ранней стадии сетевых вычислений, а название "облачные" вычисления происходит от общего использования символа "облако" в системных схемах сетевых вычислительных и коммуникационных систем (ThinkGrid 2013). Позже к облаку относят Интернет, который соединяет сети вычислительных систем через инфраструктуру связи. Концепция облачной обработки данных представляет собой тип сетевой структуры для управления и совместного использования вычислительных ресурсов, в основном с целью максимизации окупаемости инвестиций (ROI) на начальном этапе внедрения вычислительных ресурсов (Armbrust и др., 2010 г.). Облачная обработка данных в современной ИТ-среде расширяют совместное использование CPU и полосы пропускания для более тщательного обмена вычислительными ресурсами с помощью аппаратной виртуализации, сервис-ориентированной архитектуры и предоставления вычислительных услуг как вида утилитов через Интернет.

### **1.3.2 Обслуживание в режиме "вопрос-ответ"**

Архитектура, основанная на услугах, позволяет предоставлять вычислительные услуги по требованию. Облачная обработка данных становятся новой парадигмой коммунальных услуг, которая предоставляет вычислительную мощность и приложения в качестве услуги потребителям как государственных, так и частных организаций.

## 12 Чаовой Ян и Чен Сюй

---

Утилиты облачной обработки данных следуют общепринятой бизнес-модели энергетических компаний, которая основана на стоимости фактического использования, бизнес-модели с оплатой за использование. Таким образом, потребности потребителя в вычислениях могут быть удовлетворены более своевременно путем направления поставщику новых запросов на обслуживание, а поставщик взимает с потребителя плату на основании фактического использования.

### 1.3.3 Совместное использование вычислительной техники и снижение затрат

Благодаря бизнес-модели утилитов облачной обработки данных, клиенты могут быть освобождены от бремени постоянных инвестиций в покупку вычислительного оборудования и программного обеспечения, а также сэкономить на расходах на обслуживание системы. Потратившись на приобретение вычислительных услуг, заказчики переносят затраты на операционную деятельность. Поставщики вычислительных услуг собирают разнородные вычислительные ресурсы и распределяют их динамически и гибко в соответствии с потребностями. Добавляя аппаратную виртуализацию промежуточного программного обеспечения для управления и брокерского обслуживания оборудования, поставщики вычислительных услуг получают максимальную отдачу от первоначального оборудования за счет инвестиций, сокращая время простоя системы. Благодаря облачной обработке данных потребители и провайдеры формируют беспроигрышную комбинацию.

### 1.3.4 Надежность

Помимо экономии средств, приложения в облаке обычно достигают повышенной надежности с общим резервированием либо за счет расширения вычислительных мощностей одного облака, либо за счет интеграции нескольких облаков. Поставщики услуг облачной обработки данных могут легко добавлять новое оборудование в пул ресурсов, когда требуется большая вычислительная мощность. В соответствии с соглашениями об уровне обслуживания (SLA) несколько услуг могут быть связаны друг с другом. Таким образом, пользователи облачной обработки данных могут интегрировать несколько услуг от нескольких провайдеров, что потенциально снижает риски эксклюзивной зависимости от одного провайдера и повышает надежность. Исследования показали, что с помощью "облачной" обработки данных веб-приложения могут улучшить онлайн доступность (Armbrust и др., 2010 г.).

### 1.3.5 Появление облачной обработки данных

Достижения в области вычислительных технологий и технологий интернет-коммуникаций помогают использованию вычислений выйти за рамки простого совместного использования на основе времени на уровне физических компонентов (CPU, RAM) и перейти к совместному использованию на основе виртуализации на системном уровне, который обычно называют "облачной" обработкой данных. Как только вычислительные возможности могут быть предоставлены в качестве услуги, происходит внешнее развертывание ИТ-ресурсов, таких как серверы, хранилища или приложения, и приобретают их в качестве услуг.

### Проблемы применения геонаук в вычислительных инфраструктурах 13

---

Новая вычислительная парадигма позволяет внедрить более высокий уровень специализации для повышения эффективности производства (Dillon et al. 2010). Облачная обработка данных сочетается с множественными вычислительными технологиями виртуализации оборудования, сервис-ориентированной архитектурой и утилитами, как будет подробно описано в Главе 3.

Облачная обработка данных освобождает потребителей от необходимости приобретать собственное вычислительное оборудование, программное обеспечение и обслуживание, а также предоставляют следующие обобщенные функции:

- Вычислительные и инфраструктурные ресурсы и приложения предоставляются на сервисной основе.
- Услуги предлагаются поставщиками клиентам на платной основе.
- Виртуализация вычислительных ресурсов позволяет предоставлять их по требованию и обеспечивает динамическую масштабируемость вычислительных ресурсов.
- Услуги предоставляются в виде комплекса, включая вспомогательную инфраструктуру.
- Доступ к облачной обработке данных обычно осуществляется через веб-браузеры или через интерфейс прикладного программирования (API), настроенный по умолчанию.

Amazon был одним из первых провайдеров облачной обработки данных. Amazon Elastic Compute Cloud (EC2) абстрагирует общее аппаратное обеспечение как виртуальные машины Amazon с различными уровнями производительности, которые затем предоставляются клиентам в качестве услуг, которые они могут выбрать в зависимости от своих потребностей. Эластичность сервисов позволяет потребителям облака масштабировать вычислительные ресурсы в соответствии с требованиями по мере необходимости. Например, доступные в настоящее время вычислительные мощности EC2 варьируются от небольшого экземпляра (экземпляр по умолчанию) с 32-битной платформой (одно виртуальное ядро и 1,7 Гбайт памяти и 160 Гбайт памяти) до различных высокопроизводительных конфигураций, которые ориентированы на решение самых сложных задач. Например, для высокопроизводительных приложений наивысшей конфигурацией является 64-битная платформа с 8 виртуальными ядрами, 68,4 Гбайт памяти и 1690 Гбайт памяти.<sup>1</sup>

Как поставщики облачных услуг, Amazon выпустила EC2 в 2006 году, а Microsoft - Azure в 2008 году. За этим вскоре последовали многие открытые облачные решения, такие как Eucalyptus. В дополнение к обслуживанию на уровне операционной системы, некоторые облачные провайдеры обслуживают клиентов, которые являются разработчиками программного обеспечения, либо с поддержкой всех этапов разработки программного обеспечения, либо с платформой для специализированного использования, например, для управления контентом. Например, Google позволяет разработчикам приложений запускать их на базе инфраструктуры Google. Все больше провайдеров облачной обработки данных выходят на рынок, предлагая программное обеспечение как услугу. Классическим примером является Google Mail (Gmail).

---

<sup>1</sup> Согласно информации, предоставленной Amazon на <http://aws.amazon.com/ec2/instance-types/>.

## **1.4 ПРЕИМУЩЕСТВА И НЕДОСТАТКИ ОБЛАЧНОЙ ОБРАБОТКИ ДАННЫХ ДЛЯ ПРИЛОЖЕНИЙ ГЕОНАУКИ**

### **1.4.1 Преимущества облачной обработки данных**

Первоначальная концепция и механизм совместного использования вычислительных мощностей были сформированы в академической сфере, чтобы позволить как можно большему числу пользователей использовать мэйнфреймы одновременно (Richards и Lyn 1990). Современная облачная обработка данных предоставляют академическим исследователям вычислительные мощности, намного превышающие те, которые они получали ранее. В эпоху мэйнфреймов возможности вычислительных ресурсов были ограничены максимальной мощностью мэйнфрейма. Когда требования выходили за рамки существующих мощностей, приходилось приобретать новое оборудование. Поскольку облачная обработка данных дает возможность полностью удовлетворить спрос на вычисления, исследователи в области академических вычислений, которые пользуются традиционными механизмами совместных вычислений, такими как суперкомпьютеры, могут использовать потенциально неограниченные вычислительные ресурсы (Dikaiakos и др., 2009 г.). Экономия инвестиций в покупку новых дополнительных вычислительных ресурсов, а также затрат за счет использования более эффективного и мощного управления могут быть реализованы (Ху 2012).

Благодаря возможности доступа ко всем вычислительным ресурсам в виртуализированной вычислительной среде, работа критически важной научной вычислительной задачи может получить повышенную надежность, так как отказоустойчивые вычислительные ресурсы будут немедленно заменены доступными ресурсами с помощью аппаратного и программного обеспечения, а также резервированием географического местоположения. Стандартизированные API для облачной обработки данных также позволяют поставщикам облачной обработки данных бесперебойно предоставлять свои услуги нескольким брокерам, занимающимся облачными вычислениями, и наоборот.

Вычислительные мощности суперЭВМ центров используются в интенсивном научном применении. По сравнению с традиционными суперЭВМ центрами, платформы облачной обработки данных более доступны и дают пользователям больше контроля над своими кластерами. Рехр и др. (2010) показывают, как Amazon EC2 может обеспечить надежную высокопроизводительную поддержку общих научных вычислений. Таким образом, облачная обработка данных предоставляет новые возможности для научных вычислений с потенциально неограниченной доступностью виртуализированных вычислительных ресурсов. Облачная обработка данных - это вычислительная парадигма нового поколения, движимая вызовами 21-го века, которые требуют совместного использования и объединения вычислительных ресурсов для удовлетворения потребностей в динамических вычислениях (Yang и др., 2011).

### **1.4.2 Проблемы**

Централизованность вычислительных ресурсов делает сетевую инфраструктуру критически важной, начиная с конечных пользователей и заканчивая облачной обработкой данных. Если какая-либо из критических инфраструктур выйдет из строя, облачные службы могут быть недоступны.

## Проблемы применения геонаук в вычислительных инфраструктурах 15

---

Широкий круг конечных пользователей может потерять доступ к своим вычислительным ресурсам, если недоступен объект облачной обработки данных; например, отключение сети ЦОД Amazon Reston 2010 года вызвало глобальные последствия для пользователей Amazon EC2.

Распределение вычислительных ресурсов между организациями также приводит к окончательной потере потребителем контроля над вычислительными ресурсами. Например, обеспечение безопасности будет усложняться, а конфиденциальность данных будет трудно внедрить в общедоступную среду "облачной" обработки данных. Проблема еще более усугубляется, когда вычислительные ресурсы разделяются между странами.

Передача больших объемов данных в облаке или выполнение ресурсоемких вычислительных задач может быть экономически нецелесообразной при использовании облачной обработки данных. Поэтому облачная обработка данных не заменят все остальные режимы вычислений. Какое-то время в будущем будут сосуществовать различные вычислительные режимы (Mateescu, Gentsch и Ribbens 2011).

### 1.5 КРАТКОЕ ИЗЛОЖЕНИЕ

В данной главе в разделе 1.1 представлены требования к новой вычислительной инфраструктуре с точки зрения нескольких проблем 21-го века. В разделе 1.2 анализируются характеристики такой новой вычислительной инфраструктуры. В Разделе 1.3 обсуждается эволюция распределенных вычислений, приведшая к зарождению облачной обработки данных, а в Разделе 1.4 представлены преимущества и проблемы облачной обработки данных для применения в геонауках.

### 1.6 ПРОБЛЕМЫ

1. Перечислите три задачи в области прикладного применения геонаук, которые стоят перед нами в 21 веке, и обсудите потребности в вычислительной технике для их решения.
2. Перечислите две другие проблемы, с которыми мы сталкиваемся в области геонаук или в других областях, и обсудите потребности в вычислительных ресурсах.
3. Стихийные бедствия уносят множество человеческих жизней и приводят к большим потерям активов. Для реагирования на стихийные бедствия и смягчения их последствий ответственным органам необходима своевременная и точная информация. Могли бы Вы сосредоточиться на одном виде стихийных бедствий и изучить необходимую информацию и своевременно ее предоставить?
4. Каковы общие требования к вычислительным ресурсам для решения прикладных задач геонаук 21-го века?
5. Что такое распределенные вычисления? Можете ли Вы перечислить четыре парадигмы распределенных вычислений?
6. Что такое облачная обработка данных и как она связана с распределенными вычислениями?
7. Каковы преимущества и недостатки облачной обработки данных?

## СПИСОК ЛИТЕРАТУРЫ

- Anderson, D. P., J. Cobb, E. Korpela, M. Lebofsky, and D. Werthimer. 2002. SETI@ home: An experiment in public-resource computing. *Communications of the ACM* 45, no. 11: 56–61.
- Armbrust, M., A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee et al. 2010. A view of cloud computing. *Communications of the ACM* 53, no. 4: 50–58.
- Asimakopoulou, E. and N. Bessis. 2010. *Advanced ICTs for Disaster Management and Threat Detection: Collaborative and Distributed Frameworks*. Information Science Reference.
- Atzori, L., A. Iera, and G. Morabito. 2010. The Internet of things: A survey. *Computer Networks* 54, no. 15: 2787–2805.
- Beberg, A. L., D. L. Ensign, G. Jayachandran, S. Khaliq, and V. S. Pande. 2009. Folding@Home: Lessons from eight years of volunteer distributed computing. In *Parallel & Distributed Processing, IPDPS. IEEE International Symposium*, pp. 1–8. IEEE.
- Curtis, A. J, J. W. Mills, and M. Leitner. 2006. Spatial confidentiality and GIS: Re-engineering mortality locations from published maps about Hurricane Katrina. *International Journal of Health Geographics* 5, no. 1: 44.
- Dikaiaikos, M. D., D. Katsaros, P. Mehra, G. Pallis, and A. Vakali. 2009. Cloud computing: Distributed Internet computing for IT and scientific research. *IEEE Internet Computing* 13, no. 5: 10–13.
- Dillon, T., C. Wu, and E. Chang. 2010. Cloud computing: Issues and challenges. In *Advanced Information Networking and Applications (AINA), 24th IEEE International Conference*, pp. 27–33. University of York, UK, IEEE.
- Franchette, F. 2011. Why is it necessary to build a physical model of hypercomputation? *Proceedings of AISB'11: Computing and Philosophy* 1: 97–104.
- Gökçekus, H., U. Türker, and J. W. LaMoreaux, eds. 2011. *Survival and Sustainability: Environmental Concerns in the 21st Century*. Berlin, Heidelberg: Springer.
- Henderson-Sellers, A. and K. McGuffie. 2012. *The Future of the World's Climate*. Oxford, UK: Elsevier.
- IAPII (the InterAcademy Panel on International Issues). 2009. *Natural Disaster Mitigation: A Scientific and Practical Approach*, p. 146. Beijing: Science Press.
- IEA. 2010. Energy Technology Perspectives: Scenarios & Strategies to 2050. *International Energy Agency*. <https://www.iea.org/techno/etp/etp10/English.pdf> (accessed March 18, 2013).
- Judge, S. 2011. Texting, Social Networking Statistics Worldwide. <http://www.mobilephonedevlopment.com/archives/1394> (accessed March 28, 2013).
- Kates, R. W., W. C. Clark, R. Corell, J. M. Hall, et al. 2001. Sustainability science. *Science* 292: 641–642.
- Krishna, A. 2011. Why Big Data? Why Now? *IBM Corporation*. <http://almaden.ibm.com/colloquium/resources/Why%20Big%20Data%20Krishna.PDF> (accessed March 18, 2013).
- Kukk, C. L. and D. A. Deese. 1996. At the water's edge—Regional conflict and cooperation over fresh water. *UCLA J. Int'l L. & Foreign Aff.* 1: 21.
- Li, J et al. 2004. Study on transmission model of avian influenza. *Information Acquisition, Proceedings. International Conference*. IEEE.

Проблемы применения геонаук в вычислительных инфраструктурах 17

---

- Livi-Bacci M. 2012. *A Concise History of World Population*, 5th ed, p. 271. Malden: MA: John Wiley & Sons.
- Maier, H. R. and G. C. Dandy. 2000. Neural networks for the prediction and forecasting of water resources variables: A review of modelling issues and applications. *Environmental Modelling Software* 15, no. 1: 101–124.
- Mateescu, G., W. Gentsch, and C. J. Ribbens. 2011. Hybrid computing—Where HPC meets grid and cloud computing. *Future Generation Computer Systems* 27, no. 5: 440–453.
- Millett, L. I. and D. L. Estrin, eds. 2012. *Computing Research for Sustainability*. Washington, DC: National Academies Press.
- NRC. 2010a. *Advancing the Science of Climate Change*, p. 504. Washington, DC: National Academies Press.
- NRC. 2010b. *Adapting to the Impacts of Climate Change*, p. 272. Washington, DC: National Academies Press.
- NRC. 2011. *America's Climate Choices*, p. 118. Washington, DC: National Academies Press.
- NRC. 2012a. *International Science in the National Interest at the U.S. Geological Survey*, p. 161. Washington, DC: National Academies Press.
- NRC. 2012b. *New Research Opportunities in the Earth Sciences*, p. 117. Washington, DC: National Academies Press.
- Parkhill, D. 1966. *The Challenge of the Computer Utility*, p. 207. New York: Addison-Wesley.
- Peters, G. P., R. M. Andrew, T. Boden, J. G. Canadell, P. Ciais, C. L. Quere, G. Marland, M. R. Raupach, and C. Willson. 2012. The challenge to keep global warming below 2 [deg] C. *Nature Climate Change* 3: 4–6.
- Rehr, J. J., F. D. Vila, J. P. Gardner, L. Svec, and M. Prange. 2010. Scientific computing in the cloud. *Computing in Science & Engineering* 12, no. 3: 34–43.
- Richards, T. J. and R. Lyn. 1990. *Manual for Mainframe NUDIST: A Software System for Qualitative Data Analysis on Time-Sharing Computers*. Replee P/L.
- Ryan, P., R. Merchant, and S. Falvey. 2011. Regulation of the Cloud in India. *Journal of Internet Law* 15, no. 4: 7.
- Stainforth, D., J. Kettleborough, M. Allen, M. Collins, A. Heaps, and J. Murphy. 2002. Distributed computing for public-interest climate modeling research. *Computing in Science & Engineering* 4, no. 3: 82–89.
- Sun, M., J. Li, C. Yang, G. A. Schmidt, M. Bambacus, R. Cahalan, Q. Huang, C. Xu, E. U. Noble, and Z. Li. 2012. A Web-based geovisual analytical system for climate studies in future. *Internet* 4, no. 4: 1069–1085.
- ThinkGrid. 2013. Introduction to Cloud Computing. <http://logicplus.com.au/MenuItems/Infrastructure/Downloads/Cloud-whitepaper.pdf> (accessed March 18, 2013).
- Voas, J. and J. Zhang. 2009. Cloud computing: New wine or just a new bottle? *IT Professional* 11, no. 2: 15–17.
- Xu, X. 2012. From cloud computing to cloud manufacturing. *Robotics and Computer Integrated Manufacturing* 28, no. 1: 75–86.
- Yang, C., M. Goodchild, Q. Huang, D. Nebert, R. Raskin, Y. Xu, M. Bambacus, and D. Fay. 2011. Spatial cloud computing: How can the geospatial sciences use and help shape cloud computing? *International Journal of Digital Earth* 4, no. 4: 305–329.



## Глава 2

---

# Архитектура, концепции и характеристики облачной обработки данных

*Чаофей Ян и Цюньбин Хуан*

---

Облачная обработка данных возникла в результате эволюции распределенных вычислений (Armburst et al. 2010; Yang et al. 2009) и как реагирование на призывы к простому и быстрому совместному использованию вычислительных ресурсов для решения проблем 21 века.

### 2.1 КОНЦЕПТЫ

Доступность вычислительных ресурсов значительно повысилась со времен раннего режима использования распределенных вычислений в виде разделения во времени инфраструктуры мейнфреймов. Распределение мейнфреймов может быть запланировано только с ограниченным количеством локальных терминалов. Интернет освобождает вычислительные терминалы от размещения с мейнфреймом, а высокопроизводительные вычисления (НРС) увеличивают мощность централизованных вычислительных ресурсов. Грид-вычисления еще больше улучшают совместное использование вычислительных ресурсов в режиме динамической настройки конфигурации. Облачная обработка данных предоставляет практически неограниченные вычислительные ресурсы, к которым можно легко получить доступ. С появлением облачной обработки данных, граница совместного использования вычислительных ресурсов была значительно продвинута до утилит, ориентированных на рынок вычислительной техники.

Идею облачной обработки данных можно проследить еще с 1950-х годов; концептуальная модель была официально предложена в 1980-х годах; а развитие началось в 1990-х годах. Успешные облачные сервисы стали популярны только в последние несколько лет (Voas и Zhang 2009). Облачные сервисы начинались с обслуживания электронной почты, а затем расширялись и включали в себя множество других вычислительных возможностей в качестве сервисов (например, вычислительная мощность и хранение) или так называемый EaaS (XaaS) (Xu 2012; Yang et al. 2011). Отрасль предлагает множество типов облачных услуг, начиная с уровня инфраструктуры, например, Amazon Elastic Compute Cloud (Amazon EC2), и заканчивая уровнем приложений, таких как электронная почта, обмен документами и пул дудлов.

Свидетельствуя о неоднородности облачных услуг и давая рекомендации отрасли и учреждениям по предложению или потреблению облачных услуг, Национальный институт стандартов и технологий (NIST 2011) взял на себя лидирующую роль в определении стандартов для облачной обработки данных.

## 20 Чаовой Ян и Цюньин Хуан

---

Mell and Grance (2009) определяют облачную обработку данных как "модель, обеспечивающую повсеместный, удобный и доступ по требованию к общему пулу настраиваемых вычислительных ресурсов (например, сетей, серверов, хранилищ, приложений и служб), которые могут быть быстро предоставлены и выпущены с минимальными усилиями руководства или взаимодействия с поставщиками услуг". Далее они уточняют определение, включающее пять характеристик (самообслуживание по требованию, широкий доступ к сети, объединение ресурсов, быстрая адаптационная способность и измеримое обслуживание), три модели услуг (программное обеспечение как услуга, платформа как услуга и инфраструктура как услуга) и четыре модели развертывания (частное облако, общественное облако, публичное облако и гибридное облако). Пять характеристик, определенных NIST, подробно описаны в разделе 2.2. Они отличают облачную обработку данных от других парадигм распределенных вычислений (Yang et al. 2011).

## 2.2 АРХИТЕКТУРА ОБЛАЧНОЙ ОБРАБОТКИ ДАННЫХ

Различные архитектуры облачной обработки данных определяются с точки зрения потребителей таких услуг, поставщиков услуг облачной обработки данных и брокеров облачной обработки данных. Вариации архитектур облачной обработки данных также зависят от конкретных приложений. Наиболее популярной архитектурой является общая архитектура NIST (Liu et al. 2011), которая служит в качестве концептуальной архитектуры, не зависящей от поставщика. Архитектура, определяемая NIST, принята в данной книге и используется для представления концепций облачной обработки данных в настоящей главе.

Рисунок 2.1 адаптирован из эталонной архитектуры NIST (Liu et al. 2011), которая определяет шесть различных средств воспроизведения в облачной обработке данных. Поставщик облачной обработки данных (например, Amazon) - это человек или организация, которая обслуживает вычислительное оборудование и программное обеспечение и предоставляет облачные услуги другим лицам.

Аудитор "облака" (например, сотрудник службы безопасности, ответственный за Федеральный закон об управлении информационной безопасностью, FISMA) - это лицо или организация, которая контролирует и оценивает воздействие на безопасность, производительность и конфиденциальность "облачных" услуг, предоставляемых "облачным" провайдером. Потребитель "облачных" услуг, например, пользователь Amazon EC2, - это лицо или организация, закупающая и использующая "облачные" услуги. По аналогии с priceline.com, посредником в туристических ресурсах, "облачный" брокер - это лицо или организация, которая управляет использованием, производительностью и предоставлением "облачных" услуг и ведет переговоры об отношениях между поставщиками "облачных" услуг и потребителями "облачных" услуг. Облачный оператор является посредником, который обеспечивает подключение и транспортировку облачных услуг от провайдеров к потребителям облачных услуг. Существуют различные бизнес-модели облачной обработки данных. Например, потребитель "облачных" услуг может приобретать "облачные" услуги у провайдеров при посредничестве "облачного" брокера или без него. Пользователи в данной книге определяются как конечные пользователи доменных приложений, работающих на облачных сервисах, которые управляются потребителями облачных услуг.

## Архитектура, концепции и характеристики облачной обработки данных 21

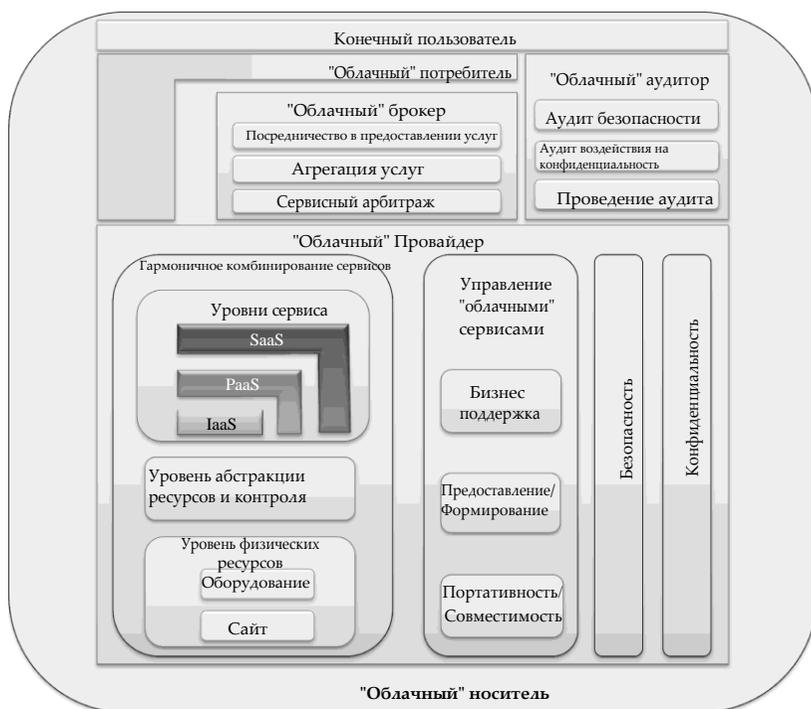


Рисунок 2.1 Эталонная модель облачной обработки данных (пересмотренная по сравнению с эталонной моделью облачных вычислений NIST). (Из Liu et al. 2011).

Существуют различные модели услуг, предоставляемых облачным провайдером, такие как Infrastructure as a Service (IaaS), Platform as a Service (PaaS) и Software as a Service (SaaS), как показано на рисунке 2.1.

- IaaS обеспечивает возможность предоставления вычислений, хранения, сетей и других фундаментальных вычислительных ресурсов, на которых могут быть развернуты операционные системы и приложения. Например, с помощью IaaS можно получить и получить доступ к виртуальному серверу, аналогично доступу к традиционному удаленному серверу. Популярным представителем IaaS является Amazon EC2.
- PaaS предоставляет вычислительную платформу, на которой потребители могут разрабатывать программное обеспечение, используя инструменты и библиотеки, предоставляемые облачным провайдером, а также развертывать программное обеспечение в облачных сервисах. Типичным примером PaaS являются Windows Azure и Visual Studio, предоставляемые компанией Microsoft.
- SaaS поддерживает программное обеспечение и данные, размещаемые в облаке и предоставляемые в виде услуг. Потребители или пользователи "облачных" услуг обычно получают доступ к данному типу услуг через веб-браузер. Популярным приложением SaaS является Gmail.

### 2.3 ХАРАКТЕРИСТИКИ

Некоторые характеристики облачной обработки данных (Mell and Grance 2009) отличают их от других парадигм распределенных вычислений (например, грид-вычисления) (Yang et al. 2011). Различия между парадигмами распределенных вычислений подробно описаны в главе 3. Ниже приводится краткое описание.

- *По запросу*—Облачная обработка данных обычно имеет огромный пул вычислительных ресурсов на задней панели для поддержки доступа потребителей по запросу. Например, "облачная" служба Google насчитывает более 1 миллиона серверов в трех десятках центров обработки данных по всему миру (Newton 2010); Amazon<sup>1</sup>, Microsoft<sup>2</sup> и другие поставщики общедоступной "облачной" обработки данных также управляют тысячами и миллионами физических серверов. Облака частных предприятий также имеют сотни серверов (Newton, 2010г.). Таким образом, облачная обработка данных может дать потребителям беспрецедентную вычислительную мощность по требованию, подкрепленную крупномасштабным вычислительным пулом.
- *Самообслуживание с минимальным взаимодействием с провайдерами*—Для потребителя возможности предоставления и выпуска ресурсов облачной обработки данных часто представляются неограниченными и могут быть приобретены в любом количестве в любое время без взаимодействия с облачными провайдерами, что похоже на наши покупки на Amazon.com.
- *Независимый доступ к широкополосной сети и устройству*—Облачные ресурсы доступны по сети и могут быть доступны через стандартные интерфейсы (например, мобильные телефоны, ноутбуки и персональные цифровые помощники [КПК]). Синхронизация данных между разнородными устройствами включает в себя сложные операции. Среди многих устройств трудно сохранять и поддерживать последние копии данных и информации. Облачная обработка данных может облегчить трудности синхронизации данных между различными копиями. Поскольку данные автоматически синхронизируются в "облаке", разные устройства могут получить одну и ту же версию данных при подключении к Интернету. Например, сервис Apple iCloud позволяет пользователям синхронизировать пакеты программ, адреса электронной почты, приложения, мультимедийные файлы и телефонные номера между компьютерами iPhone, iPad и Apple.
- *Пул ресурсов (для консолидации различных типов вычислительных ресурсов)*— Вычислительные ресурсы поставщика объединены для обслуживания нескольких потребителей с помощью модели с несколькими арендаторами. В рамках данной модели различные физические и виртуальные ресурсы динамически распределяются и перераспределяются в соответствии с потребительским спросом (Liu et al. 2011). Традиционные ИТ-системы, как правило, используют только от 10 до 15% своей полной мощности (Marston и др., 2011).

---

<sup>1</sup> См. ZDNet на <http://www.zdnet.com/blog/open-source/amazon-ec2-cloud-is-made-up-of-almost-half-a-million-linux-servers/10620>.

<sup>2</sup> См. Neowin на <http://www.neowin.net/news/main/09/11/02/inside-windows-azures-data-center-one-of-worlds-largest>.

## Архитектура, концепции и характеристики облачной обработки данных 23

Облачная обработка данных значительно улучшает совместное использование вычислительных ресурсов между организациями и повышает коэффициент использования вычислительных ресурсов до 80% (Yang и др., 2011 г.). В то же время, благодаря модели совместного использования ресурсов, основанной на виртуализации, "облачная" обработка данных снижает затраты потребителей на покупку, эксплуатацию и обслуживание вычислительных ресурсов. С точки зрения провайдера облачной обработки данных, технологии виртуализации могут применяться к различным уровням физических ресурсов. Таким образом, новейшие физические ресурсы могут быть добавлены в пул ресурсов "облака", что снижает затраты на обслуживание и модернизацию "облачной" платформы. С другой стороны, "облачные" потребители освобождаются от обслуживания инфраструктуры, передавая ответственность поставщикам "облачных" услуг на аутсорсинг.

- *Быстрая адаптационная способность (для быстрого и эффективного обеспечения, распределения и высвобождения вычислительных ресурсов)*— Внутри облачных сервисов приложения можно настроить так, чтобы они эффективно содержали больше ресурсов для работы с нагрузками, вызываемыми наплывом. При снижении нагрузки ресурсы могут быть быстро освобождены в течение нескольких секунд или минут (Huang et al. 2010).
- *Измеримое обслуживание (для поддержки моделей оплаты по факту)*— Облачная обработка данных применяет гибкие ценовые модели. Использование облачных ресурсов можно отслеживать, контролировать, регистрировать и взимать плату, обеспечивая прозрачность потребляемых услуг как для поставщика, так и для потребителя облачных услуг. Как правило, поставщики облачной обработки данных позволяют потребителям платить за использование вычислений по часам без долгосрочных обязательств. Поставщики облачных услуг также позволяют клиентам резервировать ресурсы на длительный срок по низкой оптовой цене. Кроме того, некоторые провайдеры (например, Amazon EC2) даже предоставляют ценовое предложение (по пунктам) для проведения торгов на неиспользуемые облачные ресурсы по еще более низкой цене.
- *Надежность*— Потребители облачной инфраструктуры могут получить от провайдеров вспомогательные услуги, такие как постоянное совершенствование проверенных физических инфраструктур и центров обработки данных. Такие услуги соответствуют новым политикам, исправлениям безопасности и обновляются без дополнительной оплаты. Например, Соглашение об уровне сервиса (SLA) Amazon EC2 обязывает потребителей обеспечить 99,95% доступности услуг во всех регионах Amazon EC2.

### 2.4 МОДЕЛИ УСЛУГ

Модель услуг представляет собой категоризацию типов сервисов, доступных через облачные службы. Настоящая книга посвящена наиболее проверенным моделям услуг IaaS, PaaS, SaaS и DaaS, особенно IaaS.

Как показано на рисунке 2.2, многоуровневая логика вычислений включает в себя аппаратное обеспечение, операционную систему, службу виртуализации (ядро облачной обработки данных), управление платформой, управление программным обеспечением, управление данными и доменные приложения.

24 Чаовой Ян и Цюньин Хуан

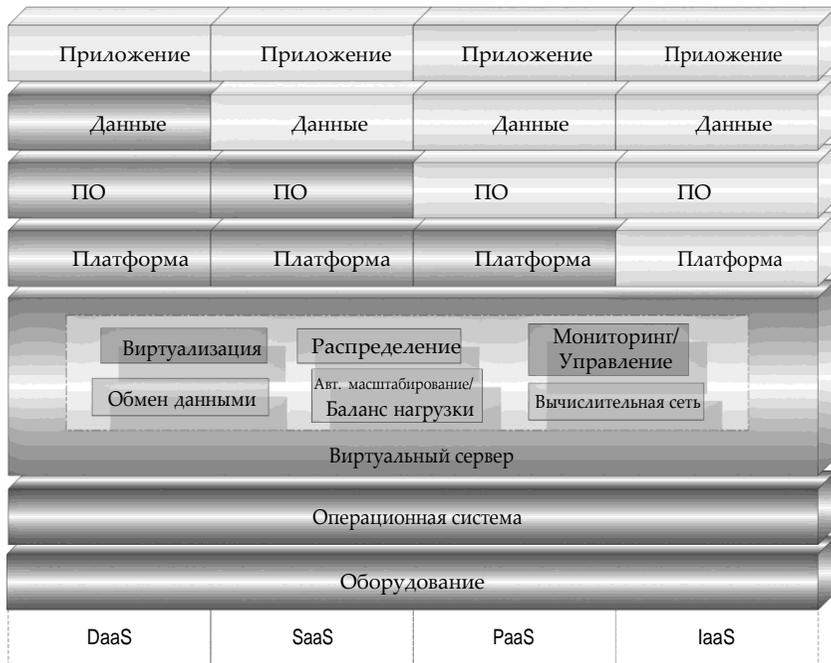


Рисунок 2.2 Модель сервиса облачной обработки данных в архитектуре технической логики. (Отредактировано Yang et al., 2011.)

Если поставщик облачных услуг предоставляет облачный сервис в виде виртуального вычислительного сервера, включающего три нижних уровня, то сервисный режим называется *IaaS*, например Amazon EC2. Если облачный провайдер предоставляет облачную услугу в качестве платформы, включающей нижние четыре уровня для разработки, тестирования и развертывания приложений и программного обеспечения, то сервисный режим называется *PaaS*, например, Windows Azure с Visual Studio. Если облачный провайдер предоставляет облачный сервис в виде программного обеспечения, включающего нижние пять уровней, то сервисный режим называется *SaaS*, например, Gmail и Google Earth. Если облачный провайдер предоставляет облачный сервис в виде облачной услуги передачи данных поверх всех слоев, за исключением приложений, то сервисный режим называется *DaaS*.

Приложения могут находиться на любом уровне услуг, IaaS, PaaS, SaaS и DaaS, в зависимости от функциональных возможностей. IaaS, PaaS и SaaS предоставляются как типичные облачные услуги с самого начала применения современной облачной обработки данных. Большая проблема с данными (The White House 2012) указывает на необходимость DaaS (например, data.gov). Отрасль также предоставляет другие вычислительные функции в качестве услуги, например, "Сеть как услуга" (Network as a Service - NaaS), "Безопасность как услуга" (Security as a Service - SECaaS) и "Хранение как услуга" (Storage as a Service - STaaS). В совокупности все модели сервисов обозначены EaaS (XaaS) (Yang и др. 2011).

## Архитектура, концепции и характеристики облачной обработки данных 25

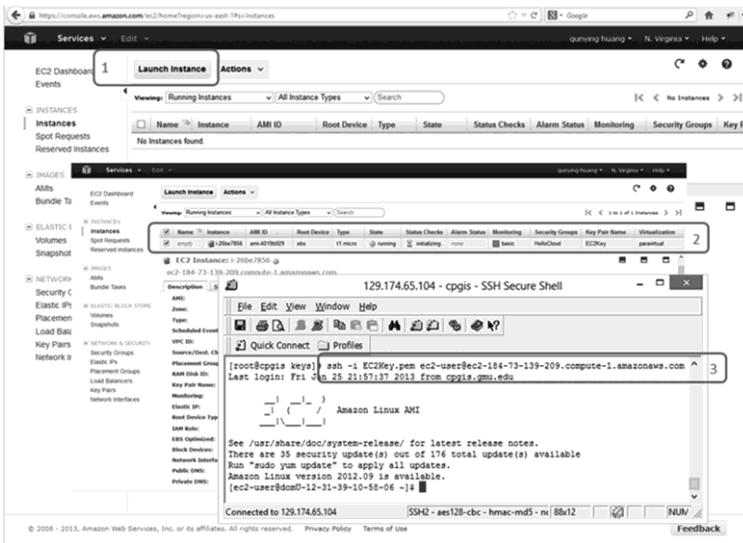


Рисунок 2.3 Как отмечено в последовательности, потребители облачной обработки данных могут: (1) Войти в AWS и запустить экземпляр, (2) проверить статус экземпляра и (3) войти в виртуальный экземпляр для настройки сервера

В качестве практического примера на рис. 2.3 показан Amazon EC2, который является представителем IaaS. Процедура начинается с покупки компьютера (виртуального экземпляра) путем входа в консоль EC2, запущенную в веб-браузере, и выбора типа запускаемого компьютера. После того, как экземпляр запущен, к нему можно получить доступ как к удаленному компьютеру, аналогично доступу к физическим серверам. Возможность развернуть и протестировать проект Visual Studio на облаке Azure была интегрирована с Microsoft Visual Studio посредством незначительной настройки с использованием учетной записи Azure (Рисунок 2.4). Подробности использования Amazon EC2 и Windows Azure представлены в Главе 4.

## 2.5 МОДЕЛИ РАЗВЕРТЫВАНИЯ И ТИПЫ ОБЛАКА

В целом, существует четыре типа развертывания облаков, включая публичное облако, частное облако, общественное облако и гибридное облако (Liu et al. 2011) (рис. 2.5).

- *Публичные облака* доступны для открытого доступа и использования публикой. Общеизвестная облачная система обычно предоставляется и управляется компанией. Потребители платят за то, что они потребляют. Поэтому публичное облако также известно как коммерческое. На сегодняшний день публичное облако является, пожалуй, самым популярным и часто-используемым типом облака.

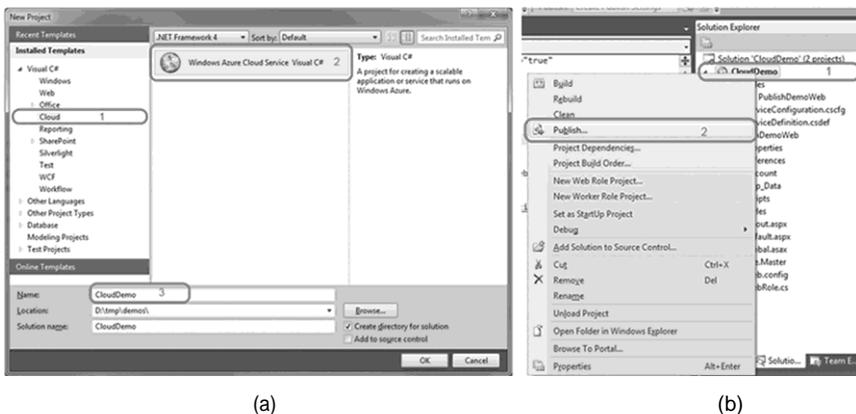


Рисунок 2.4 Создание, разработка, тестирование (а) и развертывание (б) приложений на Azure с помощью Microsoft VisualStudio.

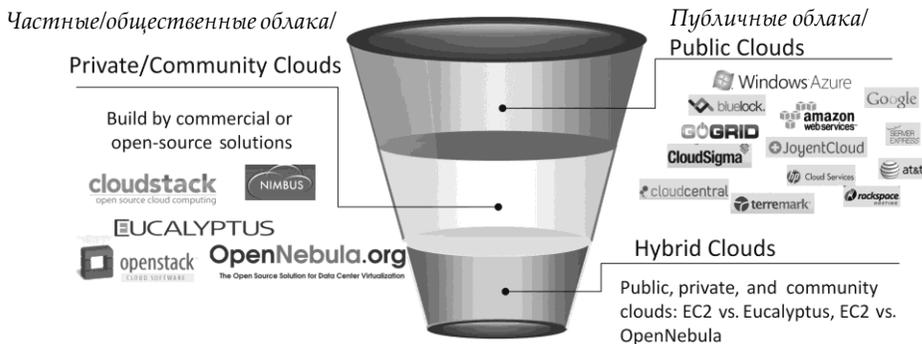


Рисунок 2.5 (См. цветную вставку) Типы облаков и программные решения.

- **Общественные облака** инициируются для того, чтобы служить обществу с общими интересами, такими как миссия, безопасность, соблюдение требований, юрисдикция и другие. Они предоставляются исключительно одной или нескольким организациям, и управляются внутри организации или третьей стороной.
- **Частные облака** предоставляются для исключительного использования одной организацией (например, бизнес-подразделениями). Они могут принадлежать и управляться предприятием, академическим учреждением, правительственной организацией или их консорциумом. Такие частные облака предлагают те же экономические и эксплуатационные преимущества, что и публичные облака, позволяя при этом компаниям или организациям сохранять абсолютный контроль над своими ИТ-ресурсами. Например, компания Hewlett-Packard (HP) перешла на частную облачную обработку данных; компания объединила свои существующие 85 центров обработки данных с 19 000 сотрудников в шесть облачных центров обработки данных с половиной штата сотрудников (Newton 2010).

Существует несколько решений с открытым исходным кодом для преобразования существующей физической инфраструктуры в частное облако или общественное облако, такие как Eucalyptus, CloudStack, OpenStack и OpenNebula (Глава 13) (Huang et al. 2012).

- Гибридные облака - это комбинация двух или более из вышеперечисленных типов облаков, созданных для решения конкретных задач или удовлетворения конкретных потребностей. Например, предприятия, занимающиеся программным обеспечением ИТ, могут создать гибридное облако с двумя частными облачными системами, в которых одна действует в качестве операционной системы, а другая служит для разработки и тестирования. Кроме того, публичные и частные облака могут быть объединены для достижения экономической эффективности и вычислительной мощности по требованию. Например, enStratus предоставляет брокерские услуги для более чем 10 облачных платформ<sup>1</sup>; OpenNebula может быть использована для построения частного облака, а его интерфейс прикладного программирования (API) может быть использован для доступа к публичному облаку Amazon EC2.

## 2.6 ОБЗОР РЕСУРСОВ ОБЛАЧНОЙ ОБРАБОТКИ ДАННЫХ

### 2.6.1 Коммерческие облака

Охваченные преимуществами экономической эффективности, автоматизации, масштабируемости и гибкости облачных услуг, многие организации переходят от традиционных физических ИТ-систем к облачным вычислениям; коммерческие ИТ-компании все чаще предоставляют облачные услуги (Armbrust и др., 2010; Huang и др., 2013). Например, Amazon предлагает несколько типов облачных сервисов, включая IaaS и PaaS. Windows Azure предоставляет услугу PaaS, с помощью которой пользователи могут быстро создавать, развертывать и управлять приложениями в глобальной сети центров обработки данных, управляемых компанией Microsoft. Подробная информация о веб-службах Amazon и службах Windows Azure представлена в Главе 4. В данном разделе кратко рассказывается о некоторых других популярных коммерческих облачных сервисах.

- Google App Engine (GAE)-Google рассказал об облачной обработке данных PaaS в нескольких статьях, опубликованных в период с 2003 по 2006 год. GAE был выпущен в качестве прототипа в 2008 году. Потребители могут разрабатывать веб-приложения на основе API GAE для размещения в центрах обработки данных Google. Данная система предлагает автоматическое масштабирование для веб-приложений и динамическое распределение вычислительных ресурсов по требованию. GAE является бесплатным вплоть до ограниченного количества вложенных ресурсов. Плата взимается за дополнительное хранение, пропускную полосу или необходимое приложению время работы.<sup>2</sup>
- Salesforce - Платформа Salesforce предоставляет потребителям различные услуги по созданию преобразующих приложений для бизнеса по управлению взаимоотношениями с клиентами (CRM),<sup>1</sup> в том числе:

<sup>1</sup> См. Dell Software на <http://enstratus.com/>.

<sup>2</sup> См. Google Developers на <https://developers.google.com/appengine/>.

(1) Force.com в целях создания приложений для сотрудников и настройки системы Salesforce; (2) Heroku для создания клиентских приложений на любом языке программирования; (3) Database.com для совместного использования и безопасного хранения данных в облачной базе данных; (4) Site.com для создания веб-сайтов и цифрового маркетинга; (5) Identi.ca для единой, социальной, доверенной личности в облаке; (6) AppExchange как доверенная среда облачных приложений; (7) ISVforce для создания мощных коммерческих приложений для клиентов облачных вычислений, и (8) Remedyforce для предоставления управления ИТ-услугами следующего поколения в облаке.

- Joyent-Joyent является одним из ИТ-поставщиков, который предоставляет облачные услуги по маршрутизации, включая программируемую балансировку нагрузки, вычисления, хранение, VPN (виртуальная частная сеть) и брандмауэр (межсетевой экран) сервиса. В отличие от многих "облачных" провайдеров (таких как Amazon), использующих технологию Xen для виртуализации, Joyent разрабатывает собственное программное обеспечение для виртуализации, SmartOS, для обеспечения стабильности и безопасности услуги. SmartOS обеспечивает сочетание аппаратного обеспечения и виртуализации операционной системы (ОС) для поддержки эффективной, надежной и высокопроизводительной "облачной" обработки данных.<sup>2</sup> Joyent обеспечила поддержку около 25% приложений Facebook, а еще один из ее клиентов, LinkedIn, за один месяц совершил более 10 миллионов посещений. Другой популярный сервис микроблогов Identi.ca также выбрал Joyent в качестве платформы.
- Rackspace - В июле 2010 года компания Rackspace, к которой присоединились лидеры технологических отраслей, таких как NASA, Citrix, Dell, NTT DATA, Rightscale и другие, запустила новый проект OpenStack с открытым исходным кодом для облачной обработки данных.<sup>3</sup> С момента своего создания, OpenStack был задуман с открытым исходным кодом, чтобы построить общую платформу для облачных провайдеров, а также для клиентов. OpenStack помог Rackspace быстро развиваться, предоставив возможность Rackspace конкурировать с другими крупными компаниями, такими как Amazon. В настоящее время, облачные продукты Rackspace внедряются на OpenStack.
- GoGrid<sup>4</sup>—GoGrid - это компания, принадлежащая американскому интернет-провайдеру ServePath, которая имеет долгую историю в сфере хостинга и предоставляет услуги интернет-хостинга, выделенных серверов, виртуальных хостов и виртуальных серверов.<sup>5</sup> GoGrid предлагает первую в мире веб-консоль управления для мультисерверного управления. С помощью консоли или интерфейсов API пользователи "облачной" обработки данных могут предоставлять и масштабировать виртуальные и физические серверы, хранилища, сети, балансировку нагрузки и брандмауэры в режиме реального времени в нескольких центрах обработки данных в течение нескольких минут.

<sup>1</sup> См. Salesforce на <http://www.salesforce.com/platform/overview/>.

<sup>2</sup> См. Joyent на <http://joyent.com/>.

<sup>3</sup> См. OpenStack Cloud Software на <http://openstack.org/>.

<sup>4</sup> См. GoGrid на <http://www.gogrid.com/>.

<sup>5</sup> См. CrunchBase на <http://www.crunchbase.com/company/servepath>.

## Архитектура, концепции и характеристики облачной обработки данных 29

---

Кроме того, GoGrid также предоставляет услуги выделенного хостинга, где потребители облачных услуг могут предоставлять выделенные (физические) серверы, интегрируя при этом гибкость и масштабируемость услуг хостинга облачных серверов.

- *Verizon's CaaS*—Помимо телекоммуникаций и дополнительных услуг, Verizon Business также инициирует деятельность в облаке. Например, Verizon CaaS позволяет клиентам оплачивать такие ресурсы ЦОДа, как хранилища и хостинг приложений, динамически в зависимости от объема потребляемых ими ресурсов.
- *IBM—SmartCloud* - это облачный сервис, который был создан корпорацией IBM в июне 2009 года. SmartCloud способен помочь компаниям быстро создать облако. Данное высокомасштабируемое облачное решение, сочетающее в себе возможности инфраструктуры и платформ для обеспечения гибкого управления рабочими нагрузками, управления жизненным циклом, а также отказоустойчивого, высокомасштабируемого резервирования на гетерогенных гипервизорных и аппаратных платформах. Reservoir - это совместная исследовательская инициатива IBM и Европейского Союза в области облачной обработки данных, которая позволяет осуществлять крупномасштабное развертывание и управление сложными ИТ-услугами в различных административных областях, ИТ-платформах и географических регионах.

### 2.6.2 Облачные решения с открытым исходным кодом

Несколько программ с открытым исходным кодом приложили усилия для исследования и разработки технологий и инфраструктуры для облачной обработки данных. Например, доступно несколько решений с открытым исходным кодом для развертывания и управления большим количеством виртуальных машин для создания высоко доступной и масштабируемой платформы облачной обработки данных, таких как Nimbus, Eucalyptus (Nurmi et al. 2009), OpenNebula (Sotomayor et al. 2009), CloudStack и Virtual Workspaces. Подробная информация о Eucalyptus, CloudStack, OpenNebula и Nimbus содержится в Главах 13 и 14. В данном разделе представлены несколько других проектов с открытым исходным кодом.

- *Amazon Elastic MapReduce (EMR)* - веб-сервис для эффективной обработки и анализа больших данных в облаке. Фреймворк Hadoop интегрирован для реализации вычислительной парадигмы MapReduce для распределенной обработки больших наборов данных на вычислительных кластерах с Amazon EC2, Amazon S3 и другими веб-сервисами Amazon (AWS). EMR также поддерживает технологии, связанные с Hadoop, такие, как Hive, Pig и HBase.
- *Xen Cloud Platform (XCP 2010)* - решение для виртуализации облачной инфраструктуры. Использует Xen в качестве гипервизора и предоставляет инструменты управления виртуальными машинами и сетью на одном хосте. Однако, в отличие от других облачных решений с открытым исходным кодом (например, Eucalyptus), XCP не предоставляет общую архитектуру для IaaS. Имеются схожести с VMware ESXi, который предоставляет инструмент для автоматического создания, настройки и обслуживания виртуальных машин.

### 30 Чаовой Ян и Цюньин Хуан

---

- *TPlatform* (Peng et al. 2009) is a cloud solution that provides a development platform for Web mining applications. It is a typical PaaS inspired by Google cloud technologies. TPlatform's infrastructure is supported by three components: a scalable file system called Tianwang File System (TFS) similar to the Google File System (GFS), the BigTable data storage mechanism, and the MapReduce programming model.
- *Apache VCL* (VCL 2008) - система с открытым исходным кодом для удаленного доступа к вычислительным ресурсам через веб-интерфейс. Ресурсы обычно размещаются в центре обработки данных. VCL может служить брокером для доступа к автономным машинам (например, лабораторному компьютеру в университетском городке). Структура VCL включает в себя: (1) веб-клиент, предоставляющий пользовательский интерфейс, который позволяет запрашивать ресурсы VCL и управлять ими; (2) сервер базы данных, хранящий информацию о резервированиях VCL, управлении доступом и инвентаризации машин и среды; и (3) несколько узлов управления для управления подмножеством ресурсов VCL.

## 2.7 КРАТКОЕ ИЗЛОЖЕНИЕ

В данной главе представлены основы облачной обработки данных на основе определений NIST. Раздел 2.1 знакомит с основными концепциями облачной обработки данных. Раздел 2.2 знакомит с архитектурой облачной обработки данных. В разделе 2.3 представлены характеристики облачной обработки данных. Раздел 2.4 знакомит с сервисными моделями облачной обработки данных. В разделе 2.5 представлены типы облачной обработки данных на основе модели развертывания. В разделе 2.6 представлен обзор доступных облачных сервисов и решений с открытым исходным кодом. Остальная часть книги посвящена технической стороне того, как облачной обработки данных может использоваться для поддержки приложений геонаук.

## 2.8 ПРОБЛЕМЫ

1. Что такое определение облачной обработки данных NIST?
2. Какие три самые популярные модели обслуживания облачной обработки данных?
3. Каковы основные характеристики облачной обработки данных? Исходя из вашего понимания, обсудите, какой из них самый важный.
4. Каковы четыре различные модели развертывания облачной обработки данных и сообщества пользователей?
5. Перечислите четыре коммерческих облачных сервиса и их поставщиков.
6. Перечислите четыре решения для облачной обработки данных с открытым исходным кодом и их владельцев.

## СПИСОК ЛИТЕРАТУРЫ

- Armbrust, M., A. Fox, R. Griffith et al. 2010. A view of cloud computing. *Communications of the ACM* 53, no. 4: 50–58.
- Huang, Q., J. Xia, C. Yang, K. Liu et al. 2012. An experimental study of open-source cloud platforms for dust storm forecasting. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, pp. 534–537. ACM.
- Huang, Q., C. Yang, K. Benedict, S. Chen, A. Rezugui, and J. Xie. 2013. Enabling dust storm forecasting using cloud computing. *International Journal of Digital Earth*. doi:10.1080/17538947.2012.749949.
- Huang, Q., C. Yang, D. Nebert, K. Liu, and H. Wu. 2010. Cloud computing for geosciences: Deployment of GEOSS Clearinghouse on Amazon's EC2. In *Proceedings of the ACM SIGSPATIAL International Workshop on High Performance and Distributed Geographic Information Systems*, pp. 35–38. ACM.
- Liu, F., J. Tong, J. Mao, R. Bohn, J. Messina, L. Badger, and D. Leaf. 2011. NIST cloud computing reference architecture. *NIST Special Publication* 500, 292.
- Marston, S., Z. Li, S. Bandyopadhyay, J. Zhang, and A. Ghalsasi. 2011. Cloud computing—The business perspective. *Decision Support Systems* 51, no. 1: 176–189.
- Mell, P. and T. Grance. 2009. The NIST definition of cloud computing. *National Institute of Standards and Technology* 53, no. 6: 50.
- Newton, J. 2010. Is cloud computing green computing? *GP Solo* 27, no 8: 28–31.
- Nurmi, D., R. Wolski, C. Grzegorzczak, G. Obertelli, S. Soman, L. Youseff, and D. Zagorodnov. 2009 (May). The eucalyptus open-source cloud-computing system. In *Cluster Computing and the Grid. CCGRID'09. 9th IEEE/ACM International Symposium*, pp. 124–131. IEEE.
- Peng, J., X. Zhang, Z. Lei, B. Zhang, W. Zhang, and Q. Li. 2009. Comparison of several cloud computing platforms. *Proceedings of 2nd International Symposium on Information Science and Engineering*, pp. 23–27.
- Sotomayor, B., R. S. Montero, I. M. Llorente, and I. Foster. 2009. Virtual infrastructure management in private and hybrid clouds. *Internet Computing, IEEE* 13, no. 5: 14–22.
- The White House, Office of Science and Technology Policy. 2012. Obama Administration Unveils “Big Data” Initiative: Announces \$200 Million in New R&D Investments. [http://www.whitehouse.gov/sites/default/files/microsites/ostp/big\\_data\\_press\\_release\\_final\\_2.pdf](http://www.whitehouse.gov/sites/default/files/microsites/ostp/big_data_press_release_final_2.pdf) (accessed January 18, 2013).
- VCL. 2008. Apache VCL. <https://vcl.ncsu.edu/> (accessed January 18, 2013).
- Voas, J. and J. Zhang. 2009. Cloud computing: New wine or just a new bottle? *IT Professional* 11, no. 2: 15–17.
- XCP. 2010. Xen Cloud Platform. <http://www.xen.org/products/cloudxen.html> (accessed January 18, 2013).
- Xu, X. 2012. From cloud computing to cloud manufacturing. *Robotics and Computer-Integrated Manufacturing* 28, no. 1: 75–86.
- Yang, C., M. Goodchild, Q. Huang et al. 2011. Spatial cloud computing: How can the geospatial sciences use and help shape cloud computing? *International Journal of Digital Earth* 4, no. 4: 305–329.
- Yang, C. and R. Raskin. 2009. Introduction to distributed geographic information processing research. *International Journal of Geographical Information Science* 23, no. 5: 553–560.



## Глава 3

---

# Технологии реализации

*Чжэньлун Ли, Цюньин Хуан и Чжиэнь Гуй*

---

Облачная обработка данных обеспечивает (1) практически неограниченное хранилище для архивирования и управления растущим объемом данных, (2) вычислительные возможности по требованию для поддержки приложений с интенсивными вычислениями и (3) услуги для эффективного использования сети, хранилища и вычислительных ресурсов. Такие возможности стали доступными благодаря множеству технологий, которые способствовали появлению облачной обработки данных. Чтобы помочь читателям понять, как возникла и развивалась облачная обработка данных, в настоящей главе представлены ключевые технологии, обеспечивающие возможность облачной обработки данных, включая усовершенствования оборудования, распределенные вычисления, виртуализацию, распределенные файловые системы и Web x.0.

### 3.1 АППАРАТНОЕ ОБЕСПЕЧЕНИЕ

#### 3.1.1 Многоядерные и специализированные многоядерные технологии

Высокопроизводительные вычисления (HPC) стали незаменимым инструментом для эффективного решения сложных научных, инженерных и бизнес-задач. Однако модель одноядерных и многопоточных вычислений, используемая в традиционных высокопроизводительных вычислениях, не может удовлетворить интенсивный спрос на вычисления (Pase and Eckl 2005). В целях восполнения данного пробела, появилась многоядерная технология, которая используется для кластерных вычислений с конца 1990-х годов. Многоядерная технология позволяет собрать несколько ядер центрального процессора (ЦП) на одном кристалле для совместного использования кэша и памяти. За счет балансировки задач между несколькими ядрами ЦП общая производительность многоядерных систем может быть значительно улучшена. Точно так же многоядерная технология позволяет интегрировать от десятков до сотен процессорных ядер в чип, такой как графический процессор (GPU), что дополнительно увеличивает вычислительную мощность. Многоядерные и специализированные многоядерные технологии широко используются в средах высокопроизводительных вычислений (Chai, Gao и Panda, 2007) из-за характеристик низкого потребления электроэнергии, эффективного использования пространства и хорошей производительности в операциях с общей памятью (Sagona 2009). Какие преимущества для облачной обработки данных от многоядерных и специализированных многоядерных технологий? В эпоху облачной обработки данных провайдеры обычно создают несколько распределенных центров обработки данных с массивными физическими компьютерами для поддержки потребителей, выделяющих облачные ресурсы по запросу.

Каждый физический компьютер может использоваться одновременно несколькими потребителями, и запросы доступа назначаются разным ядрам ЦП. В модели «Инфраструктура как услуга» (IaaS) каждый потребитель может использовать вычислительную мощность физического компьютера с помощью виртуальной машины (Раздел 3.3). В таких условиях физические вычислители в облачных центрах обработки данных участвуют в тяжелых вычислениях и повышают пропускную способность. Возможности параллельной обработки многоядерных и специализированных многоядерных процессоров помогают облегчить вычислительную нагрузку на каждый физический компьютер. Таким образом, многоядерные и специализированные многоядерные технологии позволяют поставщикам облачных услуг создавать энергоэффективные и высокопроизводительные центры обработки данных, которые являются основными строительными блоками облачной обработки данных.

### 3.1.2 Вычислительная сеть

Облачная обработка данных - модель, в которой ресурсы и услуги абстрагируются от базовой инфраструктуры и предоставляют гибкие услуги по запросу в многопользовательской среде. Сеть имеет решающее значение для облачной обработки данных как минимум в двух аспектах:

- Облачный провайдер предоставляет услуги потребителям и пользователям облачной обработки данных через вычислительную сеть. Компьютерная сеть соединяет вместе все облачные компоненты. Без сетей потребители не могут получить доступ к своим облачным сервисам. Вычислительная инфраструктура, программные платформы, приложения и данные могут быть физически доступны только потребителям и пользователям.
- Связь вычислительной сети между потребителями и поставщиками облачных услуг определяют технологические различия между различными моделями развертывания. В частном облаке потребители и поставщики находятся в пределах одной доверенной сети. В публичном облаке все облачные сервисы предоставляются через сеть облачного оператора. В гибридном облаке может существовать защищенное соединение между сетью потребителя и сетью поставщика. В общественном облаке структура сетевых подключений зависит от архитектуры организации, использующей облако.

### 3.1.3 Хранилище

Хранилище - это устройство, которое традиционно называют жестким диском, ленточной системой и другими устройствами, подключенными к компьютеру. Облачное хранилище - это хранилище, которое может быть предоставлено потребителям как услуга, включая сетевое оборудование, онлайн-хранилище, онлайн-резервное копирование, онлайн-документы и другое. Являясь неотъемлемой частью облачного сервиса, облачное хранилище представляет собой виртуализированный расширяемый ресурс.

В среде облачной обработки данных по-прежнему очень сложно хранить большие объемы данных и управлять ими. Сложному облачному сервису обычно требуется большой объем хранилища, чтобы обеспечить гибкую поддержку потребителей. К счастью, быстрое развитие технологий жестких дисков привело к постоянному увеличению емкости хранилища, а также к снижению цен. Облачная среда - важный компонент инфраструктуры облачной обработки данных. Это окажет значительное влияние на развитие облачных сервисов в будущем.

### **3.1.4 Смарт-устройства**

Облачный сервис предлагает потребителям возможность доступа к информации по запросу с различных устройств независимо от оборудования. Распространение различных смарт-устройств (например, смартфонов и планшетных компьютеров) ускоряет развитие облачных сервисов, обогащая их каналы доступа для потребителей и пользователей облачной обработки данных. Например, при поездке в автомобиле при запросе онлайн-карт можно быстро получить маршрут на основе условий дорожного движения в реальном времени через облачную службу. Благодаря возможности использования различных беспроводных сетей (например, WiFi, 3G и 4G) смарт-устройства помогают значительно популяризировать облачные сервисы.

## **3.2 ВЫЧИСЛИТЕЛЬНЫЕ ТЕХНОЛОГИИ**

### **3.2.1 Парадигма распределенных вычислений**

Идея распределенных вычислений состоит в использовании вычислительной мощности нескольких распределенных компьютеров для решения более крупных вычислительных задач. При распределенных вычислениях крупномасштабная задача делится на множество подзадач, каждая из которых решается одним или несколькими компьютерами. Распределенные вычисления включают несколько вариантов, таких как кластерные вычисления, грид-вычисления и утилиты вычисления.

Несмотря на то, что по-прежнему существуют разногласия касательно отношений между облачной обработкой данных и вариантами распределенных вычислений, общепринято считать, что распределенные вычисления являются наиболее очевидной технологией-предшественником, которая позволила зародиться облачной обработке данных (Youseff, Butrico, and Silva 2008), и что первоначальная идея облачной обработке данных произошла от распределенных вычислений. Такая эволюция явилась результатом смещения фокуса с инфраструктуры, которая предоставляет ресурсы хранения и вычислительных ресурсов (например, в случае грид-вычислений), на инфраструктуру, основанную на экономике (Foster et al. 2008). Облачная обработка данных нацелена на то, чтобы сделать обширные и высоко виртуализированные ресурсы легко доступными и масштабируемыми через Интернет. Данные ресурсы предназначены для различных услуг, включая программное обеспечение, платформу и инфраструктуру.

### 3.2.2 Модель вычислительной архитектуры

С развитием парадигм распределенных вычислений возникли две модели: модель клиент / сервер (C / S) и модель браузер / сервер (B / S).

- Модель C/S была представлена в 1980-х годах и предлагала центральное хранилище компьютеров, в то время как персональные компьютеры и рабочие станции служили терминалами, позволяя людям запускать программы локально (Toggraph and Morgens 2008). C/S описывает отношения между двумя сотрудничающими компьютерными программами, где одна отправляет запрос от клиента, а другая на сервере обрабатывает запрос.
- Модель B/S возникла с развитием Интернета и веб-технологий, а веб-браузер использовался в качестве клиентской программы. Данная модель объединяет различные клиентские программы в модели C / S и упрощает разработку, обслуживание и использование системы.

Облачная обработка данных обеспечивают удобный сетевой доступ по запросу к общему пулу конфигурируемых вычислительных ресурсов, а облачные сервисы доступны с различных клиентских устройств через интерфейс клиента, в основном через веб-браузеры (Mell and Grance 2011). С данной точки зрения B / S - ключевая модель вычислительной архитектуры в эволюции облачной обработки данных.

### 3.3 ВИРТУАЛИЗАЦИЯ

Одной из основных функций облачной обработки данных является абстракция физической реализации, позволяющая скрыть технические детали от потребителей (например, на каком оборудовании фактически запускается приложение, где оно расположено и другие детали конфигурации). Данная функция поддерживается технологией виртуализации. Виртуализация поддерживает инфраструктуру облачной обработки данных (например, Amazon EC2) для предоставления вычислительной мощности потребителям из удаленных мест через Интернет (Liu and Orban 2008). Виртуализация также позволяет вычислительной системе динамически получать или освобождать вычислительные ресурсы, чтобы система была устойчивой к отказам компонентов. Консолидация серверов, миграция гостевой среды выполнения и защита от вредоносного кода - вот некоторые из наиболее веских причин для использования виртуализации (Liu and Orban 2008). Благодаря виртуализации облачная обработка данных имеет возможность создавать динамическое количество виртуальных машин (VM) в зависимости от потребностей приложения. Например, в зависимости от интенсивности и характера вычислений, задача может потребовать вычислительных ресурсов от одного до сотен процессоров. Виртуализация обеспечивает масштабируемость и гибкость облачной обработки данных, скрывая детали развертывания от потребителей (Vaquero et al. 2008). Например, с помощью виртуализации IaaS может разделять, назначать и динамически изменять размер вычислительных ресурсов для создания специальных систем, по запросу.

### 3.3.1 Реализация виртуализации

Гипервизор, устанавливающий уровень абстракции между виртуальными машинами и базовым оборудованием, представляет собой программу, которая реализует виртуализацию и позволяет нескольким операционным системам совместно использовать один аппаратный хост. Гипервизор может захватывать инструкции ЦП и действовать как координатор инструкций для доступа к аппаратным контроллерам и периферийным устройствам (рис. 3.1).

Существует широкий спектр подходов к виртуализации. Среди них три ведущих подхода: полная виртуализация, паравиртуализация и виртуализация оборудования (Sahoo, Mohapatra и Lath 2010).

- *Полная виртуализация* - основана на парадигме хост / гость, где каждый гость работает на виртуальной имитации аппаратного уровня. В среде полной виртуализации гипервизор работает непосредственно на оборудовании, а гостевая операционная система (ОС), на которой работает виртуальная машина, управляется гипервизором. Гостевая ОС не связана с ОС хоста, потому что отсутствует информация о том, что ОС не работает непосредственно на реальном оборудовании.
- *Паравиртуализация* - Главный недостаток подхода полной виртуализации - это накладные расходы на производительность, связанные с гипервизором. Один из способов уменьшить такие накладные расходы - изменить ОС виртуальных машин и сообщить виртуальным машинам, что они работают в виртуальной среде и могут работать вместе с гипервизором. Данный метод называется паравиртуализацией. Такой подход представляет каждой виртуальной машине абстракцию оборудования, которое похоже на базовое физическое оборудование, но не идентично ему. Паравиртуализация пытается предоставить большинство функций непосредственно от базового оборудования, а не абстрагироваться от него.
- *Виртуализация оборудования* - это решение виртуализации, в котором гипервизор встроен в схему аппаратного компонента, а не поддерживается сторонним программным приложением.



Рисунок 3.1 Реализация виртуализации.

### 3.3.2 Решения виртуализации

В настоящее время разработан широкий спектр решений виртуализации, включая коммерческие решения и решения с открытым исходным кодом. Например, Xen, VMware и VirtualBox поддерживают широкий спектр операционных систем в качестве хостов или гостей. В результате проблема совместимости программного обеспечения устраняется путем гомогенизации на уровне программного обеспечения (McEvoy and Schulze 2008). В таблице 3.1 показаны широко используемые решения виртуализации, связанные с ними облачные сервисы и подходы к их виртуализации.

- *Xen*—Решение с открытым исходным кодом, использующее технологию паравиртуализации (Barham et al. 2003). Чтобы использовать Xen, необходимо изменить основные коды операционной системы физического компьютера. Следовательно, Xen подходит для ОС с открытым исходным кодом, таких как BSD, Linux и Solaris, но не подходит для проприетарных ОС, таких как Windows, поскольку их кодовая база защищена. Xen широко используется провайдерами хостинга Linux, такими как Amazon EC2, GoGrid, 21Vianet CloudEx и Rackspace Mosso.
- *KVM* (Виртуальная машина на основе ядра) - полное решение виртуализации для Linux на оборудовании x86, содержащее расширения виртуализации.<sup>1</sup> Многие облачные решения (например, Nimbus, CloudStack и OpenNebula) поддерживают виртуализацию Xen и KVM.
- *VirtualBox*—Еще одно решение с открытым исходным кодом в соответствии с условиями Стандартной общественной лицензии GNU (GPL) версии<sup>2</sup>. Данный продукт для виртуализации x86 и AMD64 / Intel64 уровня предприятия. VirtualBox работает на хостах Windows, Linux, Macintosh и Solaris и поддерживает большое количество гостевых ОС, включая, помимо прочего, Windows (NT 4.0, 2000, XP, Server 2003, Vista, Windows7), Linux (2.4 и 2.6) и OpenBSD.<sup>1</sup>

Таблица 3.1 Решения виртуализации и связанные с ними облачные сервисы

Гипервизор	Облачные сервисы	Подход виртуализации	Тип ПО
Xen	Amazon EC2, GoGrid, 21Vianet, CloudEx, RackSpace Mosso	Паравиртуализация	с открытым исходным кодом
KVM	CloudStack,	Полная виртуализация	с открытым исходным кодом
Nimbus			
VirtualBox	VirtualBo	Полная виртуализация	с открытым исходным кодом
x			
SmartOS	Joyent	Сочетание аппаратного обеспечения и виртуализации ОС	с открытым исходным кодом
Virtual PC	Windows	Паравиртуализация	Коммерческая
Azure			
VMware	AT&T Synaptic, Verizon CaaS	Полная виртуализация	Коммерческая

<sup>1</sup> См. KVM на [http://www.linux-kvm.org/page/Main\\_Page](http://www.linux-kvm.org/page/Main_Page).

- *SmartOS*—Специализированная платформа гипервизора с открытым исходным кодом на базе Illumos.<sup>2</sup> Поддерживает два типа виртуализации: виртуальные машины ОС и виртуальные машины KVM. Виртуальные машины ОС - это легкое решение для виртуализации, предлагающее хорошую производительность и все функции, которые имеются у Illumos. Виртуальные машины KVM - это решение для полной виртуализации для работы различных гостевых ОС, включая Linux, Windows, BSD и др.<sup>3</sup>
- *VMware*<sup>4</sup>—Коммерческий продукт, использующий решение полной виртуализации. VMware VMotion может мгновенно перемещать работающую виртуальную машину с одного хоста на другой. Его производительность выше, чем у некоторых других выпусков технологии VMware, и он развернут в производственной среде 70% клиентов VMware и использует полную виртуализацию серверов, хранилищ и сетей. VMware VMotion использует файловую систему кластера VMware для управления доступом к хранилищу виртуальной машины. В рамках VMotion активная память и точное состояние выполнения виртуальной машины быстро передаются по высокоскоростной сети с одного физического сервера на другой, а доступ к дисковой памяти виртуальных машин мгновенно переключается на новый физический хост. Поскольку сеть также виртуализирована с помощью VMware ESX, виртуальная машина сохраняет свою сетевую идентификацию и соединения, обеспечивая плавный процесс миграции.

### 3.4 РАСПРЕДЕЛЕННАЯ ФАЙЛОВАЯ СИСТЕМА

Несмотря на то, что быстрые, хорошо разработанные устройства хранения данных (например, жесткие диски) значительно увеличили объемы хранения данных, а также скорость чтения / записи данных, отказ дискового оборудования может создать единую точку отказа системы, приводящую к потере. Поскольку оборудование хранения не может гарантировать высокую надежность, для многих крупных ИТ-компаний стратегией стало построение надежной распределенной системы хранения данных с группой дешевого оборудования для обеспечения высокой доступности, надежности и масштабируемости для службы хранения данных. Такая система хранения основана на специализированной файловой системе, называемой распределенной файловой системой (DFS).

Облачная служба хранения - важная емкость облачной службы, которая обеспечивает практически неограниченные, защищенные и отказоустойчивые функции хранения и доступа к данным. Многие популярные сервисы облачного хранения, такие как Dropbox, iCloud, Google Drive, SkyDrive и SugarSync, с энтузиазмом используются потребителями в свободном доступе.

<sup>1</sup> См. VirtualBox на <https://www.virtualbox.org/>.

<sup>2</sup> См. Illumos на <http://wiki.illumos.org/display/illumos/illumos+Home>.

<sup>3</sup> См. SmartOS на <http://smartos.org/>.

<sup>4</sup> См. VMware на <http://www.vmware.com/products/datacenter-virtualization/vsphere/vmotion.html>.

Доступность, надежность и масштабируемость, предлагаемые DFS, имеют особое значение для службы облачного хранилища. Следовательно, облачные сервисы используют технологию распределенного хранения данных.

В следующих подразделах представлены основные идеи и характеристики DFS и двух наиболее популярных реализаций DFS, файловой системы Google и распределенной файловой системы Hadoop.

### 3.4.1 Введение в распределенную файловую систему

Файловая система - это абстракция для хранения, извлечения и обновления набора файлов. Управляет доступом к данным и метаданным файлов, а также к доступному пространству устройств, на которых находятся файлы. В зависимости от носителя и целей использования файловые системы можно разделить на дисковые / ленточные файловые системы, распределенные файловые системы и некоторые другие файловые системы специального назначения.<sup>1</sup>

Распределенная файловая система (DFS) - это тип файловой системы, которая обеспечивает доступ к файлам с нескольких хостов через компьютерную сеть (Yeager 2003). Данная система позволяет нескольким машинам совместно использовать файлы и ресурсы хранения.<sup>2</sup> DFS обычно работает поверх локальных файловых систем (файловая система нижнего уровня). Клиентские узлы не имеют прямого доступа к базовому хранилищу блоков, но взаимодействуют по сети с использованием протокола. Протокол может ограничивать доступ к файловой системе в зависимости от списков авторизации доступа или возможностей как на серверах, так и на клиентах, а также от того, как разработан протокол. Напротив, в файловой системе с общим диском все узлы имеют равный доступ к блочному хранилищу, в котором расположена файловая система. В таких системах контроль доступа должен находиться на клиенте. DFS могут включать в себя средства для прозрачной репликации и отказоустойчивости.

С развитием распределенных вычислений и облачной обработки данных многие DFS предлагаются и разрабатываются для доступа к крупномасштабным приложениям и приложениям с большими данными, а также для удовлетворения других требований облачных сервисов, таких как Amazon S3 и Oracle Automatic Storage Management Cluster File System (Oracle ACFS) (Shakian 2011).

### 3.4.2 Файловая система Google

Файловая система Google (GFS) (Ghemawat, Gobioff и Leung 2003) - это масштабируемая распределенная файловая система, разработанная Google для больших распределенных приложений с интенсивным использованием данных. Обеспечивает отказоустойчивость при работе на недорогом стандартном оборудовании.<sup>3</sup>

---

<sup>1</sup> См. File System на [http://en.wikipedia.org/wiki/File\\_system](http://en.wikipedia.org/wiki/File_system).

<sup>2</sup> См. Clustered File System на [http://en.wikipedia.org/wiki/Distributed\\_file\\_system](http://en.wikipedia.org/wiki/Distributed_file_system).

<sup>3</sup> См. Google File System на [http://en.wikipedia.org/wiki/Google\\_File\\_System](http://en.wikipedia.org/wiki/Google_File_System).

Разделяя многие из тех же целей, что и предыдущие распределенные файловые системы, GFS исследовала различные конструкции, наблюдая и анализируя рабочие нагрузки приложений Google и технологическую среду. По сравнению с сетевой файловой системой (NFS) (широко распространенной традиционной DFS для Linux и UNIX, разработанной Sun Microsystems в 1984 г.), основные различия заключаются в следующем (Ghemawat, Gobioff, and Leung 2003):

- *Более крупные фрагменты* - размер фрагмента Google составляет 64 МБ. Данный дизайн больше подходит для больших файлов и обеспечивает лучшую производительность ввода-вывода.
- *Множественная репликация данных* - чтобы гарантировать надежность, GFS обеспечивает как минимум три избыточности данных.
- *Не локальное кэширование* - в отличие от NFS, который предлагает локальное кэширование на стороне клиента, GFS не поддерживает локальное кэширование.

Приняв вышеуказанную схему, GFS может обеспечить высокую совокупную производительность большому количеству клиентов, а также очень высокую масштабируемость, надежность и доступность. GFS широко используется в Google не только в качестве платформы хранения для создания, обработки и архивирования данных, но и для исследований и разработок, требующих больших наборов данных.

### 3.4.3 Распределенная файловая система Apache Hadoop

Распределенная файловая система Hadoop (HDFS) (Borthakur 2007; Shvachko et al. 2010) - распределенная файловая система, разработанная для работы на стандартном оборудовании. HDFS служит одним из важных модулей (а также подпроектом) проекта Apache Hadoop, который представляет собой широко используемую программную среду с открытым исходным кодом для надежных и масштабируемых распределенных вычислений. Многие ИТ-компании, такие, как Yahoo, Intel и IBM, приняли HDFS в качестве технологий хранения больших данных.

Как и GFS, HDFS имеет следующие общие характеристики<sup>1</sup>: (1) Неисправности - это скорее нормальное явление, чем исключение; (2) Большие наборы данных (огромный размер файла, обычно от ГБ до ТБ); (3) Поточковый доступ к данным, например, файл читается последовательно; (4) Добавить для записи; (5) Сотни одновременных доступов на запись; (6) Пропускная способность важнее задержки, и (7) перемещение вычислений дешевле, чем перемещение данных. Это особенно верно, когда размер набора данных большой. Таким образом, HDFS отличается высокой отказоустойчивостью и предназначена для развертывания на недорогом оборудовании. HDFS обеспечивает высокопроизводительный доступ к данным приложения и подходит для приложений с большими наборами данных. Он ослабляет некоторые требования POSIX для обеспечения потокового доступа к данным файловой системы.

И GFS, и HDFS используют архитектуру ведущий / ведомый. Главный сервер (NameNode) управляет пространством имен файловой системы и регулирует доступ клиентов к файлам.

<sup>1</sup>См. Hadoop на [http://hadoop.apache.org/docs/stable/hdfs\\_design.html](http://hadoop.apache.org/docs/stable/hdfs_design.html).

Несколько ведомых устройств (DataNodes), обычно по одному на узел в кластере, управляют хранилищем, подключенным к узлам, на которых они работают. Узлы данных отвечают за обслуживание запросов на чтение и запись от клиентов файловой системы. Узлы данных также выполняют создание, удаление и репликацию блоков по команде от узла NameNode.

Передовые технологии распределенных файловых систем могут обеспечить высокую доступность, надежность и удовлетворить требования крупномасштабного и высокопроизводительного одновременного доступа. Это стало основой облачной обработки данных для хранения и управления большими данными, а также используется в сочетании с платформами параллельной обработки данных, такими как Map Reduce.

### 3.5 WeB x.0

Всемирная паутина (WWW или Интернет) - данная модель обмена информацией, которая соединяет распределенные, множественные источники и разнородную информацию с помощью транспортного протокола гипертекста (HTTP). WWW предоставляет массивную систему, позволяющую пользователям сети получать доступ к огромным объемам информации через сетевую инфраструктуру - Интернет.

- *Web 1.0*—Первое поколение появилось еще в 1993 году, когда гиперссылки между веб-страницами начались с публикации WWW для широкой публики<sup>1</sup>. Основное внимание на данном этапе было сосредоточено на создании Интернета и обеспечении публичного доступа и коммерческого использования. Веб-страницы были статичными, и потребители могли только просматривать их, а не вносить свой вклад в содержание.<sup>2</sup>
- *Web 2.0*—Выражение «Web 2.0» впервые было введено в обращение в 2004 году компанией O'Reilly Media<sup>3</sup>, обозначая второе поколение Интернета, позволяющее потребителям взаимодействовать, сотрудничать и координировать свои действия друг с другом. В отличие от Web 1.0, Web 2.0 позволял пользователям делать больше, чем просто просматривать Web-страницы, предоставляя динамически редактируемые Web-страницы, такие, как блоги, вики-сайты и различные Web-приложения, через Web-браузеры. Такая возможность отличает Web 2.0 от общей модели обмена информацией; он превратился в новую вычислительную парадигму: вычисления «сеть как платформа»<sup>4</sup>, которые служат одной из фундаментальных технологий для облачной обработки данных.
- *Web 3.0*—Это третье поколение Интернета, стремящееся сделать Интернет более интеллектуальным за счет адаптации технологий семантического Интернета, интеллектуального анализа данных, машинного обучения и искусственного интеллекта. Поэтому Web 3.0 также называют Intelligence Web.

<sup>1</sup> См. Tim Berners-Lee на [http://en.wikipedia.org/wiki/Tim\\_Berners-Lee](http://en.wikipedia.org/wiki/Tim_Berners-Lee).

<sup>2</sup> См. Web 1.0 на [http://en.wikipedia.org/wiki/Web\\_1.0#cite\\_note-3](http://en.wikipedia.org/wiki/Web_1.0#cite_note-3).

<sup>3</sup> См. Paul Graham на <http://www.paulgraham.com/Web20.html>.

<sup>4</sup> См. O'Reilly на <http://oreilly.com/Web2/archive/what-is-Web-20.html>.

<sup>1</sup> На момент написания данной статьи реальной службы на основе Web 3.0 еще не было. Вот пример, дающий основное представление о том, как выглядит сервис Web 3.0. Если кто-то хочет пойти поужинать и посмотреть фильм «Аватар», с помощью Web 2.0 ему / ей, возможно, придется поискать фильм, чтобы найти театр рядом с его / ее местоположением, а затем поискать рестораны рядом с театром. С Web 3.0 ему / ей просто нужно выполнить поиск «Аватар, фильм и ужин», а поисковая система выполнит всю оставшуюся работу в среде облачных вычислений и предоставит список вариантов с учетом местоположения театра и ресторана, отсортированных по доступным временным интервалам и ценам.

- *Web 4.0*—Это все еще развивающийся концепт без четкого и точного определения. Обычно считается, что в эпоху Web 4.0 компьютеры будут намного умнее благодаря более интеллектуальным технологиям взаимодействия человека и компьютера, таким как интерфейсы, управляемые разумом (Aghaei, Nematbakhsh, and Farsani 2012). Например, если кто-то хочет найти информацию о сегодняшней погоде, ему / ей может потребоваться только сесть перед компьютером и подумать о желаемой информации, и служба на основе Web 4.0 будет читать мысли человека и реагировать.

В следующих подразделах представлены несколько важных технологий / характеристик, связанных с Web x.0, которые служат ключевыми технологиями для облачной обработки данных.

### 3.5.1 Веб-сервисы

Веб-сервис - это программная система, предназначенная для поддержки межмашинного взаимодействия по сети.<sup>2</sup> Язык описания веб-сервиса (WSDL) - это машиночитаемый формат, разработанный для описания интерфейса веб-сервиса на основе расширяемого языка разметки (XML). Простой протокол доступа к объектам (SOAP) - это спецификация протокола, которая позволяет другим системам взаимодействовать с веб-сервисами посредством обмена структурированной информацией через Интернет с использованием HTTP. Несмотря на то, что WSDL и SOAP разработаны для описания поведения стандартного веб-сервиса, они не являются обязательными. За последние несколько лет многие приложения Web 2.0 перешли с веб-сервисов на основе SOAP на коммуникации на основе REST (передача репрезентативного состояния) (Benslimane, Dustdar and Sheth 2008), которые стали преобладающей моделью проектирования веб-сервисов. Службы на основе REST позволяют потребителям получать информацию с помощью простых методов HTTP, таких, как GET, POST, PUT и DELETE.

<sup>1</sup> См. Lifeboat Foundation на <http://lifeboat.com/ex/Web.3.0>.

<sup>2</sup> См. W3C на <http://www.w3.org>.

Общие веб-интерфейсы API (интерфейсы прикладного программирования), такие как Google API и Yahoo API, являются хорошими примерами служб на основе REST. В геопространственной области стандарты службы веб-картографии (WMS), службы веб-покрытия (WCS) и службы каталогов для Интернета (CSW), разработанные Открытым геопространственным консорциумом (OGC) <sup>1</sup>, также основаны на REST.

Сеть предназначена для улучшения доступности информации, и широкое использование веб-сервисов, как на основе SOAP, так и на основе REST, значительно расширило такую возможность. Кроме того, веб-сервисы предоставляют стандартные средства взаимодействия между различными программными приложениями, работающими на различных платформах и / или мейнфреймах <sup>2</sup>, что значительно улучшает взаимодействие информации через Интернет.

Важной характеристикой облачной обработки данных являются услуги через Интернет, представленные архитектурой X как услуга (XaaS). Веб-сервисы, строительные блоки для различных веб-приложений, играют важную роль в эволюции облачной обработки данных, особенно на уровнях SaaS и DaaS.

### 3.5.2 Сервис-Ориентированная Архитектура

В программной инженерии сервис-ориентированная архитектура (SOA) - это компонентная модель, основанная на сервисах, включающая набор принципов и методологий для проектирования и разработки программного обеспечения в форме взаимодействующих сервисов.<sup>3</sup> Система на основе SOA состоит из множества слабо связанных сервисов. Каждая услуга является функциональной единицей системы и определяется сервисным интерфейсом. Данное определение интерфейса не зависит от языков программирования, операционных систем и оборудования. Реализованные службы публикуются в сети (например, в Интернете), в которой другие разработчики могут получать доступ к ним и повторно использовать их в других приложениях. Три основных преимущества использования SOA - это повторное использование компонентов, интеграция существующей системы и независимость от языка / платформ.

- *Повторное использование компонентов* - Поскольку каждый компонент определяется и реализуется как веб-сервис, такие сервисы могут быть опубликованы и доступны другим пользователям через Интернет. Для дальнейшего улучшения возможности повторного использования разработан набор стандартов веб-сервисов, таких, как REST, SOAP и WSDL.
- *Существующая системная интеграция* - SOA может использоваться для интеграции существующих систем путем определения веб-интерфейса системных функций и предоставления их предприятию стандартным согласованным способом (Geoffrey 2009).

<sup>1</sup> См. OGC на <http://www.opengeospatial.org/>.

<sup>2</sup> См. Web Services Architecture at <http://www.w3.org/TR/ws-arch/>.

<sup>3</sup> См. Service-Oriented Architecture на [http://en.wikipedia.org/wiki/Service-oriented\\_architecture](http://en.wikipedia.org/wiki/Service-oriented_architecture).

Данные услуги реализуются путем подключения к существующим функциям и доступны для создания новой системы. В таком сценарии SOA выступает в качестве связующего звена приложения для объединения существующих функций.

- *Независимость от языка и платформы* - Большинство современных стандартов веб-сервисов основаны на XML (eXtensible Markup Language), независимом от платформы языке разметки, который удобен для чтения человеком и компьютером. Поведение веб-сервиса при вводе, выводе и обработке четко определено стандартами на основе XML, такими как WSDL, обеспечивая общий независимый от языка механизм для реализации и использования таких служб. Когда данные службы публикуются через SOAP в Интернете, к ним можно получить доступ через любой компьютер, подключенный к Интернету, который не зависит от платформы.

Облачная обработка данных в значительной степени использует концепцию SOA, особенно на уровнях SaaS и PaaS. Хотя есть некоторые совпадения между концепциями облачной обработки данных и SOA, они имеют разные акценты. SOA - архитектура, ориентированная на ответ на вопрос «как разрабатывать приложения», в то время как облачная обработка данных - это инфраструктура, подчеркивающая решение «как доставлять приложения».<sup>1</sup> На рис. 3.2 показано, как технологии Web и SOA помогли эволюции облачной обработки данных.

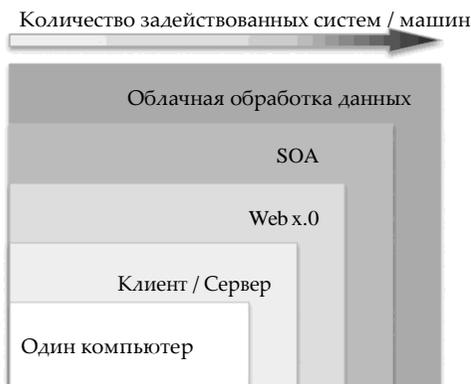


Рисунок 3.2 Эволюция облачной обработки данных.

<sup>1</sup> Системная инженерия в MITRE на [http://www.mitre.org/work/tech\\_papers/tech\\_papers\\_09/09\\_0743/09\\_0743.pdf](http://www.mitre.org/work/tech_papers/tech_papers_09/09_0743/09_0743.pdf).

### 3.6 ЗАКЛЮЧЕНИЕ

Возникновение, развитие и готовность облачной обработки данных обусловлены и становятся возможными благодаря множеству технологий, обсуждаемых в данной главе. Многоядерные и специализированные многоядерные технологии помогают снизить нагрузку на обработку и позволяют поставщику облачных услуг создавать энергоэффективные и высокопроизводительные центры обработки данных. Быстро развивающиеся технологии жестких дисков решают проблемы огромных и постоянно растущих требований к хранению и облачной обработке данных. Сетевая технология является фундаментальной основой облачной обработки данных, поскольку она соединяет все облачные компоненты вместе и предоставляет облачные сервисы потребителям. Данные три аппаратных улучшения значительно расширили возможности развития облачной инфраструктуры.

Распределенные вычисления служат предшествующей технологией, которая позволила зародиться облачной обработке данных. Некоторые важные концепции распределенных вычислений, такие как интеграция существующих ИТ-ресурсов, широко применяются в облачной обработке данных. Технология виртуализации абстрагируется от реализации облачных платформ, инкапсулируя детали оборудования, что позволяет создавать динамическое количество виртуальных машин в зависимости от потребностей приложения. Виртуализация играет наиболее важную роль при реализации платформ облачной обработки данных, особенно для модели IaaS.

Распределенная файловая система - это ядро технологии распределенного хранения данных, которая является фундаментальной моделью хранения в облачной обработке данных. Модель распределенного хранилища не только обеспечивает практически неограниченные, безопасные и отказоустойчивые функции хранения и доступа к данным, но также позволяет крупным ИТ-компаниям создавать надежные системы с дешевым оборудованием. Благодаря таким преимуществам распределенная файловая система играет важную роль в развитии облачных сервисов, особенно облачного хранилища.

Связанные с Интернетом технологии играют решающую роль в развитии облачной обработки данных. Веб-сервисы действуют как строительные блоки для облачной платформы, особенно на уровнях SaaS и DaaS. SOA используется для руководства внедрением платформ облачной обработки данных. Технология Web-сервисов и сервис-ориентированная концепция SOA делают возможным сервис / приложение по запросу. Наконец, Интернет предоставляет возможность беспрепятственно сотрудничать, интегрировать, представлять и обмениваться информацией через Интернет в различных сообществах.

### 3.7 КРАТКОЕ ИЗЛОЖЕНИЕ

Данная глава знакомит с технологиями, обеспечивающими облачную обработку данных, включая усовершенствования оборудования (Раздел 3.1), вычислительные технологии (раздел 3.2), виртуализацию (Раздел 3.3), распределенную файловую систему (Раздел 3.4) и Web x.0 (Раздел 3.5). Заключительное обсуждение приведено в Разделе 3.6.

### 3.8 ПРОБЛЕМЫ

1. Перечислите ключевые обеспечивающие технологии облачной обработки данных с точки зрения аппаратного обеспечения, вычислений, файловой системы и Интернета.
2. Почему многоядерные технологии могут помочь поставщикам облачных услуг создавать энергоэффективные центры обработки данных?
3. Почему интеллектуальные устройства (например, смартфоны и планшеты) могут способствовать популяризации облачных сервисов? Вы использовали свое устройство (а) для доступа к облачным сервисам?
4. В чем заключаются основные различия между кластерными вычислениями, грид-вычислениями, служебными вычислениями и облачной обработкой данных?
5. Почему вычислительные сети важны для облачной обработки данных?
6. Что такое виртуализация и как она помогает реализовать облачную обработку данных? Перечислите четыре решения виртуализации.
7. Каковы основные различия между распределенной файловой системой (DFS) и традиционной файловой системой? Приведите, пожалуйста, два примера DFS.
8. Перечислите поколения Web.
9. Каковы характеристики SOA и какое отношение SOA имеет к облачной обработке данных?
10. Подумайте о технологии и обсудите, почему, по вашему мнению, это важно для облачной обработки данных.

### СПИСОК ЛИТЕРАТУРЫ

- Aghaei, S., M. A. Nematbakhsh, and H. K. Farsani. 2012. Evolution of the World Wide Web: From Web 1.0 to Web 4.0. *International Journal of Web & Semantic Technology (IJ West)* no. 3: 1–10.
- Barham, P., B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, ... and A. Warfield. 2003. Xen and the art of virtualization. *ACM SIGOPS Operating Systems Review* 37, no. 5: 164–177.
- Benslimane, D., S. Dustdar, and A. Sheth. 2008. Services mashups: The new generation of Web applications. *Internet Computing, IEEE* 12, no. 5: 13–15.
- Borthakur, D. 2007. The Hadoop Distributed File System: Architecture and design. *Hadoop Project Web Site* no. 11: 21.
- Chai, L., Q. Gao, and D. K. Panda. 2007. Understanding the impact of multicore architecture in cluster computing: A case study with Intel dual-core system. In *Cluster Computing and the Grid. CCGRID 2007. 7th IEEE International Symposium*, pp. 471–478. IEEE.
- Foster, I., Y. Zhao, I. Raicu, and S. Lu. 2008. Cloud computing and grid computing 360-degree compared. In *Grid Computing Environments Workshop. GCE'08*, pp. 1–10. IEEE.
- Geoffrey, R. 2009. Cloud Computing and SOA, Systems Engineering at MITRE, SOA Series. [http://www.mitre.org/work/tech\\_papers/tech\\_papers\\_09/09\\_0743/09\\_0743.pdf](http://www.mitre.org/work/tech_papers/tech_papers_09/09_0743/09_0743.pdf) (accessed April 23, 2013).

- Ghemawat, S., H. Gobioff, and S. Leung. 2003. The Google File System. *ACM SIGOPS Operating Systems Review* 37, no. 5: 29–43. ACM.
- Liu, H. and D. Orban. 2008 (May). Gridbatch: Cloud computing for large-scale data-intensive batch applications. In *Cluster Computing and the Grid, 2008. CCGRID'08. 8th IEEE International Symposium*, pp. 295–305. IEEE.
- McEvoy, G.V. and B. Schulze. 2008. Using clouds to address grid limitations. In: *Proceedings of the 6th International Workshop on Middleware for Grid Computing* no. 11. Leuven, Belgium.
- Mell, P. and T. Grance. 2011. The NIST definition of cloud computing. *NIST Special Publication* 800: 145.
- Pase, D. M. and M. A. Eckl. 2005. *A Comparison of Single-Core and Dual-Core Opteron Processor Performance for HPC*. IBM xSeries Performance Development and Analysis. Research Triangle Park, NC: IBM.
- Sagona, P. 2009. A preliminary performance analysis of medical image registration on single-core and multicore clusters. *Lecture Notes in Computer Science*. doi 10.1.1.130.1367.
- Sahoo, J., S. Mohapatra, and R. Lath. 2010. Virtualization: A survey on concepts, taxonomy and associated security issues. In *Computer and Network Technology (ICCNT), 2nd International Conference*, pp. 222–226. IEEE.
- Shakian, A. 2011. Oracle Cloud File System. Oracle White Paper. <http://www.oracle.com/technetwork/products/cloud-storage/cloudfs-overview-wp-279856.pdf?ssSourceSiteId=ocomen> (accessed March 12, 2013).
- Shvachko, K., H. Kuang, S. Radia, and R. Chansler. 2010. The Hadoop Distributed File System. In *Mass Storage Systems and Technologies (MSST), IEEE 26th Symposium*, pp. 1–10. IEEE.
- Tograph, B. and Y.R. Morgens. 2008. Cloud computing. *Communications of the ACM* 51, no. 7.
- Vaquero, L. M., L. Rodero-Merino, J. Caceres, and M. Lindner. 2008. A break in the clouds: Towards a cloud definition. *ACM SIGCOMM Computer Communication Review* 39, no. 1: 50–55.
- Yeager, P. S. 2003. A distributed file system for distributed conferencing system. Ph.D. diss., Florida: University of Florida.
- Youseff, L., M. Butrico, and D. D. Silva. 2008. Toward a unified ontology of cloud computing. *Grid Computing Environments Workshop. GCE'08*. IEEE.

## Часть II

---

# Внедрение приложений на цифровую платформу в облачных сервисах

---

В данной части постепенно вводятся общие процедуры развертывания приложений геонаучных исследований в облачных сервисах. Во-первых, проиллюстрированы шаги по развертыванию простого веб-приложения в облачных сервисах, чтобы читатели могли узнать, как использовать облачную обработку данных (Глава 4). Затем демонстрируются приложение, управляемое базой данных, и приложение для высокопроизводительных вычислений (HPC), позволяющее использовать облачные приложения для практических геонаучных исследований (Глава 5). Наконец, стратегии выбора облачных сервисов вводятся и обсуждаются на основе критериев оценки облачных сервисов и моделей затрат (Глава 6).



## Глава 4

---

# Как применять облачную обработку данных

*Кай Лю, Цюньбин Хуан, Цзичжэ Ся,  
Чжэньлун Ли и Питер Лостритто*

---

В данной главе на примере простого веб-приложения демонстрируются основные этапы развертывания приложений в облачных сервисах.

### 4.1 ПОПУЛЯРНЫЕ ОБЛАЧНЫЕ СЕРВИСЫ

#### 4.1.1 Введение

Облачные сервисы предоставляют потребителям вычислительные ресурсы (хранилище, сеть и вычисления) в виртуализированном виде. Конечный успех облачного сервиса определяется поддерживаемыми приложениями и улучшениями, внесенными в приложения по сравнению с традиционной вычислительной службой (Dastjerdi, Garg, and Vuууа, 2011). Улучшения обычно могут быть достигнуты в производительности приложений, экономии затрат, доступности, стабильности, безопасности, подотчетности и других аспектах, требуемых конкретными приложениями (Emeakaroha et al. 2011). Хорошо продуманный процесс миграции или развертывания приложений в облачном сервисе имеет решающее значение для максимизации улучшений (Afgan et al. 2011).

Процессы развертывания различаются в зависимости от выбранного облачного сервиса и сложности приложения (Emeakaroha et al. 2011). Например, «Инфраструктура как услуга» (IaaS) и «Платформа как услуга» (PaaS) могут поддерживать разные типы приложений, и этапы развертывания таких приложений часто отличаются. Различные приложения также могут иметь разные требования к операционной системе и конфигурациям. Например, простому веб-приложению может потребоваться только веб-сервер, но сложное приложение, такое как система финансового управления, может включать в себя базу данных, финансовый анализ, инструменты управления, серверные и клиентские программные модули, а также компоненты резервного копирования данных. Чтобы определить лучший процесс развертывания, потребители часто начинают с анализа требований приложений и доступных облачных сервисов. Затем рабочий процесс развертывания может быть разработан в соответствии с конкретными требованиями к облачным сервисам и приложениям (Afgan et al. 2011).

В данной главе демонстрируются основные шаги по развертыванию приложений в двух популярных облачных сервисах: Amazon Web Services (AWS) и Windows Azure.

### 4.1.2 amazon aWs и Windows azure

Amazon AWS и Windows Azure становятся все более популярными, о чем свидетельствуют результаты анализа Google Insights for Search (рис. 4.1). Количество поисковых запросов по двум терминам Amazon AWS и Windows Azure постоянно увеличивалось с конца 2008 г. по 2013 г.

Как первый успешный и крупнейший поставщик облачных услуг в мире, Amazon предоставляет наиболее полный набор облачных услуг и удобный графический пользовательский интерфейс (GUI). Данный сервис широко используется многими организациями для размещения своих облачных приложений. Например, Amazon стал первым поставщиком облачных услуг, прошедшим сертификацию среднего уровня Федеральным законом об управлении информационной безопасностью (FISMA), и предоставляет Управление государственных услуг (GSA) услуги правительственным агентствам США. Преимущество Windows Azure заключается в предоставлении как операционных систем Windows, так и платформы разработки Visual Studio для обеспечения PaaS.

- **AWS**—Так называется набор веб-сервисов, предоставляемых Amazon Inc. Он работает с 2006 года и обслуживает сотни тысяч клиентов по всему миру. AWS - это комплексная платформа, которая предлагает список облачных сервисов, таких как вычисления, хранение, доставка контента, база данных, развертывание и управление, сетевые и другие сервисы (Varia and Mathew 2012). В данной книге используются следующие AWS: Amazon Elastic Compute Cloud (Amazon EC2, сервис, обеспечивающий изменяемые вычислительные мощности), Elastic Block Store (EBS) и Amazon Simple Storage Service (Amazon S3, интернет-сервис хранения).

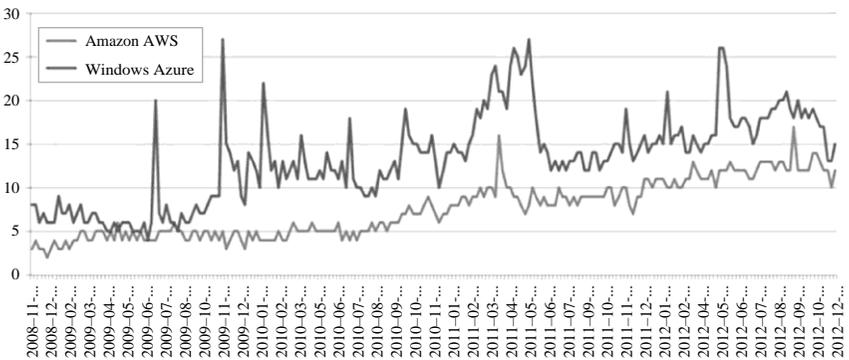


Рисунок 4.1 (См. цветную вставку.) Тенденции поиска среди Amazon Web Services (AWS) и Windows Azure.

- *Windows Azure*—Облачный сервис Microsoft, находящийся в открытом доступе. Он состоит из четырех частей, включая Windows Azure, SQL Azure, Windows Azure AppFabric и Windows Azure Marketplace (Chappell 2010). Windows Azure - это среда Windows для развертывания приложений и хранения данных на компьютерах в центрах обработки данных Microsoft; SQL Azure - это облачная служба реляционной базы данных; Windows Azure AppFabric - это облачная инфраструктурная служба для приложений, работающих в облаке или локально; и Windows Azure Marketplace - это онлайн-сервис для покупки облачных данных и приложений. Windows Azure также предоставляет три типа служб выполнения: виртуальные машины (ВМ), службу веб-сайта и облачные сервисы. Возможность создавать виртуальную машину по запросу, будь то из стандартного образа или настроенного образа, может быть использована для создания виртуальной машины Windows или Linux для развертывания приложений. Служба веб-сайта предлагает управляемую веб-среду с использованием информационных служб Интернета (IIS). Пользователи облака могут переместить существующий веб-сайт IIS на веб-сайты Windows Azure, не запуская виртуальную машину.

## 4.2 МОДЕЛЬ ВНЕДРЕНИЯ: ПРОСТОЕ ВЕБ-ПРИЛОЖЕНИЕ

Распространенным типом приложения в Интернете является веб-сайт. Веб-приложения широко используются для публикации геонаучных приложений для распределенных пользователей. HTML - это основа веб-приложения. В настоящем разделе демонстрируется развертывание веб-приложения Hello Cloud на Amazon AWS и Windows Azure.

### 4.2.1 HTML-дизайн для веб-приложения Hello Cloud

Приложение Hello Cloud представляет собой простой файл index.html со ссылками на официальные страницы шести облачных сервисов или решений: Amazon AWS, Windows Azure, Apache CloudStack, Eucalyptus Cloud, Nimbus и OpenNebula (рисунок 4.2).

Содержимое HTML-файла показано ниже.

```
<!DOCTYPE html>  
<html>  
<body>  
<h1>Hello Cloud</h1>
```



## Hello Cloud

- [Amazon AWS](#)
- [Windows Azure](#)
- [Apache Cloudstack](#)
- [Eucalyptus Cloud](#)
- [Nimbus](#)
- [OpenNebula](#)

Рисунок 4.2 Домашняя страница веб-сайта Hello Cloud.

```
<ul>
<li><a href="http://aws.amazon.com/">Amazon AWS</a></li>
<li><a href="http://www.windowsazure.com/en-us/">Windows
  Azure</a></li>
<li><a href="http://incubator.apache.org/cloudstack/">Apache
  Cloudstack</a></li>
<li><a href="http://www.eucalyptus.com/">Eucalyptus Cloud
  </a></li>
<li><a href="http://www.nimbusproject.org/">Nimbus</a></li>
<li><a href="http://openebula.org/">OpenNebula</a></li>
</ul>
</body>
</html>
```

### 4.2.2 Веб-серверы

Веб-сервер необходим для доставки веб-приложения в браузеры. В данной книге используются три типа веб-серверов: HTTP-сервер Apache (httpd), сервер Apache Tomcat и Microsoft Internet Information Services (IIS). HTTP-сервер Apache - один из самых популярных веб-серверов в Интернете<sup>1</sup>. Сервер Apache Tomcat - это программное обеспечение с открытым исходным кодом, в котором реализованы технологии Java Servlet и JavaServer Pages. IIS - еще один популярный веб-сервер, разработанный Microsoft для Windows. IIS 7.5 поддерживает HTTP, HTTPS, FTP, FTPS, SMTP и NNTP. Раздел 4.3 описывает, как перенести приложение Hello World Web на HTTP-сервер Apache и Microsoft IIS, работающие в облачном сервисе.

<sup>1</sup>См. Apache на <http://httpd.apache.org/>.

## 4.3 РАЗВЕРТЫВАНИЕ ВЕБ-ПРИЛОЖЕНИЯ НА ОБЛАЧНЫХ СЕРВИСАХ

### 4.3.1 Веб-сервисы Amazon

С помощью Консоли управления AWS<sup>1</sup> или Amazon EC2 AMI Tools два пользователя облака могут запросить запуск экземпляра на основе предварительно определенного образа машины Amazon (AMI). Экземпляр можно использовать как серверный компьютер, чтобы пользователи могли развертывать свои приложения.

На рис. 4.3 показано, как развернуть приложение Hello Cloud на платформе Amazon EC2. Подробности каждого шага следующие:

*Шаг 1. Зарегистрируйтесь в Amazon AWS* - Для входа в Amazon EC2 требуется учетная запись AWS. Чтобы создать учетную запись AWS, пользователи облака могут перейти на веб-страницу AWS<sup>2</sup> и щелкнуть «Зарегистрироваться», после чего потребители будут перенаправлены к мастеру создания учетной записи. После создания учетной записи AWS ее можно использовать для доступа к AWS (рисунок 4.4)

*Шаг 2. Авторизация доступа к сети* - данный шаг используется для: (1) включения доступа пользователей к экземпляру с помощью SSH (Secure Shell) или RDP (удаленный рабочий стол); и (2) разрешить экземпляру принимать веб-трафик через определенный порт. Консоль управления AWS, простой веб-интерфейс, можно использовать для авторизации доступа к сети. Пользователи облака могут получить доступ к консоли, (1) выбрав Консоль управления AWS после входа в AWS (рисунок 4.5) и (2) выбрав EC2 в категории Compute & Networking, чтобы войти в панель управления EC2.



Рисунок 4.3 Процесс развертывания Hello Cloud на Amazon EC2.

<sup>1</sup> См. AWS на <http://aws.amazon.com>.

<sup>2</sup> См. AWS на <http://aws.amazon.com/ec2/>.

<sup>3</sup> См. AWS на <http://aws.amazon.com/>.

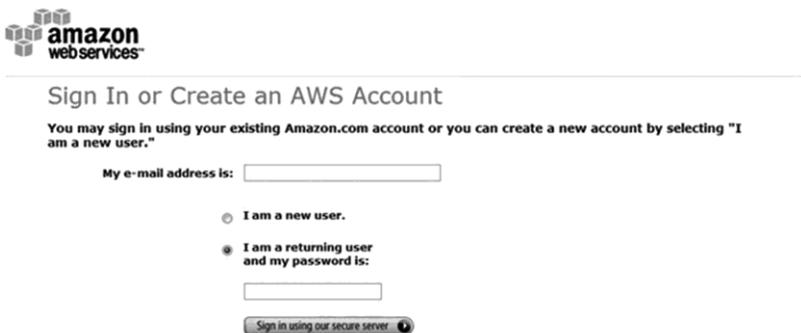
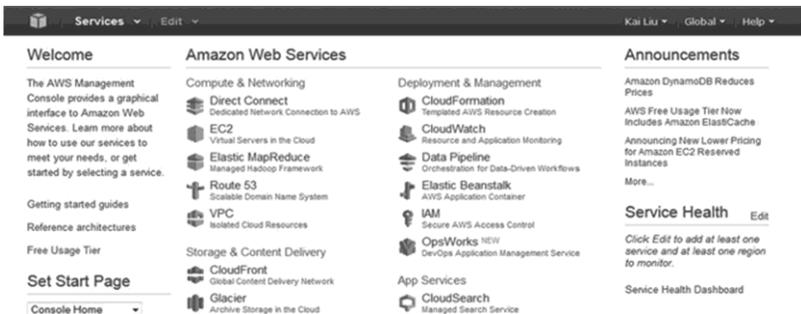


Рисунок 4.4. Войдите в систему или создайте учетную запись AWS.



(a)



(b)

Рисунок 4.5 Консоль управления AWS. (a) Кнопка Консоли управления AWS; (b) Интерфейс Консоли управления AWS.

Группы безопасности, расположенные на панели навигации панели управления EC2 (рис. 4.6), действуют как брандмауэр, контролирующий трафик, которому разрешен доступ к экземпляру. Потребители могут выбрать группы безопасности, а затем создать группу безопасности.

На рисунке 4.7 показано, как создать новую группу безопасности для авторизации доступа к сети для приложения Hello Cloud. Для приложения требуются SSH и HTTP-доступ, поэтому потребителям необходимо добавить их соответственно.

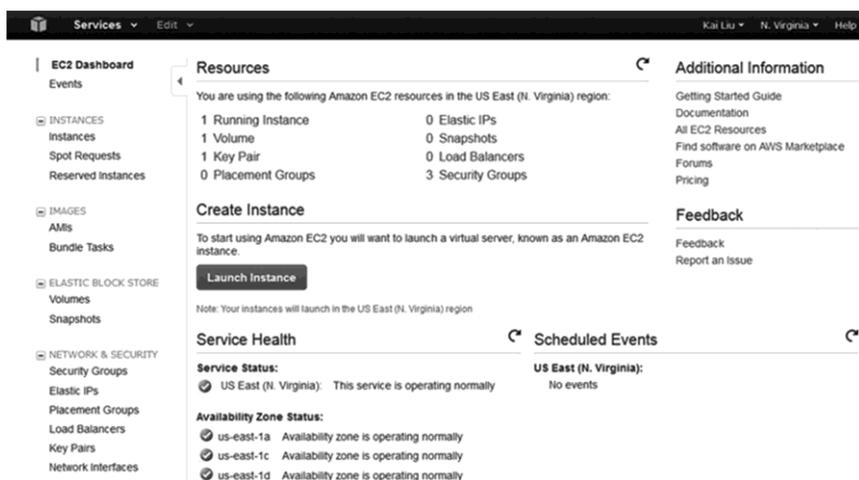


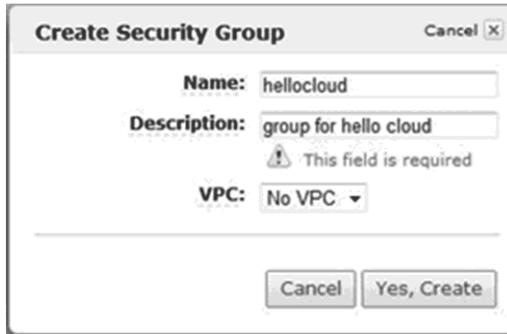
Рисунок 4.6 Интерфейс приборной панели EC2. Меню «Группы безопасности» находится на левой части панели навигации.

На рисунке 4.7b показано, как создать новое правило для SSH. Потребители должны выбрать SSH в раскрывающемся списке и ввести общедоступный IP-адрес в нотации CIDR (бесклассовая междомменная маршрутизация) в качестве источника. В этом случае включен SSH из 129.174.63.89. На рис. 4.7c показано, как создать новое правило для HTTP. Источник 0.0.0.0/0 разрешает доступ по протоколу HTTP из любого места. Потребители должны нажать Применить изменения правила после добавления правил.

*Шаг 3. Запуск экземпляра - Key Pairs* требуется для запуска экземпляра и будет использоваться для входа в экземпляр после запуска. Потребители должны перейти на страницу Key Pairs на панели инструментов EC2, чтобы создать пару ключей, а затем сохранить данную пару ключей на локальном компьютере (рисунок 4.8)

Для размещения приложения Hello Cloud выбран общедоступный AMI с CentOS 6.3. Пользователи облака могут нажать кнопку «Запустить экземпляр» на панели управления EC2 (рис. 4.6), чтобы запустить экземпляр. После нажатия кнопки пользователи будут направлены к мастеру, чтобы выбрать «Запустить опцию» для запуска нового экземпляра. Пользователи могут напрямую искать доступный CentOS AMI с предустановленной ОС, щелкнув вкладку AWS Marketplace и введя «CentOS» в качестве ключевого слова (рисунок 4.9).

На рисунке 4.10 показаны результаты поиска на торговой площадке AWS. В нем перечислены краткие сведения о каждом результате, такие как имя AMI, версия, продавец, цена и описание. Первый элемент - это официальный образ CentOS 6.3 x86\_64. Пользователи могут выбрать данный AMI, щелкнув его заголовок, а затем нажав «Продолжить» на следующей странице.



(a)



(b)



(c)

Рисунок 4.7 Создать группу безопасности. (a) Создайте группу «hellocloud»; (b) Создайте новое правило для SSH; (c) Создайте новое правило для HTTP.



Рисунок 4.8 Создайте пару ключей на странице «Key Pairs».

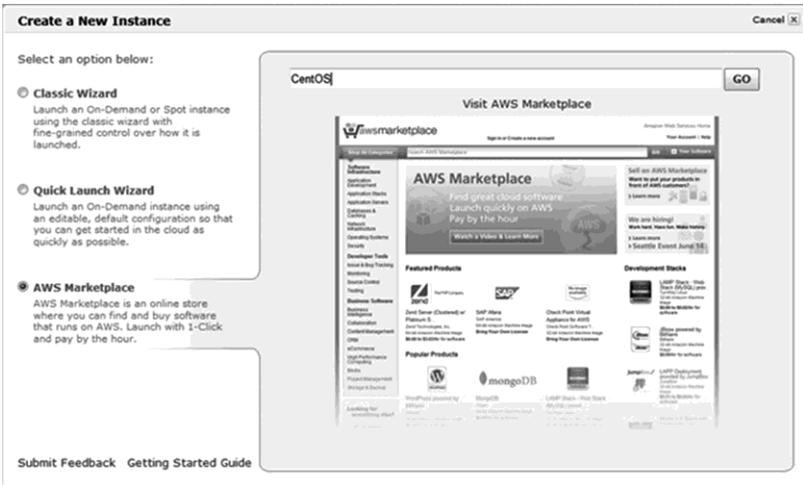


Рисунок 4.9 Запуск нового экземпляра путем поиска и использования AMI из AWS Marketplace.

После выбора AMI пользователи облака могут выбрать конфигурации для экземпляра (такие, как регион, тип экземпляра, настройки брандмауэра и пара ключей), а затем запустить экземпляр одним щелчком мыши (рисунок 4.11). Группа безопасности, созданная на шаге 2, должна использоваться в качестве настроек брандмауэра, а пара ключей, созданная на шаге 3, должна использоваться как key pair. Если потребители предпочитают больше возможностей, они могут щелкнуть Запуск с вкладкой Консоль EC2, чтобы запускать продукты Marketplace через Консоль EC2.



Рисунок 4.10 Результаты поиска CentOS в AWS Marketplace.

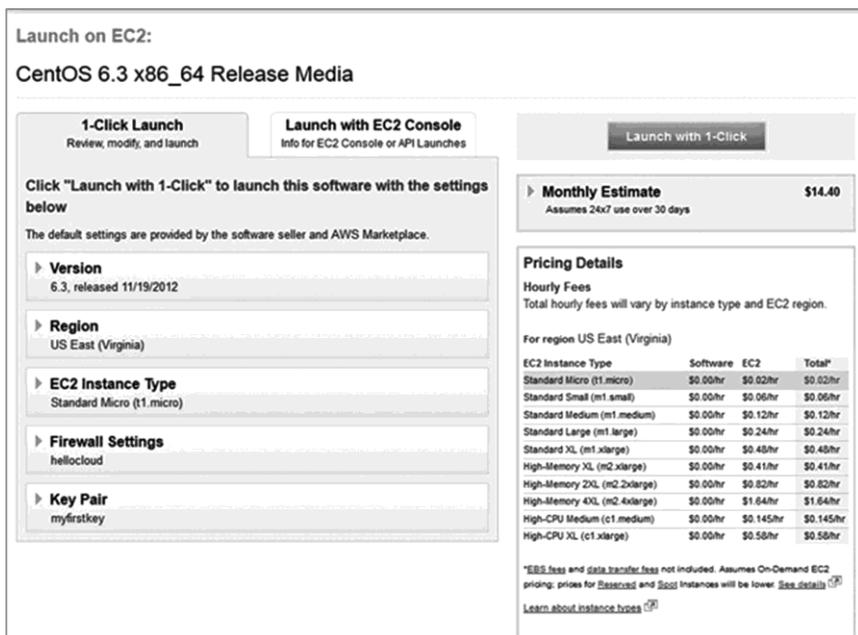


Рисунок 4.11 Запускаем экземпляр одним щелчком мыши.

*Шаг 4. Вход в экземпляр* - после запуска экземпляра пользователи осуществляют вход в экземпляр. После запуска экземпляра пользователи могут войти в него, чтобы получить полный корневой доступ с помощью метода удаленного доступа SSH. В операционных системах Linux или Mac пользователям необходимо использовать открытый ключ пары ключей, созданный при запуске экземпляра, для входа в экземпляр в формате, указанном ниже

## Как применять облачную обработку данных 61

```
$:chmod 400 ssh-keypair.pem  
$:ssh -i ssh-keypair.pem username@ec2-xxx-xxx-xxx-xxx.  
compute-1.amazonaws.com
```

ssh-keypair.pem - это файл открытого ключа, а «username» - это имя учетной записи для входа в систему Linux. Имя учетной записи по умолчанию - «root» для системы Redhat Linux и «ubuntu» для системы Ubuntu. Если АМІ подготовлен Amazon, учетной записью по умолчанию обычно является «ec2-user».

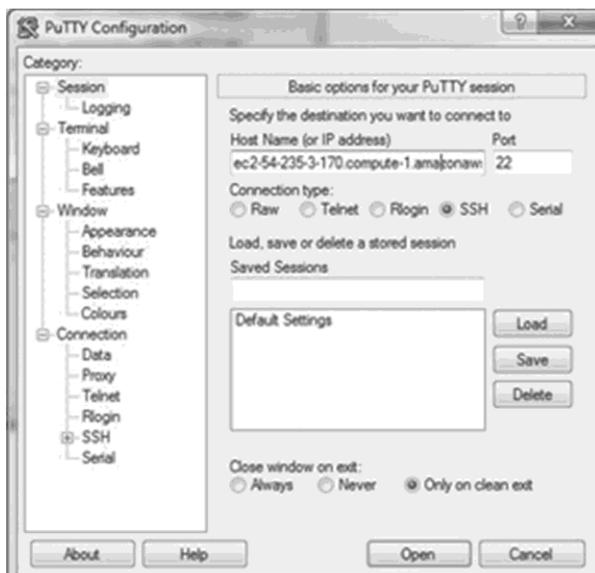
PuTTY<sup>1</sup> можно использовать для подключения экземпляра с компьютера Windows. Файлы PuTTY можно загрузить с официального веб-сайта<sup>2</sup>. В данном примере PuTTYgen необходим для преобразования пары ключей Amazon в закрытый ключ, а PuTTY необходим для подключения к работающему экземпляру. Перед входом в экземпляр пользователи облака должны загрузить пару ключей в PuTTYgen и нажать «Сохранить закрытый ключ», чтобы сохранить ключ в формате PuTTY (рисунок 4.12). После преобразования закрытого ключа пользователи могут подключаться к экземпляру с помощью PuTTY, введя IP-адрес экземпляра (рисунок 4.13a) в поле Host name и выбрав закрытый ключ в окне авторизации (щелкните Connection-SSH-Auth) (Рисунок 4.13b).



Рисунок 4.12 Преобразуйте пару ключей Amazon в закрытый ключ.

<sup>1</sup> См. PuTTY на <http://www.chiark.greenend.org.uk/~sgtatham/putty/>.

<sup>2</sup> См. PuTTY Download page на <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>.

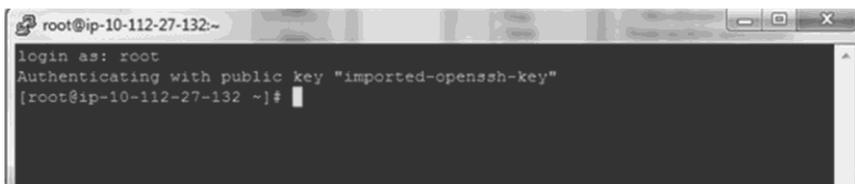


(a)



(b)

Рисунок 4.13 Войдите в экземпляр, используя PuTTY. (a) Введите имя хоста; (b) Выберите закрытый ключ; (c) Введите "root" в качестве имени пользователя.



(с)

*Рисунок 4.13* (Продолжение) Войдите в экземпляр, используя PuTTY. (с) Введите "root" в качестве имени пользователя.

После нажатия кнопки «Открыть» в PuTTY будет создано новое окно оболочки. Потребители должны ввести "root" в качестве имени пользователя для входа в экземпляр.

*Шаг 5. Установите и настройте веб-сервер* - HTTP-сервер Apache (httpd) используется для развертывания приложения Hello Cloud. Следующие команды можно использовать для установки и настройки httpd в экземпляре CentOS.

```
$: yum install httpd  
$: service httpd start  
$: chkconfig httpd on
```

Первая команда устанавливает httpd. Команда yum используется для автоматической установки программного обеспечения или библиотеки в CentOS. Вторая команда запускает httpd-сервер. Третья команда включает автоматический запуск службы httpd.

Экземпляр имеет брандмауэр по умолчанию, установленный поставщиками АМІ. Потребители должны соответствующим образом настроить брандмауэр. Следующие команды используются для настройки брандмауэра в приложении Hello Cloud.

```
$: iptables -I INPUT -p tcp --dport 80 -j ACCEPT  
$: service iptables save
```

Первая команда включает ввод веб-трафика на порт 80. Вторая команда используется для сохранения правил брандмауэра.

*Шаг 6.* Перенесите файл Hello Cloud на экземпляр - scp<sup>1</sup> используется для переноса файлов с большинства компьютеров Linux и Mac на экземпляр. Команда может безопасно копировать файлы и каталоги между удаленными хостами. Следующая команда может использоваться для передачи HTML-файла Hello Cloud.

<sup>1</sup>См. Linux на About.com на [http://linux.about.com/od/commands/l/blcmdl1\\_scp.htm](http://linux.about.com/od/commands/l/blcmdl1_scp.htm).

```
$: scp -i myfirstkey.pem index.html ubuntu@  
ec2-54-235-3-170.compute-1.amazonaws.com:/root
```

В команде `myfirstkey.pem` - закрытый ключ; индекс `html` - это HTML-файл, необходимый для передачи; `root` - это имя учетной записи в экземпляре Amazon EC2. `/root` - это имя каталога в экземпляре Amazon EC2.

Передача файлов с компьютеров Windows на экземпляр EC2 отличается. Один из самых простых способов - использовать WinSCP.<sup>1</sup> WinSCP - виджет с открытым исходным кодом, включающий SFTP, SCP, FTPS и FTP для Windows. Основная функция - передача файлов между локальным и удаленным компьютером. WinSCP.exe можно загрузить с официального веб-сайта. После установки для передачи файлов необходимы имя хоста, имя пользователя и файл закрытого ключа. Имя хоста - это IP-адрес экземпляра EC2, имя пользователя - это имя для входа в экземпляр, представленный на шаге 4, а файл закрытого ключа - это файл `ppk`, созданный PuTTYgen.

Будет установлен сеанс SFTP для подключения экземпляра и локального рабочего стола Windows при нажатии кнопки «Войти» (рисунок 4.14). Затем потребители могут передавать файлы с локального компьютера в экземпляр Amazon EC2 (рисунок 4.15).

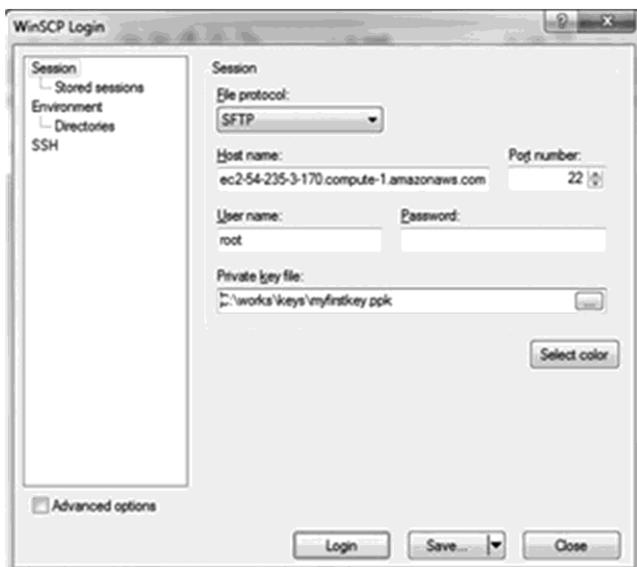


Рисунок 4.14 Вход в WinSCP.

<sup>1</sup> См. WinSCP на <http://winscp.net/eng/index.php>.

*Шаг 7.*Развертывание приложения. Потребителям необходимо развернуть приложение Hello Cloud в корневом каталоге документов httpd (/var/www). На данном шаге можно использовать следующие команды.

```
#:mv /etc/httpd/conf.d/welcome.conf /etc/httpd/conf.d/  
welcomebak.conf  
#:mv /root/index.html ./hellocloud /var/www/html  
#:service httpd restart
```

Первая строка перемещает домашнюю страницу httpd по умолчанию на другое имя. Вторая строка перемещает домашнюю страницу Hello Cloud в каталог документов. Третья строка изменяет контексты безопасности папки HTML, чтобы сделать ее доступной из браузера. Четвертая строка перезапускает HTTP-сервер. К приложению Hello Cloud после развертывания можно будет получить доступ через URL-адрес EC2 в браузере, как показано на рисунке 4.16.

*Шаг 8.* Создайте АМІ из запущенного экземпляра - наконец, новый АМІ может быть создан на основе экземпляра Hello World. В случае сбоя рабочего экземпляра систему можно очень быстро восстановить из АМІ, запустив экземпляра с помощью нового АМІ.

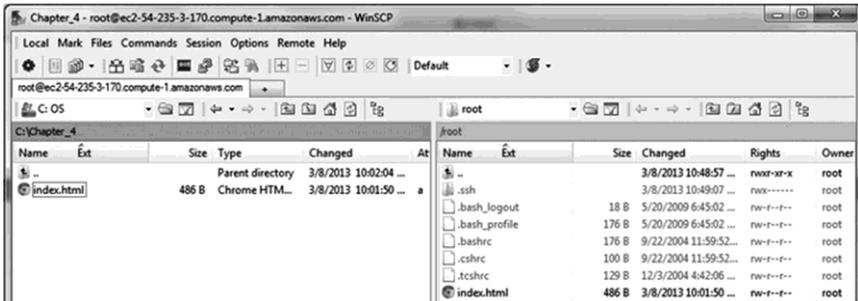


Рисунок 4.15 Передача файла с помощью WinSCP.

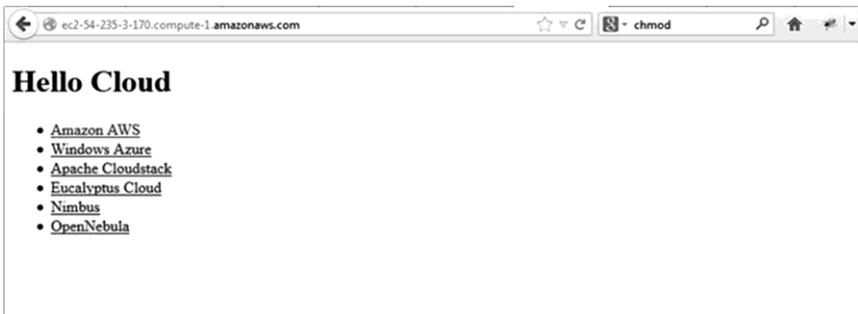


Рисунок 4.16 Hello Cloud в Amazon Cloud.

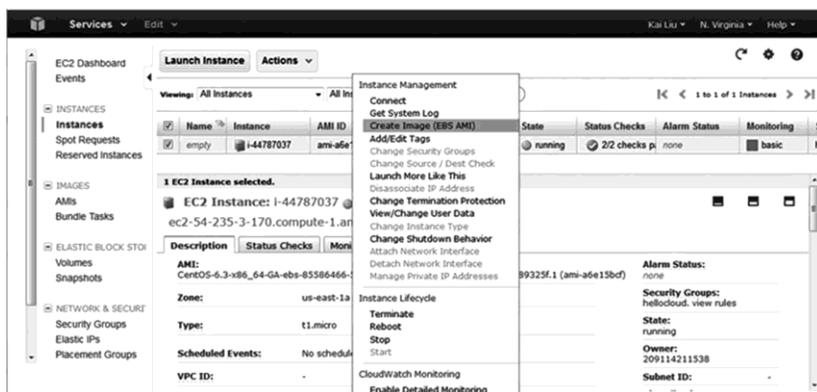


Рисунок 4.17 Создайте AMI из работающего экземпляра.

Чтобы создать AMI из запущенного экземпляра, щелкните правой кнопкой мыши запущенный экземпляр на панели инструментов EC2 и выберите Create Image (EBS AMI) (рисунок 4.17).

### 4.3.2 Windows azure

Есть два способа создать простой веб-сайт в Windows Azure: (1) с помощью виртуальной машины и (2) с помощью службы веб-сайта. На рисунке 4.18 показаны этапы развертывания Hello Cloud на виртуальной машине Windows:

*Шаг 1. Зарегистрируйтесь в Windows Azure.* Для использования Windows Azure пользователям облака необходимо зарегистрировать учетную запись на веб-сайте Windows Azure.<sup>1</sup> На рис. 4.19 показан интерфейс управления облаком со списком служб Windows Azure (например, веб-сайтов, виртуальных машин, хранилище, база данных и другие службы) после входа в систему.

*Шаг 2. Создание виртуальной машины.* В настоящем примере для развертывания простого веб-приложения используется виртуальная машина Windows и службы ИС. Щелкните вкладку «Виртуальные машины» и нажмите «Создать виртуальную машину». На рис. 4.20 показан шаг для запуска виртуальной машины Windows путем (1) выбора образа виртуальной машины (включая систему), размера виртуальной машины и местоположения виртуальной машины (выберите расположение рядом с потребителями, чтобы получить максимальную производительность сети) и (2) ввести пароль и DNS-имя. Доменное имя веб-сайта будет добавлено к cloudapp.net. Например, если имя DNS- «hellocloudtest», имя домена для веб-сайта будет http://hellocloudtest.cloudapp.net. После нажатия кнопки «Создать виртуальную машину» Windows Azure создаст новую виртуальную машину на основе входных данных (рис. 4.20).

<sup>1</sup> См. Windows Azure на <http://www.windowsazure.com>.

## Как применять облачную обработку данных 67



Рисунок 4.18 Процесс развертывания Hello Cloud в Windows Azure.

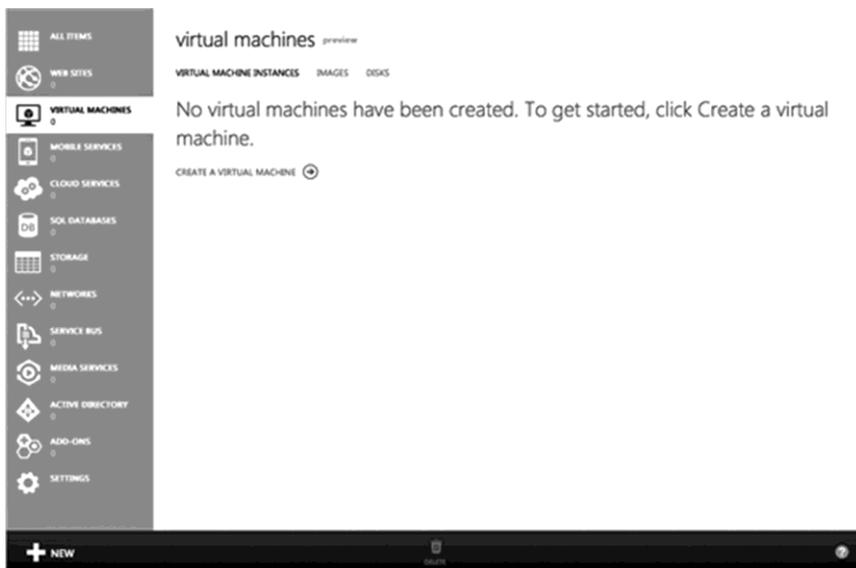


Рисунок 4.19 Интерфейс управления облаком Windows Azure.

*Шаг 3. Разрешить доступ к сети* - порт 80 требуется приложению Hello Cloud для включения веб-трафика. Пользователи облака могут авторизовать доступ к сети: (1) перейдя к вновь созданной виртуальной машине на портале Windows Azure Preview (рисунок 4.22) и щелкнув вкладку **КОНЕЧНЫЕ ТОЧКИ** (рисунок 4.22a) и (2) нажав кнопку **ДОБАВИТЬ КОНЕЧНУЮ ТОЧКУ** в нижней части окна экрана (рис. 4.22a) и открыв публичный порт 80 протокола TCP как **ЧАСТНЫЙ ПОРТ 80** (рис. 4.22b).



Рисунок 4.20 Запустите виртуальную машину Windows.

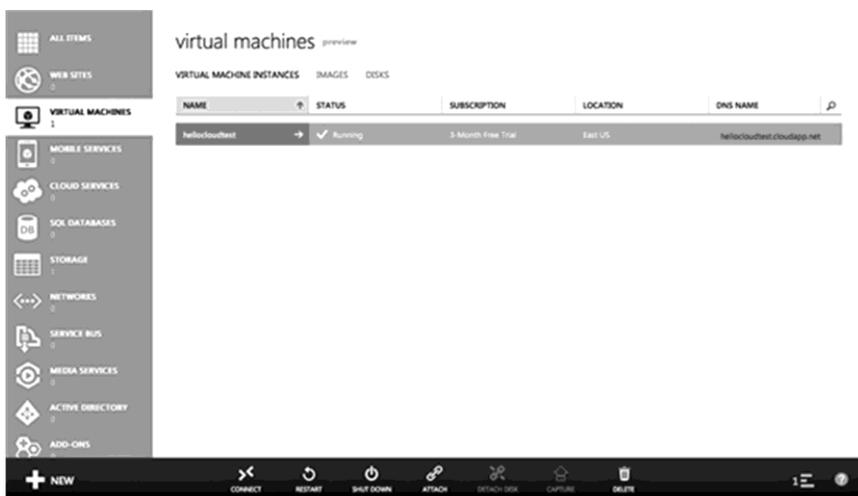


Рисунок 4.21 Интерфейс управления облаком Windows после запуска виртуальной машины.

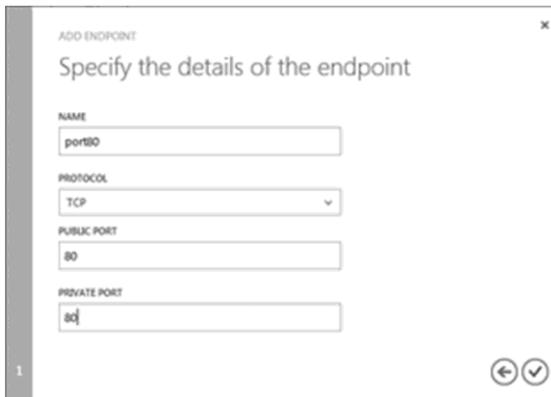
*Шаг 4.* Войдите в виртуальную машину. После запуска виртуальной машины ее предварительно можно будет просмотреть на портале Windows Azure. Пользователи облака могут нажать кнопку «Подключиться» в нижней части портала (рис. 4.21), чтобы загрузить файл RDP для подключения.

*Шаг 5.* Включите IIS. Сервер IIS используется для развертывания приложения Hello Cloud. Сервер IIS можно включить, добавив роли и функции на панели мониторинга Server Manager после входа в виртуальную машину (рис. 4.23).

## Как применять облачную обработку данных 69



(a)



(b)

Рисунок 4.22 Разрешите доступ к сети. (а) ДОБАВИТЬ КОНЕЧНУЮ ТОЧКУ; (б) Укажите детали конечной точки.

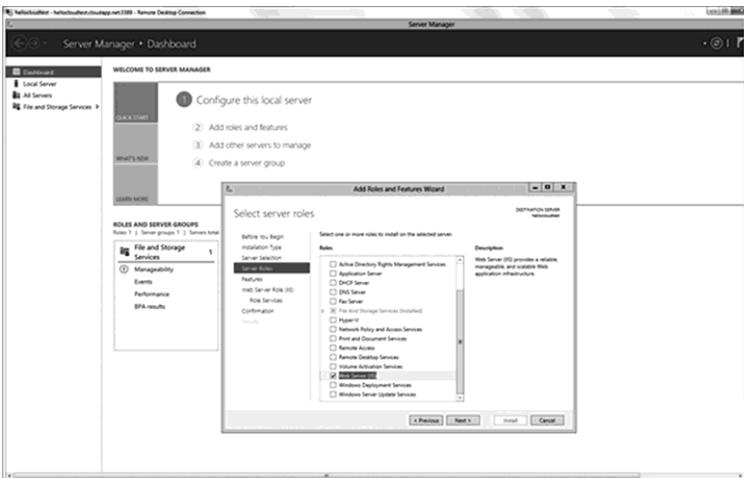


Рисунок 4.23 Включите IIS на виртуальной машине.

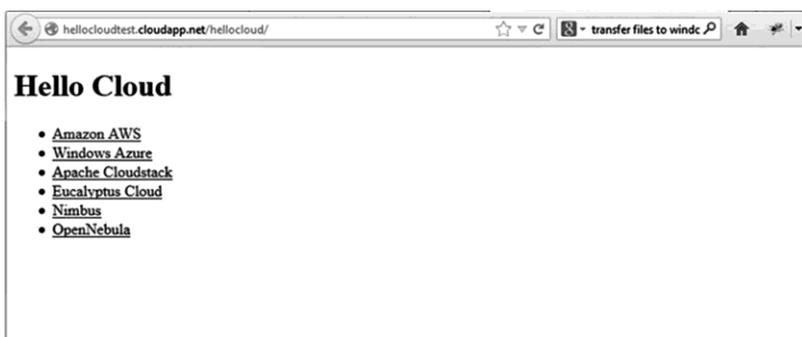


Рисунок 4.24 Приложение Hello Cloud в Windows Azure.

*Шаг 6. Перенос файлов Hello Cloud на виртуальную машину.* Пользователи облака могут просто копировать и вставлять или перетаскивать файлы между локальными компьютерами и виртуальными машинами.

*Шаг 7. Развертывание приложения* - пользователи облака могут создать папку с именем «hellocloud» в каталоге C:\inetpub\wwwroot на виртуальной машине, а затем скопировать домашнюю страницу приложения Hello Cloud в папку Hellocloud. После развертывания к приложению Hello Cloud можно будет получить доступ через URL-адрес виртуальной машины в браузере (рис. 4.24).

*Шаг 8. Захват образа работающей виртуальной машины* - потребители облака могут записать образ, выполнив следующие действия: (1) Запустите систему на виртуальной машине; (2) Выключите виртуальную машину в интерфейсе управления облаком (рисунок 4.21); и (3) Нажмите кнопку «Захват» в нижней части интерфейса управления облаком (рис. 4.21), чтобы сделать снимок. На рисунке 4.25 показано диалоговое окно после нажатия кнопки «Захват».

#### 4.4 ЗАКЛЮЧЕНИЕ И ОБСУЖДЕНИЕ

Различные облачные сервисы имеют разные интерфейсы и функции управления. Однако есть несколько общих шагов по использованию облачных сервисов. На рис. 4.26 показаны восемь общих шагов по развертыванию простого веб-приложения в облачной службе на базе Amazon AWS и Windows Azure.

*Шаг 1. Зарегистрируйтесь в облаке.* Первый шаг почти для всех публичных облаков и частных облаков.

*Шаг 2. Авторизовать доступ к сети.* Пользователи облака должны авторизовать доступ SSH или RDP к экземпляру и авторизовать определенный порт для разрешения веб-трафика.

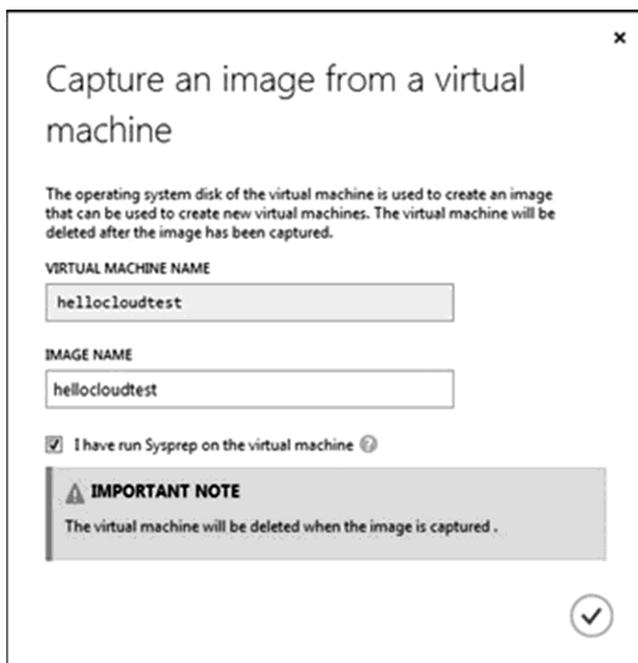


Рисунок 4.25 Сделайте снимок виртуальной машины hellocloudtest. Пользователи должны ввести имя экземпляра и установить флажок «Я запустил Sysprep на виртуальной машине».



Рисунок 4.26 Общие шаги по развертыванию простого веб-приложения в облачном сервисе. Порядок шагов в темно-сером поле может быть изменен в зависимости от различных услуг.

*Шаг 3.* Запуск экземпляра. Потребители могут запустить экземпляр, используя экземпляр виртуальной машины от облачных провайдеров или с маркета экземпляров. Различные облачные сервисы предоставляют разные образы для запуска экземпляров, например, AWS предоставляет AMI, Windows Azure предоставляет VHD, а Eucalyptus предоставляет EMI (Nurmi et al. 2009).

*Шаг 4.* Войдите в экземпляр - пользователи могут войти в запущенный экземпляр с использованием SSH или RDP соответственно.

*Шаг 5.* Установите и настройте веб-сервер - после входа

Например, пользователям необходимо настроить веб-сервер (например, httpd, Apache Tomcat или IIS) для развертывания приложения.

*Шаг 6.* Перенесите файлы приложения в экземпляр. Пользователи облака могут использовать команду scp или WinSCP, чтобы инициировать перенос файлов в экземпляр.

*Шаг 7.* Развертывание приложения - пользователи облака могут развертывать свои приложения на работающем экземпляре.

*Шаг 8.* Создайте образ работающего экземпляра. Образ можно использовать для быстрого запуска экземпляра. Пользователи облака могут создать образ на основе запущенного экземпляра (создать образ в AWS и записать образ в Windows Azure).

## 4.5 КРАТКОЕ ИЗЛОЖЕНИЕ

В настоящей главе показано, как развертывать приложения в облачных сервисах. В разделе 4.1 представлены два популярных облачных сервиса: Amazon AWS и Windows Azure. В разделе 4.2 представлено простое веб-приложение. В разделе 4.3 рассказывается, как развернуть приложение на Amazon AWS и Windows Azure. Раздел 4.4 завершает и подвергает обсуждению общие этапы рабочего процесса развертывания приложений в облачных сервисах.

## 4.6 ПРОБЛЕМЫ

1. В чем разница между завершением и остановкой экземпляра на Amazon AWS? Как очистить экземпляр на AWS, если он больше не нужен?
2. В чем разница между завершением работы и удалением виртуальной машины (ВМ) в Windows Azure?
3. Каковы общие шаги по развертыванию простого веб-приложения в Amazon EC2 и Windows Azure? Каковы отличия? Какой из сервисов кажется Вам удобнее?
4. Как Вы авторизуете доступ к сети в EC2 и Azure?
5. Какова роль пары ключей в EC2?

6. Что такое образ экземпляра в облачной обработке данных? Какая связь между экземпляром и образом?
7. Как создать образ?

## СПИСОК ЛИТЕРАТУРЫ

- Afgan, E., D. Baker, A. Nekrutenko, and J. Taylor. 2011. A reference model for deploying applications in virtualized environments. *Concurrency and Computation: Practice and Experience* 24, no. 12: 1349–1361.
- Chappell, D. 2010. Introducing the Windows Azure platform. <http://go.microsoft.com/?linkid=9682907> (accessed August 11, 2013).
- Dastjerdi, A. V., S. K. Garg, and R. Buyya. 2011. QoS-Aware deployment of network of virtual appliances across multiple clouds. In *Cloud Computing Technology and Science. (CloudCom). IEEE 3rd International Conference*, pp. 415–423.
- Emeakaroha, V.C., I. Brandic, M. Maurer, and I. Breskovic. 2011. SLA-Aware application deployment and resource allocation in clouds. In *Computer Software and Applications Conference Workshops (COMPSACW): IEEE 35th Annual*, pp. 298–303.
- Nurmi, D., R. Wolski, C. Grzegorzczak et al. 2009. The eucalyptus open-source cloud-computing system. In *Cluster Computing and the Grid. CCGRID'09. 9th IEEE/ACM International Symposium*, pp. 124–131.
- Varia, J. and S. Mathew. 2012. Overview of Amazon Web Services. [http://d36cz9buwru1tt.cloudfront.net/AWS\\_Overview.pdf](http://d36cz9buwru1tt.cloudfront.net/AWS_Overview.pdf) (accessed January 23, 2013).



## Глава 5

---

# Облачные приложения для геонаучных исследований

*Кай Лю, Цюньин Хуан и Цзичжэ Ся*

---

В 4-й главе мы рассмотрели общие шаги по установке простого веб-приложения на облачные сервисы, такие как Amazon EC2 и Windows Azure. Однако процедура развертывания геонаучных приложений на облачных сервисах гораздо сложнее. В данной главе представлены общие сложности типичных геонаучных приложений, методы устранения таких сложностей при развертывании геонаучных приложений на облачных сервисах, а также приведены два практических примера.

### 5.1 ОБЩИЕ КОМПОНЕНТЫ ДЛЯ ГЕОНАУЧНЫХ ПРИЛОЖЕНИЙ

Развертывание геонаучных приложений на облачных сервисах требует более сложных аппаратных и программных конфигураций, чем простое веб-приложение. Например, сервис геонаучных данных может потребовать от пользователей входа в систему для загрузки данных. Данный процесс авторизации обычно включает в себя серверную программу для связи с базой данных с целью проверки учетных записей пользователей. Поэтому для развертывания требуется настройка и подключение к базе данных. Другим примером является моделирование в области геонаук, которое характеризуется интенсивностью вычислений и интенсивностью данных. Высокопроизводительные вычисления (HPC) могут быть использованы для проведения такого моделирования на нескольких виртуальных машинах (VM) для сокращения времени обработки данных. Несмотря на то, что облачная обработка данных превосходит по производительности запуск большого количества VM, настройку, обмен данными и планирование таких VM следует тщательно продумать для достижения достойной и стабильной производительности. На практике серверные скрипты, базы данных и HPC являются распространенными компонентами, которые необходимо учитывать при развертывании геонаучных приложений в облачных сервисах.

#### 5.1.1 Программирование на стороне сервера

Программирование на стороне сервера - это технология, при которой запрос пользователя выполняется путем запуска программы непосредственно для генерации динамических Web-страниц.

На сервере выполняется серверная программа и полученный Web контент передается конечным пользователям (Bradley 2013). Некоторые популярные языки серверного программирования включают в себя ASP, PHP, JSP, Perl и Ruby. Получение данных из базы данных или геонаучных сервисов для создания динамического веб-контента требуется для многих геонаучных веб-приложений. Программирование на стороне сервера может помочь разработчикам легко выполнить вышеперечисленные функции с помощью следующих возможностей: (1) обеспечение защищенного доступа к геонаучным сервисам, таким как службы слой карты; (2) обеспечение связи для взаимодействия с базами данных; и (3) генерирование новых результатов с использованием логики приложений и новейших данных. Разработчики приложений могут легко обновлять веб-приложение путем изменения информации, хранящейся в базе данных, без изменения веб-страниц с помощью программирования на стороне сервера. Программирование на стороне сервера требуется в качестве компонента по умолчанию (поэтому в следующих разделах это не будет подобно описано) для большинства онлайн геонаучных приложений и веб-конsoles облачных сервисов.

### 5.1.2 База данных

Еще одним общим компонентом геонаучных приложений является база данных. Приложения в области геонаук опираются на большой объем неоднородных геонаучных данных, таких как данные наблюдения Земли (НЗ). База данных может использоваться для хранения, управления и извлечения геонаучных данных и метаданных. Многие системы управления базами данных (СУБД) имеют пространственные "плагины", позволяющие традиционной базе данных хранить геонаучные данные (например, плагин PostGIS1 из PostgreSQL2, Oracle Spatial3, пространственный движок сервера Microsoft SQL4). Такие базы данных называются пространственными. Использование пространственных баз данных в приложениях для геонаук может помочь уменьшить избыточность данных, улучшить производительность доступа к данным и повысить безопасность данных. Базы данных широко используются во многих прикладных программах в области геонаук. Например, Информационный центр ГЕОСС (глава 8) создан на базе PostgreSQL/PostGIS; Climate@Home (глава 9) разработан на базе MySQL.<sup>5</sup>

### 5.1.3 Высокопроизводительные вычисления

Обработка и анализ геонаучных данных, как правило, отнимает много времени, особенно при работе с большими объемами данных. Однако быстрое реагирование требуется для многих геонаучных приложений, таких как прогнозирование пыльных бурь и маршрутизация в режиме реального времени, которые требуют больших вычислительных ресурсов. Общей проблемой для таких приложений является то, что вычислительные требования превышают возможности традиционного отдельного процессора. HPC обеспечивает вычислительное решение для данной проблемы.

<sup>1</sup> См. PostGIS на <http://postgis.net/>.

<sup>2</sup> См. PostgreSQL на <http://www.postgresql.org>.

<sup>3</sup> См. Oracle на <http://www.oracle.com>.

<sup>4</sup> См. Microsoft SQL Сервер на <http://www.microsoft.com/en-us/sqlserver>.

<sup>5</sup> См. MySQL на <http://www.mysql.com>.

## Облачные приложения для геонаучных исследований 77

Параллельные вычисления, которые являются популярным способом достижения НРС, дают возможность обрабатывать большие объемы данных с более высоким разрешением за меньшее время (Clarke 2003). Параллельные вычисления разбивают задачу последовательных вычислений на подзадачи и распределяют подзадачи на различные процессоры. В общем, существует два метода декомпозиции задач:

- (1) декомпозиция домена; данные разбиты на разделы и распределены между различными процессорами, и каждый процессор обрабатывает подмножество данных и
- (2) функциональное разложение; каждый процессор выполняет различные части процесса одновременно. Оба метода могут использоваться для обработки геонаучных данных, однако разложение в домене чаще используется для приложений в области геонаук (Huang and Yang 2011; Xie et al. 2010).

НРС-система обычно состоит из головного узла и нескольких вычислительных узлов. Вычислительные узлы могут работать независимо и связываться через компьютерную сеть. На всех узлах устанавливается и конфигурируется промежуточное программное обеспечение для мониторинга и поддержки связи между головным узлом и вычислительными узлами. Головной узел отвечает за (1) планирование и отправку задач вычислительным узлам, (2) активацию вычислительных задач посредством настройки промежуточного программного обеспечения и (3) сбор результатов с вычислительных узлов. Для развертывания системы НРС можно использовать несколько решений с открытым исходным кодом (например, Condor, MPICH2,2 и Hadoop MapReduce3). В разделе 5.3.2 рассказывается, как развернуть такую НРС-систему на облачных сервисах для поддержки интерполяции Цифровой матрицы высот (DEM).

## 5.2 ОБЛАЧНЫЕ ПРИЛОЖЕНИЯ ДЛЯ ГЕОНАУЧНЫХ ИССЛЕДОВАНИЙ

Учитывая общие компоненты, рассмотренные выше, общий рабочий процесс развертывания геонаучных приложений на облачных сервисах показан на рисунке 5.1. Хотя данный рабочий процесс похож на развертывание простого веб-приложения (Глава 4), особое внимание уделено шагам "Настройка сред" и "Развертывание приложения".

- *Настройка сред*—ВМ по умолчанию содержит операционную систему (ОС). Некоторые программы и библиотеки, такие как HTTP-серверы, DBMS и Java Runtime Environments (JRE), необходимы многим приложениям в области геонаук. Данное программное обеспечение и библиотеки должны быть установлены и сконфигурированы должным образом. На этом этапе также необходимо настроить переменные среды. Например, JAVA\_PATH и PATH требуются для Java-приложений; JRE\_PATH требуется для сервера Tomcat.

<sup>1</sup>См. HTCondor на <http://research.cs.wisc.edu/htcondor/>.

<sup>2</sup>См. MPICH на <http://www.mpich.org/>.

<sup>3</sup>См. Hadoop MapReduce Tutorial на [http://hadoop.apache.org/docs/r1.0.4/mapred\\_tutorial.html](http://hadoop.apache.org/docs/r1.0.4/mapred_tutorial.html).



Рисунок 5.1 (См. цветную вставку) Общие шаги для облачных приложений для геонаучных исследований.

- *Развертывание приложения*—Перед развертыванием приложения виртуальную машину необходимо правильно настроить в соответствии с требованиями приложения, например, настроить базу данных с помощью скриптов базы данных и настроить службу хранения виртуальной машины, службу электронной почты и службу журналов.

### 5.3 МОДЕЛЬ ВНЕДРЕНИЯ

В настоящем разделе показано, как развернуть на Amazon EC2 два типичных приложения: веб-приложения, основанные на базе данных, и приложения для высокопроизводительных вычислений.

#### 5.3.1 Веб-приложения на основе баз данных

Большинство провайдеров облачной обработки данных поставляют базы данных в качестве услуги потребителям; например, Amazon Web Services (AWS) предоставляет службу реляционных баз данных (RDS), DynamoDB и SimpleDB; Windows Azure предоставляет SQL Azure. Пользователи облачной обработки данных также могут устанавливать предпочтительные DBMS на виртуальные машины вместо использования служб баз данных. Например, потребители могут установить MySQL на VM CentOS, аналогично установке MySQL на компьютере CentOS.

Drupal<sup>1</sup> - это фреймворк управления контентом с открытым исходным кодом, который поддерживает организацию, управление и публикацию веб-содержимого с помощью сложного программного интерфейса. Приложения Geoscience могут быть построены на основе Drupal (например, портал Climate@Home, представленный в главе 9, и платформа Geoscience).<sup>2</sup> Drupal требует наличия базы данных для хранения веб-содержимого, например, MySQL, PostgreSQL и SQL-сервера. В этом случае используется MySQL, популярная база данных с открытым исходным кодом.

<sup>1</sup>См. Drupal на <http://drupal.org/>.

<sup>2</sup>См. Geospatial Platform на <http://www.geoplatform.gov>.



Рисунок 5.2 (См. цветную вставку.) Процедура развертывания сайта Drupal на EC2 (серые поля указывают дополнительные шаги для развертывания).

Данная модель внедрения демонстрирует процедуру развертывания веб-приложения на основе Drupal (поддерживаемого базой данных MySQL) на Amazon EC2. Шаги описаны на рисунке 5.2.

*Шаг 1. Разрешить доступ к сети* - порт 22 для Secure Shell (SSH) и Порт 80 для HTTP должен быть открыт.

*Шаг 2. Запустите экземпляр* - Drupal поддерживает большинство версий Linux, но сообщество Drupal настоятельно рекомендует использовать Ubuntu.<sup>1</sup> Следовательно, для размещения приложения выбран образ Ubuntu 12.04 LTS. Образ можно запустить из мастера быстрого запуска или из AWS Marketplace консоли управления Amazon EC2, как подробно описано в главе 4, раздел 4.3.1.

*Шаг 3. Войдите в экземпляр* - пользователи Linux и Mac могут использовать команду SSH в терминале для входа в экземпляр. Пользователи Windows могут использовать PuTTY (как описано в главе 4, раздел 4.3.1).

*Шаг 4. Настройка сред.* После входа в экземпляр первым шагом является настройка сред: установка и настройка HTTP-сервера Apache и СУБД MySQL. В консоли можно использовать следующие команды.

```
$: sudo apt-get update
$: sudo apt-get install apache2
$: sudo apt-get install mysql-server mysql-client
   php5-gd
$: sudo taskset install lamp-server
```

Таким образом, первая строка обновляет систему. Вторая строка устанавливает Apache2. Третья строка устанавливает MySQL и модуль GD для php5. Для пользователя MySQL «root» при установке MySQL необходим надежный пароль.

<sup>1</sup>См. Drupal на <http://drupal.org/node/850636>.

Четвертая строка устанавливает соответствующее программное обеспечение и пакеты (например, расширение MySQL для php5 - php5-mysql, HTTP-сервер - apache2). Следующую команду можно использовать для защиты установки.

```
$: sudo mysql_secure_installation
```

Рекомендуется включить функцию перезаписи Apache2, поскольку поможет с правильным перенаправлением URL-адресов Drupal (унифицированные указатели ресурсов). Чтобы включить данную функцию, необходимо изменить конфигурации file/etc/Apache2/sites-available/default, заменив AllowOverride None на AllowOverride All. Для изменения файла рекомендуется GNU nano<sup>1</sup>. Для активации модуля перезаписи Apache необходимо выполнить следующую команду.

```
$: sudo a2enmod rewrite
```

После выполнения предыдущих шагов пользователям необходимо (1) создать учетную запись MySQL для Drupal для подключения к MySQL и (2) создать базу данных MySQL для хранения содержимого веб-приложения. Следующие команды создают учетную запись и базу данных.

```
$: mysql -u root -p
mysql> create database drupal;
mysql> grant all privileges on drupal.* to ec2drupal@
    localhost identified by 'your_password';
mysql> flush privileges;
mysql> \q
```

Первая строка выполняет вход в консоль MySQL с помощью учетной записи root. Вторая строка создает базу данных под названием Drupal. Третья строка предоставляет все привилегии для SQL операций (таких как выбор, обновление, вставка и удаление) пользователю ec2drupal. После предоставления необходимых привилегий соответствующему пользователю, вызывается команда flush, чтобы завершить настройку и применить последние настройки. Последняя строка выходит из аккаунта пользователя MySQL.

*Шаг 5.* Передача файлов на экземпляр - данный шаг передает установочный файл Drupal на запущенный экземпляр. Существует два способа передачи файла: (1) загрузить его с вебсайта Drupal<sup>2</sup> непосредственно на рабочую копию; или (2) загрузить файл на локальную машину Linux/Mac/Windows, а затем загрузить на рабочую копию. Следующая команда может быть использована для загрузки установочного файла Drupal с его вебсайта (wget<sup>1</sup> поддерживается большинством систем Linux для загрузки файлов с вебсайта).

---

<sup>1</sup> См. Nano на <https://help.ubuntu.com/community/Nano>.

<sup>2</sup> См. Drupal на <http://drupal.org/project/drupal>.

```
$: wget http://ftp.drupal.org/files/projects/drupal-7.19.zip
```

Потребители могут использовать команду `scp` или `Winscp` для загрузки файлов в экземпляр (см. Глава 4, Раздел 4.3.1). Для загрузки файла Drupal может быть использована следующая команда.

```
$: scp -i ubuntu.pem drupal-7.19.zip ubuntu@ec2-23-22-98-241.compute-1.amazonaws.com:/home/ubuntu
```

*Шаг 6.* Развертывание приложения - После передачи установочного файла Drupal на экземпляр, для развертывания приложения используются следующие команды.

```
$: sudo apt-get install unzip  
$: sudo unzip drupal-7.19.zip  
$: sudo mv drupal-7.19 /var/www/drupal  
$: sudo chown www-data:www-data /var/www/drupal -R  
$: sudo service apache2 restart
```

Первая линия устанавливает утилит распаковки. Вторая и третья строки извлекают файлы Drupal из zip-папки и копируют их в корневую папку HTTP-сервера Apache по умолчанию. В четвертой строке права собственности на файлы Drupal переходят к пользователю и группе Apache. Последняя строка перезапускает HTTP-сервер Apache.

*Шаг 7.* Настройка базы данных и администраторы веб-сайтов настраиваются через веб-интерфейс по адресу <http://INSTANCEIP/Drupal/install.php>. INSTANCEIP - это IP-адрес, выделенный AWS. Пользователи могут следовать указаниям мастера (рисунок 5.3), чтобы выполнить все шаги<sup>2</sup> для запуска веб-сайта.

В следующем примере показано, как добавить и настроить модуль "Простые карты Google" на данном веб-сайте. Модуль можно загрузить с сайта проекта Drupal.<sup>3</sup> После загрузки файлы модуля могут быть извлечены в Drupal path: `/var/www/Drupal/sites/all/modules`. Затем администратор может войти на сайт и включить модуль на странице Home-Administrator-Modules (рисунок 5.4).

---

<sup>1</sup> См. About.com-Linux на [http://linux.about.com/od/comands/lblcmd1\\_wget.htm](http://linux.about.com/od/comands/lblcmd1_wget.htm).

<sup>2</sup> См. Drupal Quick Install на <http://drupal.org/documentation/install/beginners>.

<sup>3</sup> См. Drupal Simple Google Maps на [http://drupal.org/project/simple\\_gmap](http://drupal.org/project/simple_gmap).

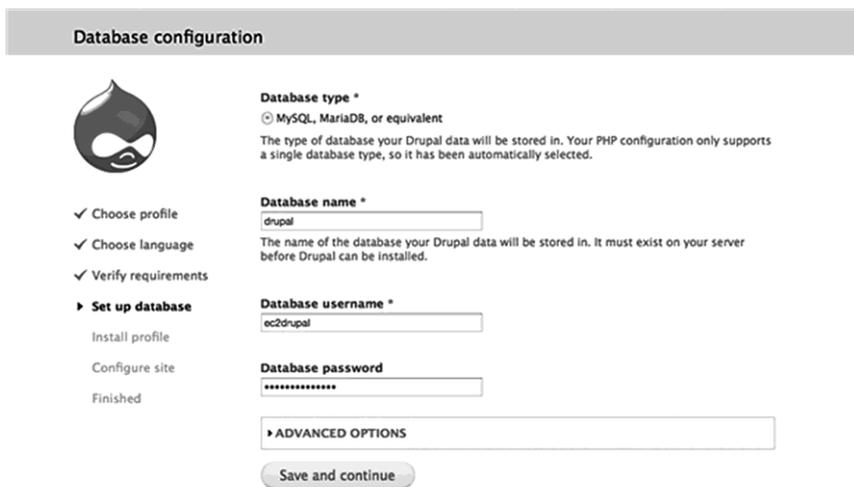


Рисунок 5.3 Настройка базы данных.

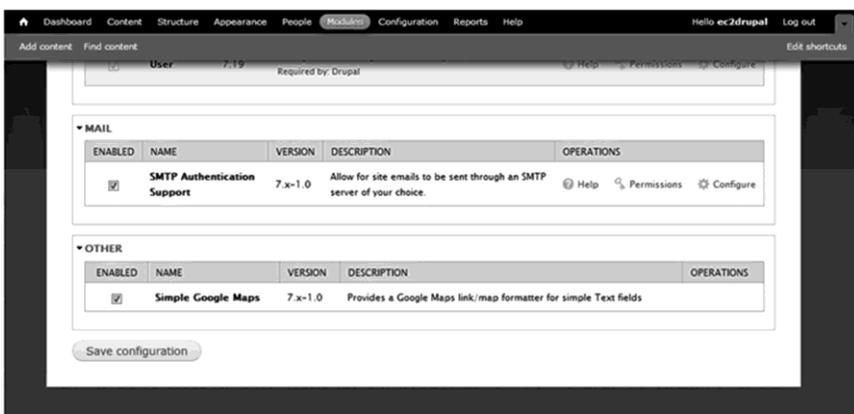


Рисунок 5.4 Добавить модуль Простые карты Google.

После включения модуля карты пользователи могут добавить поле к основному содержимому страницы на странице Home-Administration-Structure-Content types. На рисунке 5.5 показан пример того, как добавить новое поле Google Map.

Отображением нового поля можно управлять на странице Home-Administration-Structure-Content Types-Basic (рисунок 5.6).

## Облачные приложения для геонаучных исследований 83



Рисунок 5.5 Добавить новое поле в контент статьи.

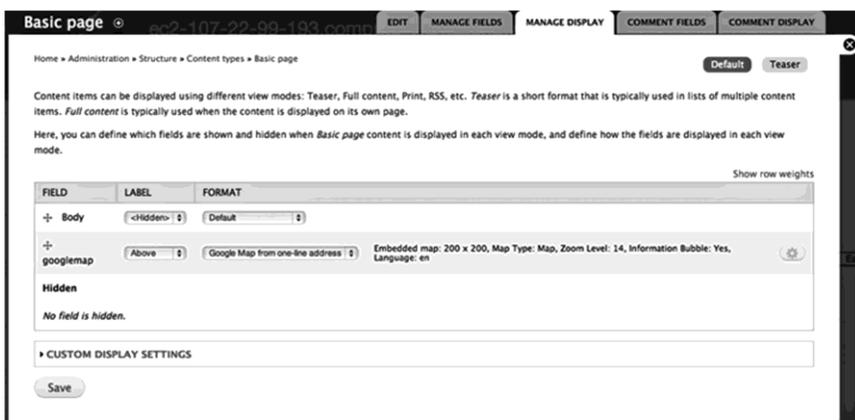


Рисунок 5.6 Управление отображением.

Новое поле Google Map можно настроить, щелкнув значок «шестеренка». Установите размер карты 800 \* 600 и уровень масштабирования 4.

При добавлении базовой страницы через страницу Home-Administration-Structure-Content Types-Basic будет добавлено поле под названием googlemap. Имя или адрес (например, США) следует ввести в поле Google Map, чтобы установить исходное местоположение на карте. Чтобы карта отображалась в отдельной вкладке, установите флажок «Предоставить ссылку на меню» и введите заголовок ссылки меню (рисунок 5.7).

После добавления поля Google Map, веб-сайт Drupal может быть запущен (Рисунок 5.8).

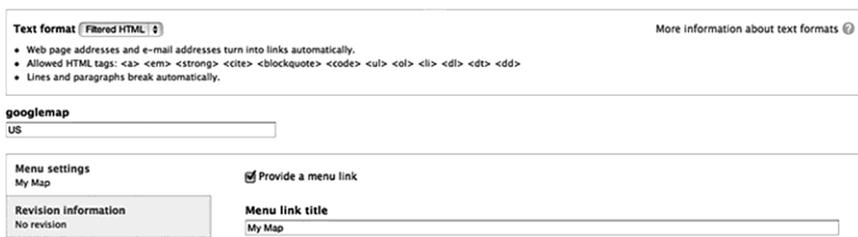


Рисунок 5.7 Добавить контент.

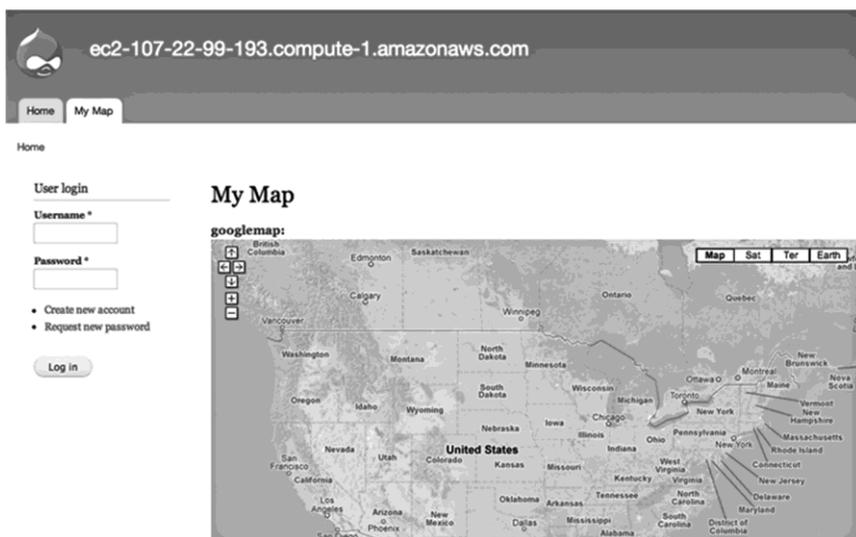


Рисунок 5.8 Интерфейс веб-сайта на Drupal.

*Шаг 8. Создайте новый АМІ из работающего экземпляра. Создайте данный АМІ, чтобы сделать полную резервную копию веб-приложения (см. Главу 4, раздел 4.3.1).*

### 5.3.2 Стандартные приложения для высокопроизводительных вычислений

В данном разделе показано, как развернуть приложение НРС на Amazon EC2 для поддержки крупномасштабной интерполяции DEM. Процедуры развертывания показаны на рисунке 5.9.

*Шаг 1. Разрешить доступ к сети (см. Главу 4, раздел 4.1) - порт 22 для SSH и порты для связи между головным узлом и вычислительными узлами (например, 9000–9999 в данном случае) должны быть открыты.*



Рисунок 5.9 (См. цветную вставку.) Процесс настройки системы НРС для запуска интерполяции DEM на EC2 (серые прямоугольники указывают дополнительные шаги для настройки виртуальной среды НРС).

*Шаг 2. Запустите экземпляр*—EC2 предоставляет экземпляры кластера для запуска приложений НРС. Экземпляры кластера<sup>1</sup> обеспечивают относительно высокие ресурсы ЦП с повышенной производительностью сети, что делает такой тип экземпляра вполне подходящим для приложений НРС. Экземпляр кластера можно запустить из специального образа машины Amazon (AMI), поддерживаемого EBS (Elastic Block Store), с использованием виртуализации аппаратной виртуальной машины (HVM) (рисунок 5.10). В дополнение к данному типу экземпляра кластера пользователи также могут выбрать экземпляры с высокой загрузкой ЦП или экземпляры кластера с высокой памятью в зависимости от того, является ли приложение геонаучных исследований интенсивным для обработки данных или вычислений (рисунок 5.11). В таком случае запускается экземпляр кластера с 8 ядрами ЦП и 23 ГБ памяти (рисунок 5.11)

После запуска экземпляра головного узла пользователи могут войти в систему через SSH. Новый экземпляр запускается с внешне доступным DNS-именем `ec2-67-202-12-83.compute-1.amazonaws.com`, который помогает получить доступ к экземпляру с помощью открытого ключа (в данном случае `EC2Key.pem`) с локального сервера. Подробные сведения о том, как войти в экземпляр Amazon EC2, см. в главе 4, раздел 4.3.1.

```
$ ssh -i EC2Key.pem  
root@ec2-67-202-12-83.compute-1.amazonaws.com
```

*Шаг 3. Установите пакеты промежуточного программного обеспечения*—В данном случае Condor является планируемым промежуточным программным обеспечением, а ниже приведены Shell-команды для установки Condor в системе CentOS 6.

<sup>1</sup>См. AWS на <http://aws.amazon.com/ec2/instance-types/>.

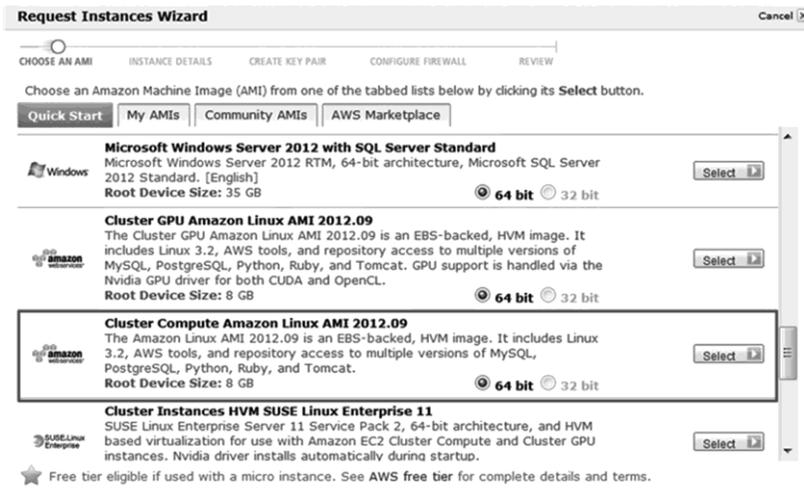


Рисунок 5.10 Запустите экземпляры кластера с Amazon Linux AMI.

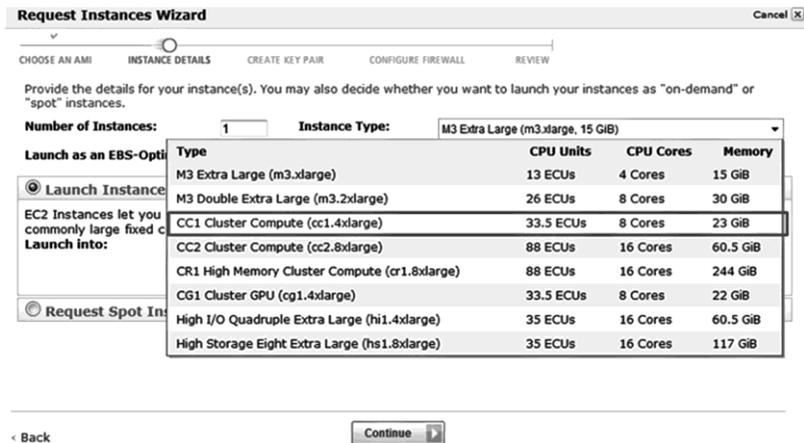


Рисунок 5.11 Выберите тип экземпляра в зависимости от требований к ЦП, памяти и сети для геонаучных приложения НРС.

```
$ rpm -Uvh http://download.fedoraproject.org/pub/epel/6/i386/epel-release-6-8.noarch.rpm ## install additional packages
$ yum install yum-plugin-priorities
$ rpm -Uvh http://repo.grid.iu.edu/osg-el6-release-latest.rpm
```

```
$ yum install condor
$ touch /etc/condor/condor_config.local ## Create
Condor configuration file
```

После успешной установки Condor необходимо создать файл конфигурации с именем local.conf в каталоге / etc / condor / config.d /, и в данный файл необходимо добавить следующую команду.

```
## OSG cluster configuration
# List of daemons on the node (Condor central manager
requires collector and negotiator,
# schedd required to submit jobs, startd to run jobs)
DAEMON_LIST = MASTER, COLLECTOR, NEGOTIATOR
```

После настройки Condor запустите и остановите службу Condor с помощью следующих команд:

```
$ service condor start ## Start the service
$ service condor stop ## Stop the service
```

Есть несколько важных команд для использования Condor (Таблица 5.1). Пользователи могут применять condor\_status, чтобы проверить, находятся ли головной узел и вычислительный узел в вычислительном пуле.

Дополнительную информацию по установке Condor можно найти на веб-сайте Condor.<sup>1</sup> Поскольку для компиляции кода интерполяции DEM используется Java Development Kit (JDK), JDK должен быть установлен на всех узлах. В экземплярах на основе AMI Amazon Linux будет предварительно установлен JDK. В противном случае двоичный пакет JDK (версия 6, обновление 14) можно загрузить с Oracle по адресу <http://java.sun.com/prod-ucts/archive/>.

Таблица 5.1 Команды Condor

Команда	Описание	Применение	Пример
condor_submit	Отправить задачу	condor_submit [submit file]	\$ condor_ submit submit.file
condor_q	Показать статус задачи	condor_q [cluster]	\$ condor_q 1170
condor_rm	Убрать задачу из очереди	condor_rm [cluster]	\$ condor_rm 1170
condor_status	Показать статус ресурсов	condor_status	\$ condor_ status-all

<sup>1</sup>См. Condor Administration Tutorial на [http://www.ccp4.ac.uk/ronan/condor\\_tutorials/scotland-admin-tutorial-2004-10-12.html](http://www.ccp4.ac.uk/ronan/condor_tutorials/scotland-admin-tutorial-2004-10-12.html).

После передачи пакета JDK в экземпляре EC2 для его установки можно использовать следующие команды.

```
[root@domU-12-31-39-13-DD-FF ~]# chmod a+x jdk-6u14-linux-x64.bin
[root@domU-12-31-39-13-DD-FF ~]# ./jdk-6u14-linux-x64.bin
```

*Шаг 5. Создайте новый AMI из работающего головного узла - такой AMI хранит зависимые объекты программного обеспечения и конфигурации для среды HPC на случай отказа головного узла. Кроме того, новые вычислительные узлы можно легко добавить в кластер, запустив вычислительные экземпляры непосредственно из данного AMI.*

*Шаг 6. Запустите другие экземпляры из нового AMI в качестве вычислительных узлов (см. Шаг 2).*

*Шаг 7. Настройте промежуточное программное обеспечение на всех узлах для обеспечения связи - после запуска вычислительного узла из нового AMI требуется дальнейшая настройка как на головном узле, так и на вычислительных узлах. На вычислительном узле конфигурации file/etc/condor/config.d/local.conf должен быть изменен на содержимое, указанное ниже*

```
## OSG cluster configuration
# List of daemons on the node (Condor central manager
# requires collector and negotiator,
# schedd required to submit jobs, startd to run jobs)
DAEMON_LIST = MASTER, SCHEDD, STARTD
```

Кроме того, на вычислительном узле должны быть локальные конфигурации file/etc/condor/condor\_config.local с указанным контентом. Конфигурации Condor по умолчанию file/etc/condor/condor\_config.local должны быть удалены во избежание путаницы.

```
UID_DOMAIN = $(FULL_HOSTNAME)
COLLECTOR_NAME = "OSG Cluster Condor at
$(UID_DOMAIN)"
FILESYSTEM_DOMAIN = $(UID_DOMAIN)
ALLOW_WRITE = *.*
CONDOR_ADMIN = root@$(FULL_HOSTNAME)
CONDOR_HOST = ip-10-112-79-17.ec2.internal ## Head
node domain name
IN_HIGHPORT = 9999
IN_LOWPORT = 9000
SEC_DAEMON_AUTHENTICATION = required
SEC_DAEMON_AUTHENTICATION_METHODS = password
SEC_CLIENT_AUTHENTICATION_METHODS = password,fs,gsi
```

## Облачные приложения для геонаучных исследований 89

---

```
SEC_PASSWORD_FILE = /var/lib/condor/condor_credential
ALLOW_DAEMON = condor_pool@*
NEGOTIATOR_INTERVAL = 20
TRUST_UID_DOMAIN = TRUE
START = TRUE
SUSPEND = FALSE
PREEMPT = FALSE
KILL = FALSE
```

*Шаг 8. Передача данных DEM и кода интерполяции.* После того как облачный кластер НРС настроен и запущен, пользователи могут передавать данные DEM и код интерполяции в головной узел (см. Раздел 5.3.1 для передачи больших объемов данных между локальными серверами и экземпляры ЕС2) и запустите тест. Чтобы отправить задачи НРС в вычислительный пул Condor, необходимо создать файл отправки, чтобы указать параметры процесса, такие как входной каталог, требования к вычислительным ресурсам, файлы программ и данных, а также номера параллельных процессов.

```
Universe = java
Executable = interpolate.class
Arguments = interpolate DEMfile.txt ## interpolate is
    the java main program, and DEMfile.txt is the input
initialdir = dir.$(Process) ## input directory
output = ../interpolate.output.$(Process) ## output
    file
error = interpolate.error.$(Process)
log = ../interpolate.log
requirements = (Memory > 1024) # Select machine with
    memory size bigger than 1024Mb
transfer_input_files = MyPoint.class, PngWriter.
    class, interpolate.class, cutfile.txt
should_transfer_files = ALWAYS
when_to_transfer_output = ON_EXIT
queue 12 ## concurrent process numbers
```

Приведенная выше команда показывает файл представления для интерполяции DEM. MyPoint.class, PngWriter.class и interpolate.class - это программы Java, а DEMfile.txt - входные данные для каждой задачи. Данный файл конфигурации имеет 12 параллельных процессов, и команду condor\_submit (Таблица 5.1), которую можно использовать для отправки задач в кластер с помощью файла отправки с именем *interpolate\_submit*.

```
[root@domU-12-31-39-13-DD-FF ~]# su condor # use the
    condor account
[root@domU-12-31-39-13-DD-FF ~]# condor_submit inter-
    polate_submit
```

После отправки задач в Condor команду `condor _q` (Таблица 5.1) можно использовать для проверки статуса всех задач. Выходные файлы `interpolate.output.X` (X обозначает номер процесса, в данном случае от 1 до 12) содержат статус каждой задачи. Раздел 16.3 в главе 16 содержит более подробную информацию об анализе результатов интерполяции DEM.

## 5.4 КРАТКОЕ ИЗЛОЖЕНИЕ

В данной главе рассказывается, как развертывать приложения для геонаучных исследований в облачных сервисах, расширяя процедуру, описанную в главе 4. В разделе 5.1 представлены общие требования к приложениям для геонаучных исследований, включая программирование на стороне сервера, базу данных и высокопроизводительные вычисления. В разделе 5.2 приведены общие шаги по развертыванию геонаучных приложений на облачных платформах. Раздел 5.3 демонстрирует подробный процесс развертывания с использованием двух практических примеров использования. Несмотря на то, что разные приложения для геонаучных исследований могут полагаться на разные технологии и, соответственно, требовать различных конфигураций оборудования и программного обеспечения, общий рабочий процесс развертывания и шаги, представленные в настоящей главе, подходят для большинства приложений.

## 5.5 ПРОБЛЕМЫ

1. Перечислите несколько других компонентов, необходимых для приложений геонаучных исследований, помимо серверных скриптов, базы данных и высокопроизводительных вычислений (HPC).
2. Каковы общие шаги по развертыванию приложения в облачных сервисах?
3. Какой сервис базы данных предоставляет Amazon AWS?
4. Какой сервис базы данных предоставляет Windows Azure?
5. Как включить другие модули, такие как электронная почта, в случае с Drupal?
6. Перечислите пять других геонаучных приложений.
7. Как настроить виртуальную среду Condor HPC в Amazon EC2?

## СПИСОК ЛИТЕРАТУРЫ

- Bradley, A. 2013. Server Side Scripting. [http://php.about.com/od/programmingglossary/g/server\\_side.htm](http://php.about.com/od/programmingglossary/g/server_side.htm) (accessed January 23, 2013).
- Clarke, K. C. 2003. Geocomputation's future at the extremes: High performance computing and nanoclients. *Parallel Computing* 29, no. 10: 1281–1295.

- Huang, Q. and C. Yang. 2011. Optimizing grid computing configuration and scheduling for geoscience analysis—An example with interpolating DEM. *Computers & Geosciences* 37, no. 2: 165–176.
- Xie, J., C. Yang, B. Zhou, and Q. Huang. 2010. High performance computing for the simulation of dust storms. *Computers, Environment and Urban Systems* 34, no. 4: 278–290.



## Глава 6

---

# Как выбрать облачные сервисы: Касательно модели стоимости облачных вычислений

*Чжипэн Гуй, Цзичжэ Ся, Наньбинь Чжоу и Цюньбин Хуан*

---

Существует множество облачных сервисов, каждый из которых имеет свои сильные и слабые стороны. Выбор подходящего сервиса облачных вычислений становится проблемой для потенциальных пользователей. В данной главе обсуждаются некоторые общие критерии выбора облачных сервисов «Инфраструктура как услуга» (IaaS) и «Платформа как услуга» (PaaS), а также представлен консультативный инструмент по облачным вычислениям для помощи в выборе облачных сервисов.

### 6.1 ВАЖНОСТЬ И ПРОБЛЕМЫ ВЫБОРА ОБЛАЧНЫХ СЕРВИСОВ

Оценка и выбор облачных сервисов - критический и сложный процесс принятия решений для пользователей облачных услуг. Например, если пользователь хочет развернуть веб-портал геонаук с большим объемом геопространственных данных и множеством функций обработки, могут возникнуть следующие вопросы:

- Какой облачный сервис выбрать для размещения приложения?
- Какая конфигурация виртуальной машины (VM) является наилучшей, не только для поддержания приложения, но и в целях рентабельной?
- Какие типы облачных хранилищ лучше всего подходят для хранения геопространственных данных?
- Где лучше всего физически разместить такие вычислительные ресурсы в соответствии с распределением потенциальных пользователей?
- Какую сумму пользователям нужно платить за свои приложения каждый день, месяц и год?

Трудности и проблемы при выборе облачных сервисов также могут возникать в связи со следующим:

- Различные функции приложения (например, объем данных, скорость передачи данных, скорость передачи данных и доступа, а также вычислительная интенсивность) могут иметь разные требования к вычислительным ресурсам (например, ЦП, память, хранилище, сеть и пропускная способность).

- С экономической точки зрения приложения совершенно разные. Например, некоторые веб-приложения разработаны для публичного доступа, а некоторые предназначены только для экспериментов или для обслуживания небольшого сообщества. Данные различия явно отражают разные бюджетные инвестиции (ограничения по оплате) и ожидания от облачных сервисов.
- С точки зрения облачных сервисов коммерческие облачные сервисы и облачные сервисы с открытым исходным кодом используют различные ИТ-технологии (например, виртуализацию и хранение) и имеют свои собственные уникальные сильные и слабые стороны в вычислительных мощностях.
- Между тем, различные модели ценообразования (например, режим по запросу / зарезервировано / торги) усложняют выбор.

Переплетение таких факторов делает выбор облачных сервисов чрезвычайно трудоемким и сложным. Это не только техническая проблема, но и проблема управления, которая включает в себя компромисс между бизнес-ожиданиями, инвестициями, предоставлением мощности и требованиями приложений. Следовательно, сделать оптимизированный выбор непросто для новичков, не имеющих предыдущего опыта или знаний в области облачных вычислений, и даже для опытных потребителей облачных вычислений.

## **6.2 ФАКТОРЫ, ВЛИЯЮЩИЕ НА ВЫБОР ОБЛАЧНЫХ СЕРВИСОВ**

Чтобы принимать мудрые решения, пользователи облака должны понимать критерии оценки облака. Были предложены различные критерии оценки (Haddad 2011; Kulkarni 2012; Repschläger et al. 2011; Rodrigues 2012). В данном разделе эти основные факторы классифицируются и представлены по трем категориям, включая предоставление емкости облачных услуг, правила ценообразования и требования к приложениям.

### **6.2.1 Предоставление и измерение емкости облачных сервисов**

Предоставление емкости является важным фактором для оценки облачных сервисов, и критерии включают в себя (CSMIC 2011; Rodrigues 2012):

- Вычислительные возможности являются важным показателем для облачных сервисов и включают в себя: (1) вычислительные возможности (например, количество и скорость ядра ЦП / ГП, объем памяти и номер виртуальной машины); (2) возможность хранения (например, объем, скорость ввода-вывода, долговечность и типы); и (3) возможности сети (пропускная способность и типы сетей).
- ИТ-безопасность и конфиденциальность (Jansen and Grance, 2011) являются критическими проблемами для пользователей облака. Федеральный закон об управлении информационной безопасностью (FISMA),

а также Федеральная программа управления рисками и авторизацией (FedRAMP) были разработаны в качестве руководства для решения проблем информационной безопасности. Таким образом, возможности защиты и гарантии безопасности и конфиденциальности можно измерить, проверив, соблюдают ли облачные провайдеры связанные с безопасностью сертификаты, такие как PCI или SAS 70, промышленные и правительственные постановления и законы, и были ли они проверены третьими лицами.

- *Надежность и лояльность* (Badger et al. 2012). Надежность описывает уверенность в том, что пользователю могут быть предоставлены доступные облачные сервисы. Важно знать, какие существуют соглашения об уровне обслуживания (SLA) и как соблюдаются данные обязательные соглашения. Кроме того, надежность описывает особенности инфраструктуры провайдера, что может свидетельствовать о высокой надежности. К ним относятся услуги по аварийному восстановлению и резервные сайты.
- *Степень настройки и гибкость / масштабируемость* (CSMIC 2011). Функции самообслуживания по требованию и возможности быстрой эластичности зависят от гибкости и возможностей настройки. Степень настройки относится к способности настраивать конфигурацию облачных сервисов в соответствии с требованиями потребителей облачных вычислений. Можно измерить по доступному количеству типов конфигурации (например, типов виртуальных машин). Некоторые провайдеры даже предлагают полностью настраиваемые виртуальные машины. Например, потребители могут указать ЦП, память, локальный диск и другие конфигурации оборудования виртуальной машины в разумных пределах. Помимо конфигурации оборудования, конфигурация программного обеспечения, такая как поддерживаемая операционная система (ОС) (например, типы и версии), также являются важными измерениями. Поскольку требования к вычислительным ресурсам для геопространственного приложения изменчивы, критически важна возможность динамического изменения предоставления вычислительных мощностей (т.е. масштабируемости). Обычно рассматриваются два типа масштабируемости: вертикальное масштабирование и горизонтальное масштабирование. Под вертикальным масштабированием понимается возможность обновления отдельных экземпляров виртуальных машин путем добавления дополнительной памяти, дополнительных процессоров или места для хранения. Горизонтальное масштабирование означает быстрое выделение новых экземпляров виртуальных машин для выполнения интенсивных задач и одновременных запросов.
- *Управляемость, удобство использования и обслуживание клиентов* (CSMIC 2011). Функции управления для координации и мониторинга облачных сервисов (например, индивидуализация и интерактивность веб-интерфейса) являются важными функциями. Хороший облачный сервис должен быть удобным для пользователя, то есть с ним легко ориентироваться, удобно использовать и сопровождать комплексными функциями мониторинга (например, производительности, использования и стоимости). Обслуживание клиентов должно включать в себя техническую помощь по телефону, электронной почте, онлайн-чату и другими способами (база знаний и форумы пользователей). Хорошее обслуживание клиентов может помочь потребителям узнать, как использовать облачные сервисы и управлять ими.

Бесплатные пробные версии - еще одно важное средство измерения, которое может помочь потребителям попробовать и протестировать услуги, прежде чем принять решение о покупке.

- *Геолокация облачных инфраструктур* не только влияет на доступность и производительность обработки данных, но также отражает некоторые ограничения политики. И потребители, и пользователи облака могут ожидать, что ресурсы облака будут ближе к их местоположению для повышения производительности и контроля. Кроме того, поставщику данных, возможно, придется соблюдать международные, федеральные или государственные нормативные акты, запрещающие хранение данных за пределами определенных физических границ (Badger et al. 2012).

## 6.2.2 Правила ценообразования облачной платформы

Чтобы выбрать рентабельный облачный сервис, который соответствует потребностям пользователя облачной обработки данных, необходимо учитывать модель стоимости:

- *Стоимость использования виртуальных машин.* Стоимость определяется количеством виртуальных машин, типами виртуальных машин, типами оплаты, интенсивностью использования и сроком аренды. Поставщики облачных услуг обычно предлагают несколько предопределенных типов виртуальных машин или полностью настраиваемые виртуальные машины, которые позволяют потребителям настраивать их в соответствии с требованиями приложений. Такие типы виртуальных машин различаются как аппаратными, так и программными конфигурациями, поэтому они предлагают разные вычислительные мощности и создают разные затраты на использование.

Существует несколько широко используемых типов оплаты: (1) Так называемый режим оплаты по мере использования позволяет потребителям оплачивать вычислительные мощности по запросу, а не по долгосрочным обязательствам.

- (2) Зарезервированный режим (с предоплатой Windows Azure и подпиской через CloudSigma) дает возможность сделать небольшой разовый платеж для резервирования облачных ресурсов на фиксированные периоды времени и, в свою очередь, получить значительную скидку по сравнению с режимом почасовой оплаты. (3) Закупка единиц - это режим выставления счетов на основе единиц (например, FlexiScale<sup>1</sup> и OpSource<sup>2</sup>). В данном режиме пользователи облака покупают кредитные единицы и используют единицы на потребляемых облачных ресурсах (например, VM, хранилище, сеть и образы программного обеспечения). Например, FlexiScale взимает 16 единиц в час за виртуальную машину с 4 ядрами ЦП и объемом ОЗУ 6ГБ. И (4) режим назначения ставок позволяет потребителям делать ставки за неиспользованные ресурсы (например, спотовые экземпляры Amazon EC2).

Интенсивность использования виртуальной машины и сети также является фактором затрат. Например, Amazon классифицирует уровни интенсивности Light, Medium и Heavy для зарезервированных типов экземпляров. За разную интенсивность взимается разная цена за единицу.

<sup>1</sup> См. FlexiScale на <http://www.flexiscale.com/>.

<sup>2</sup> См. OpSource на <http://www.opsources.net/>.

- *Стоимость передачи данных* - цена передачи данных основана на данных, передаваемых внутри и снаружи, геолокации (регион/зона разделения), а также объеме данных. Плата за передачу данных через Интернет взимается при передаче данных внутри/вне облачного сервиса или между регионами. Плата за передачу данных по регионам взимается, если данные передаются не через границы зон, а внутри одного и того же провайдера облачных услуг. Например, за передачу данных между экземплярами AWS, расположенными в разных зонах доступности в пределах одного и того же региона, взимается плата за передачу региональных данных. За передачу данных между экземплярами AWS, расположенными в разных регионах, взимается плата в виде платы за передачу данных через Интернет по обе стороны от передачи.
- *Стоимость хранения данных/базы* - Сборы за хранение данных определяются размером данных, периодом хранения и типами хранения. Поставщики могут предоставлять несколько типов хранилищ данных в соответствии с различными требованиями и целями применения. Например, Amazon Elastic BlockStore (EBS) разработан специально для экземпляров Amazon EC2 в одной и той же зоне доступности. Служба Amazon Simple Storage Service (S3) предназначена для хранения и извлечения любых объемов данных в любое время и из любого места в Интернете. S3 предлагает Стандартное хранение, Сокращенное резервное хранение (RRS) и Хранилище Glacier. Стандартная система хранения данных обеспечивает высоконадежную инфраструктуру хранения, предназначенную для критически важных и первичных данных (долговечность 99,9999999% и доступность 99,99%); система RRS обеспечивает более низкую надежность (99,99%) по сравнению со стандартной системой хранения данных и предназначена для хранения воспроизводимых данных (например, эскизов, перекодированных носителей или других обработанных данных). Хранилище Glacier предназначено для редко используемых данных (например, для цифровых мультимедийных архивов и долгосрочных резервных копий баз данных).
- *Другие расходы* - сборы за IP-адреса, взимаемые за дополнительные IP-адреса и доменные имена, связанные с экземплярами VM. Дополнительные сетевые сборы включают в себя расходы на виртуальные сети, сети доставки контента (CDN) и другие. Плата за связь относится к дополнительным платежам за доступ к облачным инфраструктурам (например, Amazon взимает плату за запросы, отправленные в S3). В некоторых "облачных" сервисах также взимается плата за поддержку.

### 6.2.3 Характеристики и требования к применению

Функции и требования приложений доминируют при выборе облачных услуг и иногда влияют на предоставляемые возможности облачных сервисов. Это должно быть тщательно продумано при выборе облачного сервиса. Многочисленные типы облачных хранилищ Amazon (описанные в разделе 6.2.2) являются хорошими примерами того, как цели использования (хранение архивов данных или необработанных/результатных данных приложения) и требования к емкости (по безопасности, надежности, скорости ввода/вывода и частоте доступа) влияют на предоставление провайдером емкости.

Другим примером является функция интенсивности применения. Приложения, интенсивно работающие с данными, требуют большого пула хранения для управления и обработки больших данных. Объем данных, производительность ввода-вывода, индексирование данных, резервное копирование данных, безопасность и надежность являются критически важными возможностями. Приложения, интенсивно использующие вычисления, требуют больших вычислительных мощностей (например, высокой скорости ЦП, большого количества процессорных ядер и большого объема памяти) для выполнения крупномасштабных вычислений. Приложение может использовать вычислительные мощности нескольких машин для повышения производительности; поэтому сетевые конфигурации и оптимизация связи являются критически важными. Тип экземпляра Amazon EC2 для высокопроизводительных вычислений предлагает возможность более эффективной поддержки данной категории приложений. Одновременные интенсивные приложения обычно включают в себя интенсивные одновременные запросы/ответы от пользователей или существующих веб-служб. Большая пропускная способность критически важна для обеспечения достойной производительности связи через Интернет. Балансировка нагрузки является еще одной возможностью для управления параллельной интенсивности, которая использует несколько, а иногда и распределенных экземпляров приложений в совместном порядке, чтобы сбалансировать запросы конечного пользователя и обеспечивает единую точку доступа к сети Web. Приложения, интенсивно использующие средства связи (или так называемые приложения без границ), в значительной степени зависят от услуг третьих сторон, которые находятся за пределами организационных и географических "границ" приложений. Точно так же, как и одновременное использование интенсивных приложений, для быстрой связи между сетями требуется большая пропускная способность, а для надежной работы требуется балансировка нагрузки.

### **6.3 ВЫБОР ОБЛАЧНЫХ СЕРВИСОВ С ИСПОЛЬЗОВАНИЕМ В КАЧЕСТВЕ ПРИМЕРА КОНСУЛЬТАЦИОННОГО ИНСТРУМЕНТА ПО ВНЕДРЕНИЮ ОБЛАЧНЫХ ТЕХНОЛОГИЙ, РАЗРАБОТАННОГО ПАРТНЕРАМИ ПО ИНФОРМАЦИОННОЙ ДЕЯТЕЛЬНОСТИ В ОБЛАСТИ ГЕОНАУК (ESIP)**

Крайне желательно спроектировать и разработать веб-консультативный инструмент, который поможет потребителям из области геонаук сравнить и выбрать наиболее подходящий облачный сервис. Такой инструмент должен объединять знания об облачных сервисах и быть способен рекомендовать облачные сервисы для достижения как экономической эффективности, так и высокой производительности. Идеальный консультативный веб-инструмент должен обеспечивать следующие возможности: (1) Помогать потребителям облачной обработки данных легко выбирать лучшие решения на основе их требований к приложениям; (2) Помогать новичкам в области облачной обработки данных понять основные понятия услуг облачной обработки данных, технологий, поставщиков и потенциальных приложений; и (3) Автоматически собирать и управлять информацией о ценах и конфигурациях нескольких поставщиков облачной обработки данных. Для проверки возможности использования консультационных инструментов для внедрения облачной обработки данных мы разработали для Федерации партнеров по информационной деятельности в области геонаук (ESIP) консультационный инструмент<sup>1</sup> для внедрения облачной обработки данных в Интернете.

<sup>1</sup> См. Cloud Computing Advisory Adoption Tool на <http://swp.gmu.edu:8080/ESIPCostModelProject>.

### 6.3.1 Архитектура консультационного инструмента

Архитектура консультационного инструмента (рис. 6.1) состоит из четырех основных компонентов, включая создателя решения, оценщика решения, облачную информационную базу данных и Web-графический интерфейс.

- Создатель решения генерирует возможные решения и возвращает рекомендуемые решения.
- Специалист по оценке решений рассчитывает потенциальную стоимость аренды и оценивает пригодность создаваемых решений.
- В облачных информационных базах данных хранится собранная информация об облачных сервисах, включая правила ценообразования, схему конфигурации и объявления о возможностях.
- Web GUI управляет взаимодействием с пользователем, представлением решения и визуализацией. Потребители облачных сервисов определяют требования и ограничения для своих приложений с помощью мастера описания требований к приложениям. Для отображения и сравнения облачных сервисов используются таблицы и диаграммы решений.

### 6.3.2 Общая последовательность выполняемых действий для выбора облачного сервиса

Для потребителя облачных услуг общая процедура выбора облачного сервиса включает в себя следующие четыре этапа: определение типа приложения, определение требований и функций приложения, поиск подходящих облачных сервисов и сравнение услуг. В следующих параграфах в качестве примера используется консультационный инструмент ESIP для подробного описания последовательности выполняемых действий при выборе облачного сервиса.

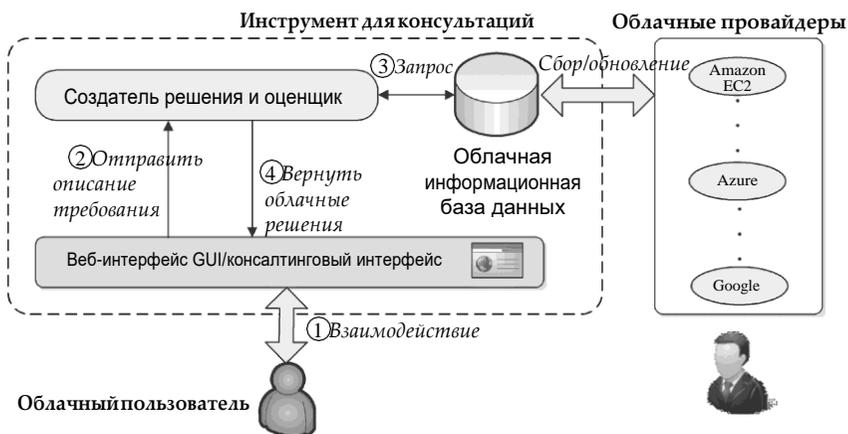


Рисунок 6.1 Архитектура и взаимодействие рабочего процесса консультативного инструмента.

Первым важным шагом при выборе облачного сервиса является понимание типа приложения, которое будет развернуто в облаке. Обычно тип приложения определяет вычислительные возможности для аппаратной и программной конфигурации облачных сервисов. К таким распространенным типам геопространственных приложений относятся веб-приложения, научные вычисления, приложения для обработки данных и другие. В консультационном инструменте три типа приложений предопределены в качестве шаблонов для потребителей (рис. 6.2): (1) Приложение - хранилище данных предназначено для хранения данных в облачном хранилище, где приложение не будет развернуто. (2) Простое веб-приложение подходит для веб-приложений малого/среднего масштаба, таких как геопространственные веб-порталы и службы (например, картографические службы). (3) Простая компьютерная прикладная программа предназначена для вычислительной техники, требующей повышенной оперативности (например, для прогнозирования пыльных бурь). Для экспертов, обладающих достаточными знаниями в области облачной обработки данных и параметров приложений, в качестве шаблона консультативной программы предлагается также Индивидуальное приложение.

Следующие три части должны быть тщательно продуманы с учетом требований и особенностей применения, включая требования к аппаратным средствам, особенности применения и предпочтения при покупке (Рисунок 6.3). Аппаратные требования могут быть либо базовыми, либо лучшими аппаратными конфигурациями для запуска приложения (например, номер ядра ЦПУ, скорость ЦПУ, объем памяти и размер локального диска). Функции приложения описывают вычислительные возможности и требования к программному обеспечению приложений (например, типы и версии ОС, интенсивность взаимодействия с пользователем, размер приложения и данных, сетевой трафик и скорость, требования к долговечности данных).

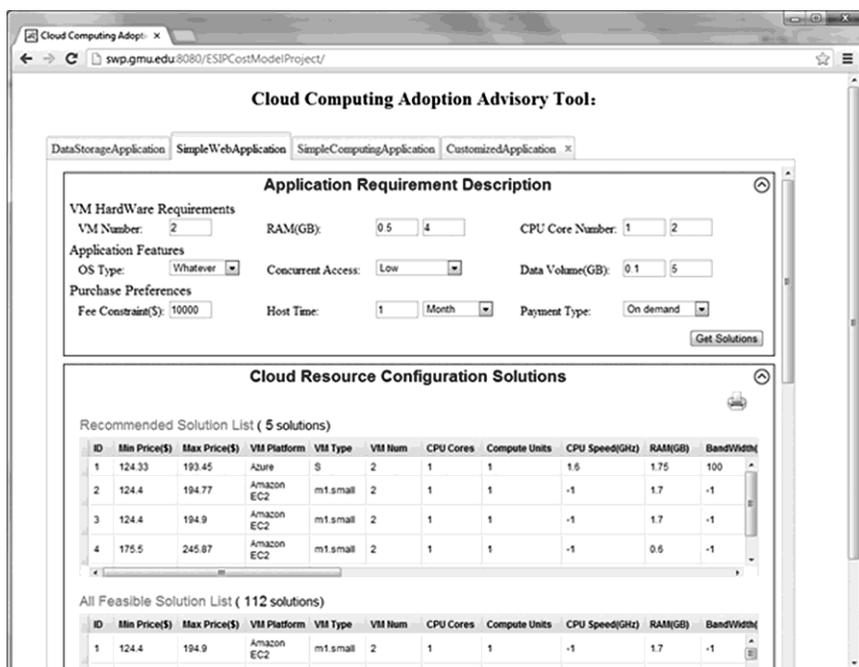


Рисунок 6.2 Основной графический интерфейс консультативного инструмента по выбору облачной обработки данных.

## Как выбрать облачные сервисы: Касательно модели стоимости облачных вычислений 101

Application Requirement Description		
<b>VM HardWare Requirements</b>		
VM Number:	<input type="text" value="1"/> <input type="text" value="4"/>	RAM(GB): <input type="text" value="2"/> <input type="text" value="8"/>
CPU Core Number:	<input type="text" value="2"/> <input type="text" value="4"/>	Local Disk(GB): <input type="text" value="10.0"/> <input type="text" value="30.0"/>
	CPU Speed(GHz): <input type="text" value="1.6"/> <input type="text" value="2.8"/>	
<b>Application Features</b>		
OS Type:	<input type="text" value="Whatever"/>	Local Data(GB): <input type="text" value="5.0"/> <input type="text" value="10.0"/>
Data Volume(GB):	<input type="text" value="60.0"/> <input type="text" value="100"/>	Application Size(GB): <input type="text" value="0.5"/>
	Task WorkLoad: <input type="text" value="Medium"/>	
<b>Purchase Preferences</b>		
Host Time:	<input type="text" value="1"/> <input type="text" value="Day"/>	Fee Constraint(\$): <input type="text" value="20000"/>
		Payment Type: <input type="text" value="On demand"/>
<input type="button" value="Get Solutions"/>		

Рисунок 6.3 Описание требований для простых вычислительных приложений.

Предпочтения при покупке относятся к тому, как потребители предпочитают платить (например, тип оплаты, срок аренды и ограничения по стоимости). В консультативном инструменте ESIP, посредством predefined шаблонов типа приложения, потребители могут напрямую указывать требования и функции приложения в веб-интерфейсе консультационного инструмента. С помощью шаблона Индивидуального приложения потребитель может указать наиболее полные требования и описания функций, а консультационный инструмент также обеспечит точные данные и оценку стоимости.

После определения требований и функций приложения потребитель может использовать их в качестве ограничений для поиска и получения осуществимых решений для облачных сервисов. Данный процесс занимает много времени. Потребитель должен собрать все конфигурации, цены и техническую информацию для всех облачных сервисов, а затем отфильтровать их, используя определенные требования и функции приложения, и создать все возможные решения для облачных сервисов.

Чтобы сделать мудрый выбор из всех возможных решений, потребитель должен провести дополнительный анализ стоимости, конфигурации, вычислительных мощностей, репутации поставщика и др. Поэтому важно выработать понимание того, как их сравнивать. Чтобы облегчить выбор облачного решения, консультационное средство динамически генерирует облачные решения в соответствии с входными данными пользователя и предоставляет интерфейс для визуального сравнения решений по стоимости, конфигурации и вычислительной мощности VM. Когда будут найдены облачные решения, в Web GUI будут представлены две таблицы и три графика. (1) В таблице "Список рекомендуемых решений" перечислены рекомендуемые решения, а в таблице "Список всех доступных решений" - отфильтрованные возможные решения (Рисунок 6.4). Конфигурация и стоимость облачных решений представлены в виде столбцов таблицы. Пользователи могут сортировать список решений по любому из столбцов таблицы. Это поможет пользователям облака ранжировать решения по различным критериям. (2) Диаграмма минимальной платы (рисунок 6.5 сверху) и диаграмма максимальной платы (рисунок 6.5 внизу) дают потребителям четкую информацию для сравнения потенциальных диапазонов платы за рекомендуемые решения.

Recommended Solution List ( 5 solutions)

ID	Min Price(\$)	Max Price(\$)	VM Platform	VM Type	VM Num	CPU Cores	Compute Units	CPU Speed(GHz)	RAM(GB)	BandWidth
1	29.04	50.5	Azure	M	4	2	2	1.6	3.5	200
2	28.63	74.08	Azure	M	4	2	2	1.6	3.5	200
3	30.13	76.56	Azure	M	4	2	2	1.6	3.5	200
4	43.8	65.26	Amazon EC2	m1.large	4	2	2	-1	7.5	-1
5	44.4	65.86	Azure	L	4	4	4	1.6	7	400

All Feasible Solution List ( 51 solutions)

ID	Min Price(\$)	Max Price(\$)	VM Platform	VM Type	VM Num	CPU Cores	Compute Units	CPU Speed(GHz)	RAM(GB)	BandWidth
1	45.3	67.76	Amazon EC2	m1.large	4	2	2	-1	7.5	-1
2	43.8	65.26	Amazon EC2	m1.large	4	2	2	-1	7.5	-1
3	44.71	91.67	Amazon EC2	m1.large	4	2	2	-1	7.5	-1
4	76.02	98.48	Amazon EC2	m1.xlarge	4	4	8	-1	15	-1
5	74.52	95.98	Amazon EC2	m1.xlarge	4	4	8	-1	15	-1

Рисунок 6.4 Список рекомендуемых решений и список всех доступных решений.

Разделив стоимость (показана в столбцах) определенного решения на несколько частей (плата за VM, плата за хранение и плата за передачу данных), пользователи могут интуитивно понять, какие проценты они платят за каждую часть сбора. (3) Диаграмма сравнения конфигурации виртуальной машины (рисунок 6.6) показывает конфигурацию виртуальной машины (номер ядра процессора, номер виртуального вычислительного устройства, скорость процессора, размер оперативной памяти, пропускную способность, размер локального диска) выбранных решений в виде линейной последовательной диаграммы. Это помогает потребителям сравнивать параметры емкости VM.

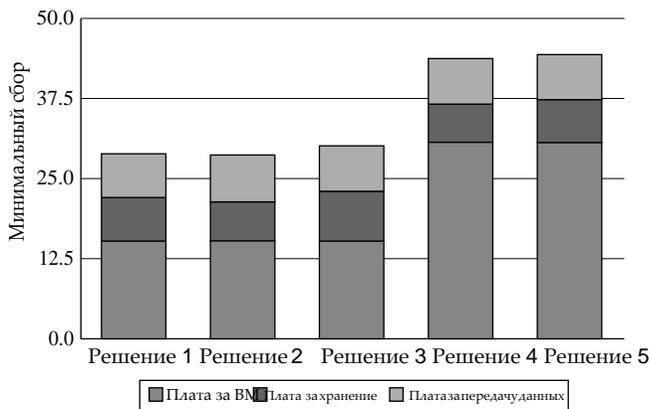
### 6.3.3 Метод внедрения

Мы используем пример вычислительного приложения, чтобы показать, как использовать консультационный инструмент для поиска облачных решений.

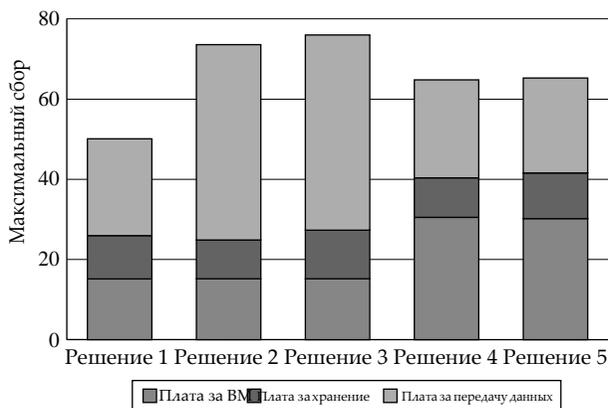
Цепочка последовательности взаимодействия включает в себя четыре этапа:

*Шаг 1.* Описание требований к приложениям - потенциальный пользователь указывает описание требований к приложениям для потенциального вычислительного приложения на странице Simple Computing Application. Поскольку вычислительное приложение требует более высоких вычислительных мощностей, чем простое веб-приложение, параметры требования приложения указываются (рисунок 6.3). Например, счетчик ядра процессора устанавливается равным как минимум 2, а желательно 4; интенсивность задачи вычисления - средняя; срок аренды - один день; тип оплаты - по требованию

*Шаг 2.* Get Solutions - Потенциальный пользователь нажимает кнопку Get Solutions для генерации решений.



(a)



(b)

Рисунок 6.5 (См. цветную вставку.) (a) Минимальный сбор и (b) Максимальный сбор.

*Шаг 3.* Таблицы и диаграммы - потенциальный пользователь взаимодействует с решениями, представленными в таблицах и диаграммах. Пять рекомендуемых решений и более осуществимые решения перечислены в двух компонентах таблицы (рисунок 6.4). Графики минимального и максимального сбора (Рисунок 6.5) показывают, что рекомендованные решения №4 и №5 имеют большую долю затрат по аренде VM. Диаграмма сравнения конфигурации VM (Рисунок 6.6) показывает, что решение №5 (полилиния с розовым цветом [см. цветовую вставку] в линейной последовательной диаграмме) предлагает самые высокие вычислительные мощности

*Шаг 4.* Печать - Пользователь выводит результаты в PDF-файл, нажимая кнопку "Печать".

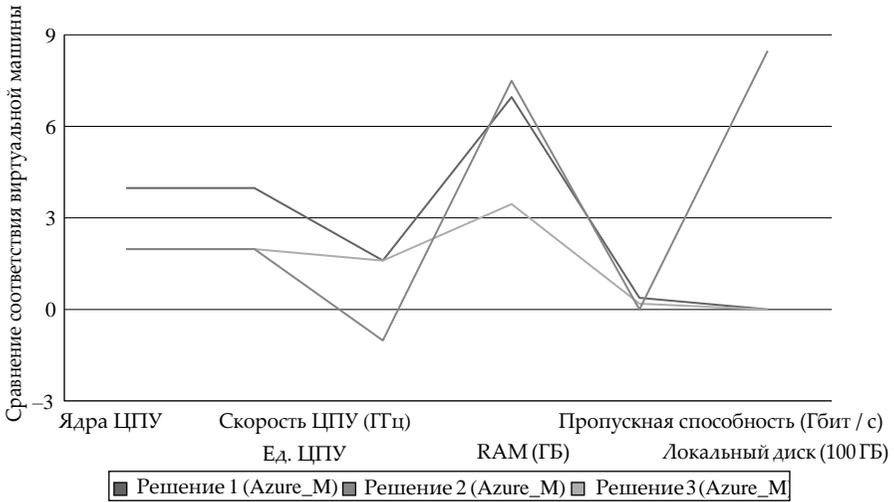


Рисунок 6.6 (См. цветную вставку.) Сравнение конфигурации виртуальных машин.

Таким образом, выбор облачного сервиса - это знания, опыт и трудоемкость. В разработанном консультативном инструменте была подтверждена разработка экспертной системы внедрения облачной обработки данных, включая расчет стоимости и подходящие модели оценки. Это может (1) создать осуществимые облачные решения в соответствии с данными пользователя облачной обработки данных; (2) рассчитать и визуально сравнить стоимость и возможности решений, и (3) рекомендовать решения на основе встроенных механизмов оценки.

### 6.4 УГЛУБЛЕННОЕ РАССМОТРЕНИЕ ВОПРОСОВ ВЫБОРА ОБЛАЧНЫХ СЕРВИСОВ И РАЗРАБОТКА КОНСУЛЬТАЦИОННЫХ ИНСТРУМЕНТОВ

Большое количество исследований было проведено по выбору облачных сервисов (Badger и др., 2012). В частности, метрики облачных сервисов<sup>1</sup> (CSMIC 2011; NIST 2012) начинаются как инициатива по созданию последовательной и работоспособной системы измерений, позволяющей заинтересованным сторонам облачных сервисов более эффективно взаимодействовать друг с другом. Модели оценки и отбора облачных сервисов (Andrzejak, Kondo, Yi 2010; Calheiros и др. 2011; Repschläger и др. 2011; Stantchev 2009) предоставляют теоретические методы, помогающие принимать решения. Разработаны системы мониторинга<sup>2</sup> и вспомогательные системы отбора/веб-сайты<sup>3</sup> (Goscinski and Brock 2010; Martens, Teuteberg, и Gräuler 2011) для поддержки облачных сервисов в режиме реального времени.

<sup>1</sup> См. NIST RATAx Cloud Metrics Sub Group на [http://collaborate.nist.gov/twiki-cloud-computing/bin/view/CloudComputing/RATax\\_CloudMetrics](http://collaborate.nist.gov/twiki-cloud-computing/bin/view/CloudComputing/RATax_CloudMetrics)

<sup>2</sup> См. Global Provider View, CloudSleuth на <https://cloudsleuth.net/global-provider-view>.

<sup>3</sup> См. FindTheBest, Cloud Computing Providers на <http://cloud-computing.findthebest.com>.

Хотя в отношении ранее обсуждавшихся аспектов был достигнут большой прогресс, некоторые проблемы все еще остаются нерешенными.

### 6.4.1 Правильность и точность моделей оценки

Без высококачественных моделей оценки облачные решения не очень точны. При этом создание надежных моделей и их проверка - большая проблема.

- *Модель расчета затрат* - несмотря на наличие вариаций в формуле расчета в зависимости от различных поставщиков и приложений, общая модель для основных приложений должна учитывать все части расчета, описанные в разделе 6.2.2. В то же время, в соответствии с неопределенностью и изменчивостью практического использования облака, невозможно сделать точный прогноз общей стоимости. Тем не менее, модель расчета затрат может предоставить детали затрат для каждой части, чтобы помочь пользователям облака четко понять все составляющие потенциальной стоимости (Li и др., 2009) и окупаемость инвестиций (ROI).<sup>1</sup> Кроме того, хорошая расчетная модель должна также иметь возможность делать обоснованные оценки неопределенностей, основанных на типах применения, особенностях и потенциальном использовании.
- Подходящие модели оценки и основные принципы отбора - Для того чтобы сделать разумный выбор, необходимо тщательно рассмотреть множество факторов (например, обеспечение емкости платформы, ограничения по стоимости, требования к применению и особенности). Модель оценки должна быть способна адаптироваться к особенностям и требованиям различных типов приложений. Основные принципы, которые должны учитываться при оценке и отборе, перечислены ниже:
- Удовлетворение всех требований со стороны пользователей облачных сервисов (например, стоимость, типы ОС, конфигурации VM, геолокация).
- Минимизация расходов.
- Максимальное выделение вычислительных мощностей.
- Учитывать предпочтения потребителя (например, конкретные поставщики и типы аренды).
- Учитывать пространственно-временное воздействие (например, потенциальное геолокационное распределение облачных ресурсов и пользователей приложений, частота доступа и распределение времени).

При углубленном рассмотрении пространственно-временных особенностей можно провести дальнейшие исследования для оптимизации пространственно-временных распределений облачных ресурсов (Yang и др., 2011), чтобы пользователи могли повысить производительность и снизить стоимость аренды (например, пытаться избежать ненужных сборов за передачу данных).

<sup>1</sup> См. Microsoft Windows Azure Platform TCO Calculator на <http://www.microsoft.com/brasil/windowsazure/tco/>.

Например, Andrzejak et al. (2010) предложили вероятностную модель, чтобы помочь пользователям оптимально делать ставки на спотовых экземплярах Amazon для достижения различных целей с желаемым уровнем уверенности.

### **6.4.2 Актуальная информация об облачных сервисах**

Правила ценообразования облачных провайдеров часто обновляются на основе стратегий использования и бизнеса. Поддержание цен в актуальном состоянии имеет решающее значение для систем выбора облачных сервисов. Без обновленной информации рекомендуемые решения и их конфигурации ненадежны. Поэтому консультативные системы должны иметь возможность обновлять информацию в режиме, близком к реальному времени, и в автоматизированном процессе после обновления предложений облачных провайдеров. Одним из возможных решений является использование функций уведомления, предоставляемых облачными провайдерами, для получения событий обновления информации и запуска процесса сканирования и обновления. Другие решения включают в себя механизмы обратной связи, позволяющие пользователям и поставщикам обновлять информацию.

### **6.4.3 Функции интерактивности и визуализации консультационного инструмента**

Хорошо продуманная и удобная система консультаций может помочь пользователям облачной обработки данных сократить время и усилия, затрачиваемые на сбор информации, а также сделать мудрый выбор благодаря взаимодействию человека и компьютера. В настоящее время большинство систем и веб-сайтов все еще находятся на ранних стадиях разработки. Сбор и представление информации в облаке являются их основными функциями.

Для интуитивного представления информации и, в конечном счете, для того, чтобы помочь потребителям сделать мудрый выбор, следует использовать более продвинутые функции взаимодействия и визуализации, особенно учитывая популярность визуального анализа. Возможно разработать новый способ проведения визуализации информации/данных и взаимодействия человека и компьютера для решения проблем путем обеспечения синергетического взаимодействия человека и компьютера. Кроме того, система должна также соответствовать требованиям различных уровней пользователей (опытных и неопытных) и помогать им получать результаты с помощью приятного учебного процесса.

## **6.5 КРАТКОЕ ИЗЛОЖЕНИЕ**

В данной главе представлены метрики облачных сервисов и важные критерии оценки и отбора облачных сервисов. Раздел 6.1 рассказывает о важности выбора облачных сервисов. В разделе 6.2 представлены критерии выбора облачных сервисов.

Раздел 6.3 демонстрирует, как выбрать облачный сервис, используя в качестве примера инструмент ESIP по консультированию в области облачной обработки данных. В Разделе 6.4 обсуждаются углубленные соображения и исследования по (1) модели затрат и оценке облачных сервисов, (2) актуальной информации и соответствующих методов сбора и обновления, и (3) функциональной конструкции консультационного инструмента облачных сервисов.

## 6.6 ПРОБЛЕМЫ

1. С какими проблемами сталкиваются пользователи "облака" при выборе облачных сервисов?
2. Каковы основные факторы, влияющие на выбор облачных сервисов?
3. Почему функции и требования к разворачиванию приложения важны для выбора облачного сервиса?
4. Для облачных сервисов IaaS и PaaS какие виды услуг (например, виртуальные машины, хранилища, резервное копирование данных, передача данных) тарифицируются? Какие услуги имеют относительно стабильный размер платы в определенные периоды времени, а какие являются более изменчивыми?
5. Пожалуйста, перечислите измерения для предоставления мощности облачных сервисов.
6. Какие факторы обычно влияют на общую стоимость при использовании VM?
7. Каковы основные принципы при выборе облачного сервиса?
8. Важны ли для выбора облачных сервисов типы облачных хранилищ и места расположения облачной инфраструктуры? Почему?

## СПИСОК ЛИТЕРАТУРЫ

- Andrzejak, A., D. Kondo, and S. Yi. 2010. Decision model for cloud computing under SLA constraints. *Modeling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS), IEEE International Symposium* [http://hal.archives-ouvertes.fr/docs/00/47/48/49/PDF/ec2pricesA\\_final.pdf](http://hal.archives-ouvertes.fr/docs/00/47/48/49/PDF/ec2pricesA_final.pdf) (accessed March 29, 2013).
- Badger, L., T. Grance, R. Patt-Corner, and J. Voas. 2012. Cloud computing synopsis and recommendations. *NIST Special Publication* 800: 146.
- Calheiros, R. N., R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya. 2011. CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Software: Practice and Experience* 41, no. 1: 23–50.
- CSMIC. 2011. Service Measurement Index (Version 1.0). Silicon Valley, Moffett Field, CA: Carnegie Mellon University. [http://csmic.org/wp-content/uploads/2011/09/SMI-Overview-110913\\_v1F1.pdf](http://csmic.org/wp-content/uploads/2011/09/SMI-Overview-110913_v1F1.pdf) (accessed March 12, 2013).
- Goscinski, A. and M. Brock. 2010. Toward dynamic and attribute based publication, discovery and selection for cloud computing. *Future Generation Computer Systems* 26, no. 7: 947–970.

- Haddad, C. 2011. Selecting a Cloud Platform: A Platform as a Service Scorecard. WSO2 White Paper. <http://wso2.org/library/whitepapers/2011/12/selecting-cloud-platform-platform-service-scorecard> (accessed March 29, 2013).
- Jansen, W. and T. Grance. 2011. Guidelines on security and privacy in public cloud computing. *NIST Special Publication 800*: 144.
- Kulkarni, P. 2012. Guidelines for Selecting Cloud Provider and Determining Cloud Type. <http://blog.harbinger-systems.com/2012/02/guidelines-for-selecting-cloud-provider-and-determining-cloud-type> (accessed March 12, 2013).
- Li, X., Y. Li, T. Liu, J. Qiu, and F. Wang. 2009. The method and tool of cost analysis for cloud computing. In *Cloud Computing. CLOUD'09. IEEE International Conference*, pp. 93–100.
- Martens, B., F. Teuteberg, and M. Gräuler. 2011. Design and implementation of a community platform for the evaluation and selection of cloud computing services: A market analysis. *ECIS Proceedings*, 215.
- NIST. 2012. NIST Cloud Computing Reference Architecture Cloud Service Metrics Description (Draft). [http://collaborate.nist.gov/wiki-cloud-computing/pub/CloudComputing/RATax\\_CloudMetrics/RATAX-CloudServiceMetricsDescription-DRAFT-v1.1.1.pdf](http://collaborate.nist.gov/wiki-cloud-computing/pub/CloudComputing/RATax_CloudMetrics/RATAX-CloudServiceMetricsDescription-DRAFT-v1.1.1.pdf) (accessed March 12, 2013).
- Repschläger, J., S. Wind, R. Zarnekow, and K. Turowski. 2011. Developing a cloud provider selection model. In *Proceedings of Enterprise Modelling and Information Systems Architectures: Proceedings of the 4th International Workshop on Enterprise Modelling and Information Systems Architectures. EMISA*. Hamburg, Germany, September 22–23. (GI publishes recent findings in informatics [i.e., computer science and information systems] as a series to document conferences that are organized in co-operation with GI and to publish the annual GI Award dissertation), p. 163.
- Rodrigues, T. 2012. Comparing Cloud Infrastructure-as-a-Service Providers (11 cloud IaaS providers compared). <http://www.techrepublic.com/blog/datacenter/11-cloud-iaas-providers-compared/5285> (accessed March 12, 2013).
- Stantchev, V. 2009. Performance evaluation of cloud computing offerings. In *Advanced Engineering Computing and Applications in Sciences. ADVCOMP'09. 3rd International Conference*, pp. 187–192.
- Yang C., M. Goodchild, Q. Huang, D. Nebert, R. Raskin, M. Bambacus, Y. Xu, and D. Fay. 2011. Spatial cloud computing—How can geospatial sciences use and help to shape cloud computing? *International Journal of Digital Earth* 4, no. 4: 305–329.

## Часть III

---

# Облачные проекты в области геонаук

---

В данной части сначала рассказывается, как использовать облачные GIS на примере ArcGIS в "облаке" (глава 7), а затем на трех практических примерах демонстрируется, как можно использовать сложные геонаучные приложения с поддержкой облачной обработки данных: 1) проект "Информационный центр GEOSS" используется для демонстрации возможностей облачной обработки данных с использованием баз данных, пространственного индекса и пространственных технологий веб-портала (глава 8); 2) проект Climate@Home используется для демонстрации возможностей моделирования геонаучных моделей, поддающихся облачной обработке данных (глава 9); и 3) проект "Прогнозирование пыльных бурь" используется для демонстрации возможностей использования гибких облачных ресурсов для поддержки вычислений, вызывающих резкие скачки напряжения в результате сбоя в работе (глава 10).



## Глава 7

---

# Использование arcGIS в облаке

*Манжу Ю, Пинде Фу, Наньинь Чжоу и  
Цзичжэ Ся*

---

В настоящей главе в качестве практического примера рассматривается набор продуктов ArcGIS1 в облаке для демонстрации того, как использовать облачную геоинформационную систему (GIS). На примере практического использования читатели могут получить четкое представление о текущем состоянии облачных GIS.

## 7.1 ВВЕДЕНИЕ

### 7.1.1 Для чего географической информационной системе необходимо облако

Многие проблемы XXI века требуют своевременной интеграции огромных массивов геопространственной информации через Географическую информационную систему (GIS) для решения глобальных и региональных проблем, таких как реагирование на чрезвычайные ситуации и планирование. Своевременная интеграция больших объемов информации требует готовности вычислительной инфраструктуры со следующими характеристиками: достаточные вычислительные возможности, минимальные энергозатраты, быстрое реагирование и широкая доступность для населения. Традиционное программное обеспечение для GIS, такое как ArcGIS, работает на настольных компьютерах и локальных серверах и ориентировано на отдельных пользователей, в основном экспертов по GIS. Оно требует установки и обслуживания как аппаратного, так и ПО, и не поддерживает потребности в крупномасштабном параллельном доступе. Облачная обработка данных предоставляет вычислительные возможности для построения и развертывания GIS как сервиса, который можно назвать облачными GIS (Mann 2011). Появление облачных GIS обусловлено необходимостью установления новых режимов обслуживания, использования и биллинга, которые могут решить существующие вычислительные проблемы и удовлетворить потребности более широкой базы пользователей. Технологии и архитектура, которые могут предложить облачную обработку данных, являются ключевыми областями исследований и разработок для решений в области GIS (Kouyoumjian 2010). Новые GIS-решения, такие как ArcGIS Online (представленная в разделе 7.2.1), были реализованы путем предоставления GIS в качестве облачных сервисов (Bhat et al. 2011).

---

<sup>1</sup>См. Esri на <http://www.esri.com/software/arcgis>.

112 Манжу Ю, Пинде Фу, Наньинь Чжоу и Цзичжэ Ся

---

Облачные GIS могут предоставлять пользователям (1) отсутствие установки и обслуживания программного обеспечения, (2) неограниченные вычислительные ресурсы и пространство для хранения данных, и (3) услуги по запросу. Кроме того, облачные GIS расширяет область применения GIS от географии до различных социальных и бизнессфер.

### **7.1.2 Примеры GIS, которым необходимо облако**

Облачные GIS могут более эффективно поддерживать крупномасштабный одновременный доступ и требования к вычислительным ресурсам, такие как помощь в реагировании на чрезвычайные ситуации. Облачные GIS удовлетворяют потребности в управлении чрезвычайными ситуациями за счет обеспечения своевременности, интерактивности, доступности и сотрудничества. Например, в 2010 году, когда масштабные наводнения охватили три четверти штата Квинсленд, Австралия, учреждения по реагированию нуждались в оперативном доступе к информации в связи с быстро меняющейся ситуацией. Австралийская компания Esri в течение 12 часов создала приложение общей оперативной картины (COP) на основе GIS, которое предоставило доступ к самой свежей и точной информации о ситуации в Брисбене. Другой успешный сценарий заключается в том, что облачный GIS поддерживает различные социальные и деловые сферы на уровне предприятия для создания решений, которые помогают клиентам, в том числе риск-менеджерам в таких организациях, как страховые и финансовые компании. Например, ArcGIS Online помогла Wall Street Network развернуть планы по обеспечению непрерывности бизнеса и ликвидации последствий катастроф даже после сбоя, вызванного наводнением в результате супербури Сэнди (Richardson 2012).

## **7.2 ИСПОЛЬЗОВАНИЕ arcGIS В ОБЛАКЕ**

В настоящее время ArcGIS в облаке включает в себя ArcGIS Online, ArcGIS for Server, ПО GIS как сервис и Мобильный сервис GIS. ArcGIS Online - это предварительно настроенная платформа, доступ к которой можно получить через веб-браузер. ArcGIS for Server - это сервер GIS, который может быть развернут на облачных сервисах. ПО GIS как сервис - это облачное программное обеспечение, которое может быть использовано в качестве сервиса. Мобильный GIS-сервис - программное обеспечение GIS как сервис для мобильных операций.

### **7.2.1 arcGIS online**

ArcGIS Online<sup>1</sup> - это облачная система для веб-картографии и управления географической информацией, которая расширяется и трансформируется в полноценное приложение Software as a Service (SaaS) (Программное обеспечение как услуга). ArcGIS Online предоставляет возможность быстро и легко получить представление о данных без установки и настройки программного обеспечения для GIS. ArcGIS Online предлагает GIS как сервис, включающий в себя интуитивно понятные инструменты для создания и публикации карт и приложений по запросу, богатую коллекцию базовых карт, демографических карт, снимков и других данных.

<sup>1</sup> См. ArcGIS на <http://qaext-ds.arcgis.com/about/>.

Удобно обмениваться картами, настроенными в ArcGIS Online, через блоги, веб-страницы, приложения и Facebook или Twitter. Пользователи могут создавать, размещать и обмениваться данными и картографическими сервисами с помощью ArcGIS Online, поддерживаемых безопасным облаком.

### 7.2.1.1 Функциональные возможности

Пользователи могут создавать веб-карты с помощью картографических онлайн-сервисов ArcGIS и геосервисов, создавать пользовательские веб-приложения и мобильные приложения с помощью ArcGIS Web Mapping API и ArcGIS Mobile Runtime Software Development Kits (SDK), а также добавлять созданные карты в ArcGIS for Desktop. Существует два основных варианта веб-сервиса: Хостинговые сервисы и контент-сервисы (Geo-related 2012).

1. *Хостинговые сервисы*—Хостинговые сервисы - это сервисы ArcGIS Server, размещаемые и управляемые компанией Esri как часть сервиса ArcGIS.com. Существует два типа хостинговых сервисов:
  - Hosted Feature Service-Сервис поддерживает запрос и редактирование функций в качестве векторов.
  - Hosted Tile Map Service-Сервис предоставляет предварительно сгенерированные (кэшированные) изображения фрагментов карт. Также позволяет пользователям загружать данные и генерировать изображения фрагментов карт, а также размещать данные на хосте.
2. *Контент сервисы*:
  - Map Services - Сервисы карт предоставляют справочно-информационные услуги с использованием данных, размещенных на сервере Esri. SOAP, REST и Rich Internet Application (RIA) контролирует/картирует API приложений:
    - i. TopographicMapServices - данные сервисы предоставляют базовые карты, демографические, справочные и другие.
    - ii. ImageServices- предоставляют данные глобальной съемки землив виде фрагментов.
    - iii. Bing Maps - карты основанные на Bing.
  - *Task Services*—Сервисы по выполнению задач доступны через SOAP и REST в API контроля/картографирования приложений RIA:
    - i. Geosearch - сервис геоисследования обеспечивает поиск функций или точек интереса для поддержки локализации карты.
    - ii. Geocoding - сервис геокодирования обеспечивает как прямое, так и обратное геокодирование адресов, достопримечательностей и административных точек. Поддерживает транзакционные задачи и пакеты. Пакеты могут быть сгенерированы или опубликованы либо с возвратом результатов, либо с предоставлением их непосредственно в виде слоя карты. Сервис охватывает США, Канаду и Европу на основе данных Tele Atlas.
    - iii. Routing - Сервис маршрутизации предоставляет маршруты и инструкции с поддержкой нескольких языков. Сервис охватывает США, Канаду и Европу. Услуга маршрутизации основана на данных Tele Atlas.

iv. *Geometry* - широко используется в геометрических операциях, таких как смена CRS, упрощение, буфер, площадь, длина, точка метки, выпуклый корпус, разрез, растяжение, связь, автозаполнение, разность, обобщение, пересечение, смещение, изменение формы, обрезка/расширение и соединение.

## 7.2.2 arcGIS для сервера

ArcGIS for Server<sup>1</sup> это сервер GIS, который может быть развернут на Amazon EC2 и Vblock Виртуальной вычислительной среды (VCE), что позволяет пользователям размещать GIS-ресурсы на своих ArcGIS для серверных систем. Также позволяет клиентским приложениям, таким как картографические веб-приложения и мобильные устройства, использовать и взаимодействовать с ресурсами. ArcGIS for Server поддерживает различные платформы ОС, включая Microsoft Windows Server, Red Hat Enterprise Linux AS/ES, SUSE Linux Enterprise Server и Ubuntu. Существует два важных компонента, связанных с ArcGIS for Server: ArcGIS Server AMI и ArcGIS Server Cloud Builder.

AMI сервера ArcGIS - предварительно настроенные шаблоны AMI. С помощью AMI пользователи могут быстро настроить полнофункциональный экземпляр ArcGIS Server на Amazon EC2. Существующие пользователи ArcGIS Server могут запросить доступ к AMI ArcGIS Server через службу поддержки клиентов Esri, предоставив свой идентификатор учетной записи Amazon EC2. Затем экземплярами ArcGIS Server можно управлять, как и любыми другими экземплярами Amazon EC2.

ArcGIS Server Cloud Builder, загружаемое настольное приложение, предоставляет пользователям различные варианты запуска ArcGIS для серверных сайтов на Amazon EC2. Данное приложение позволяет пользователям динамически масштабировать сайты в зависимости от потребностей и создавать резервные копии ArcGIS для серверных сайтов (Chappell 2010).

### 7.2.2.1 Функциональные возможности

- Централизованное управление и доставка GIS-сервисов - каждый пользователь имеет свой собственный сайт ArcGIS for Server, где он может централизовать управление всеми сервисами по картографии, снимкам, глобусам, геокодированию, управлению геоданными и многим другим. Функции GIS, такие как веб-редактирование, сетевой анализ и схемы, а также моделирование, статистика и другие гео-аналитические инструменты, также могут быть предоставлены.
- Ответ потребованию для карт и инструментов GIS - Архитектура ArcGIS для сервера позволяет пользователям быстро масштабировать GIS-системы для удовлетворения пиковых нагрузок путем добавления и удаления GIS-серверов по мере необходимости в их собственной инфраструктуре, в виртуальных средах и в облаке.
- Интеграция с корпоративными приложениями - сервисы, созданные и управляемые с помощью ArcGIS for Server, могут быть интегрированы с корпоративными приложениями - предоставляя конечному пользователю возможность использования GIS без дополнительных усилий.

<sup>1</sup>См. Esri на <http://www.esri.com/software/arcgis/arcgisserver>.

Разработчики имеют доступ к всеобъемлющим API веб-картам и SDK мобильного исполнения, которые могут быть использованы для создания удивительных картографических приложений или интегрировать карты и возможности GIS в существующие веб-приложения, мобильные и настольные приложения.

### 7.2.3 Программное обеспечение GIS как сервис

Программное обеспечение GIS как услуга предоставляет облачным клиентам доступ к приложениям, которые могут легко решать сложные задачи с помощью инструментов и данных GIS. Данное ПО интегрирует большие объемы географической информации и делает их доступными через простые в использовании веб-приложения. Эти приложения тесно связаны как с областями домена (например, геобизнес, политика и картография сообществ), так и с GIS-сообществом в целом. Появление программного обеспечения GIS как услуги призвано удовлетворить потребности пользователей в использовании GIS в сервисной среде без приобретения данных или технологии.

#### 7.2.3.1 Функциональные возможности

- *ArcLogistics*<sup>1</sup> это облачное приложение для создания оптимизированных маршрутов за меньшее количество времени. Это приложение построено на Amazon Web Services (AWS), логика его работы выполняется в виртуальной машине Amazon EC2 VM, а данные хранятся в S3 (Amazon Simple Storage Service). ArcLogistics - это не веб-браузер, а настраиваемое приложение для Windows, построенное на базе Windows Presentation Foundation. Пользователи могут ввести количество автомобилей и необходимых остановок, а затем получить обратно оптимизированную маршрутизацию для этих автомобилей.
- *Business Analyst Online*<sup>2</sup> это "облачное" приложение для работы с демографическими данными, данными пользователей и другой информацией. Предоставляя демографическую информацию и другие данные, наряду с инструментами для работы с вышеперечисленными данными, это "облачное" приложение может помочь владельцам бизнеса, планировщикам, риэлторам и другим лицам принимать лучшие решения.
- Доступ к Business Analyst Online можно получить через обычный веб-браузер. Приложение также предоставляет интерфейс веб-службы, который может быть использован другими клиентами. Например, предоставляется соответствующее приложение для iPhone, чтобы третьи лица также могли получить доступ к этим сервисам. Это приложение работает в собственных центрах данных Esri с доступом в Интернет вместо AWS или других облачных сервисов.
- *Community Analyst*<sup>3</sup> это веб-система, позволяющая пользователям просматривать и анализировать демографические, общественные и сторонние источники данных для лучшего понимания всего сообщества и принятия лучших политических решений.

<sup>1</sup> См. Esri, ArcLogistics Suite на <http://www.esri.com/software/arclogistics-suite>.

<sup>2</sup> См. Esri, Business Analyst Online на <http://www.esri.com/software/bao>.

<sup>3</sup> См. Esri, Community Analyst на <http://www.esri.com/software/arcgis/community-analyst>.

Одной из отличительных особенностей является то, что аналитик сообщества использует последние данные, так что он может получить информацию сразу же для точной области, что пользователям нужно, в том числе стандартные географии (вплоть до уровня группы блоков переписи), нарисованные вручную формы, кольца, или время диска вокруг местоположения.

Аналитик сообщества доказал свою эффективность и значительную пользу. Например, Отдел эпидемиологии и оценки программ Департамента общественного здравоохранения округа Риверсайд сделал карты для каждого соответствующего сотрудника этого департамента (ArcNews 2011/2012). Данные карты используются для пропаганды и охраны здоровья жителей и посетителей графства, а также для обеспечения доступности услуг, способствующих повышению благосостояния общества. В соответствии с результатами, полученными с помощью карт, аналитик сообщества обеспечивает удовлетворительную работу и понимание для пользователей.

#### **7.2.4 Мобильные сервисы GIS**

Мобильные устройства являются важной частью облачных GIS, служащим в качестве терминала для отображения, редактирования, сбора и анализа геопространственных данных. Растущая популярность и широкое использование мобильных устройств расширили сферу применения облачных GIS. С помощью мобильных приложений, таких как ArcGIS для Windows Mobile и Windows Tablet, ArcGIS для смартфонов и планшетов, отдельные пользователи и организации могут получить доступ к функциям, предоставляемым сервером (например, ArcGIS для сервера). Мобильные GIS расширяют возможности GIS и позволяют организациям принимать точные бизнес-решения в реальном времени и сотрудничать как в полевых, так и в офисных условиях. Данный сервис повышает эффективность и точность полевых операций, обеспечивает быстрый сбор данных и беспровную интеграцию данных, заменяет бумажный документооборот и помогает принимать своевременные и обоснованные решения.

Для сотрудников находящихся в движении - "облако" предлагает лучшую мобильность для повышения производительности труда и совместной работы. К совместно используемым данным и приложениям в облаке можно получить мгновенный доступ для обнаружения, просмотра, редактирования и сохранения изменений, а также для вызова функций геопротессорной обработки для получения результатов по требованию.

#### **7.2.5 Краткое изложение раздела**

В данном разделе представлены четыре продукта ArcGIS в облаке: ArcGIS Online, ArcGIS for Server on the cloud, GIS Software as a Service и мобильный GIS-сервис. Мощные функции и сервисы GIS поддерживаются различными облачными продуктами GIS и доставляются пользователям как по проводным, так и по беспроводным сетям. Такие продукты предоставляют функции и сервисы GIS, к которым можно быстро и легко получить доступ, ничего не устанавливая и не настраивая, что очень важно в эпоху Интернета и мобильной связи. В следующем разделе представлены три варианта использования различных продуктов ArcGIS в облаке.

Данные методы внедрения подробно демонстрируют, как облако могло бы лучше поддерживать GIS и как продукты GIS могли бы обслуживать пользователей и их приложения.

## 7.3 МЕТОД ВНЕДРЕНИЯ

### 7.3.1 Региональный анализ штата Орегон с использованием ArcGIS Online

В настоящее время городское планирование и управление стало важной прикладной областью GIS. Правительствам, особенно администраторам городского планирования, необходимо осуществлять контроль над городским и региональным строительством, чтобы сделать город более пригодным для жизни и красивым, а также найти наилучшие решения в случае возникновения чрезвычайных ситуаций.

На карте в данном методе внедрения показан региональный анализ части штата Орегон, сочетающий рост городской территории с сопутствующей инфраструктурой, природными элементами и другой информацией (рис. 7.1). Например, городская территория в части штата Орегон в 2000 году отмечена желтым цветом, а в 2010 году - желтым и зеленым (см. цветную вставку). На данной карте четко виден рост городов в этой области. Кроме того, различные символы представляют собой различные элементы. Например, коричневые линии изображают основные дороги; форма, которая выглядит как дерево, изображает Park & Ride. Следующие шаги иллюстрируют, как создать, сохранить и поделиться картой, сделанной на ArcGISOnline.



Рисунок 7.1 (См. цветную вставку) Региональный анализ карты штата Орегон.

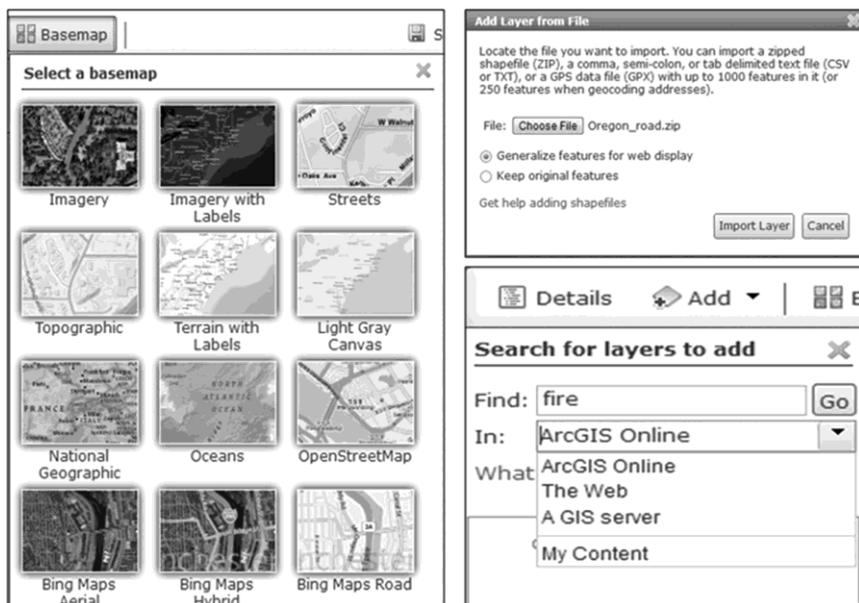


Рисунок 7.2 (См. цветную вставку) Базовые карты и данные, предлагаемые ArcGISOnline.

*Шаг 1. Выбрать базовые карты* - убедитесь, что базовая карта выбрана на основе функций и характеристик карты. Фоном для данных служит карта улиц (рис. 7.2), одна из базовых карт ArcGIS Online, которая хорошо подходит для наложения наборов данных. В дополнение к картам улиц можно также использовать океанские, топографические карты и карты изображений

*Шаг 2. Добавление слоев* - Слоями могут быть те, которые были опубликованы в Интернете, например, через ArcGIS for Server или Open Geospatial Consortium, Inc., веб-картографические сервисы, слои KML или CSV, или личные данные, хранящиеся в виде текстовых файлов с разделителями, в формате обмена GPS или в виде шейп-файлов.

*Шаг 3. Добавить данные* - основные данные, используемые в данном случае, представляют собой распределение городских территорий штата Орегон как в 2000, так и в 2010 году, извлеченные из Бюро переписи населения США.<sup>1</sup> Шаги для получения данных: (1) Найдите дороги на странице шейп-файлов 2010 и выберите Орегон от основных и второстепенных дорог; (2) Найдите Уотер и выберите Орегон из области гидрографии для округов: округ Клакамас, округ Мэрион, округ Малтнома, округ Вашингтон и округ Ямхилл; (3) Найдите городскую зону прироста и выберите Орегон из городской зоны прироста (2010) и городской зоны прироста (2000); и (4) Загрузить все выбранные файлы.

<sup>1</sup>См. U.S. Census Bureau на <http://www.census.gov/cgi-bin/geo/shapefiles2010/main>.

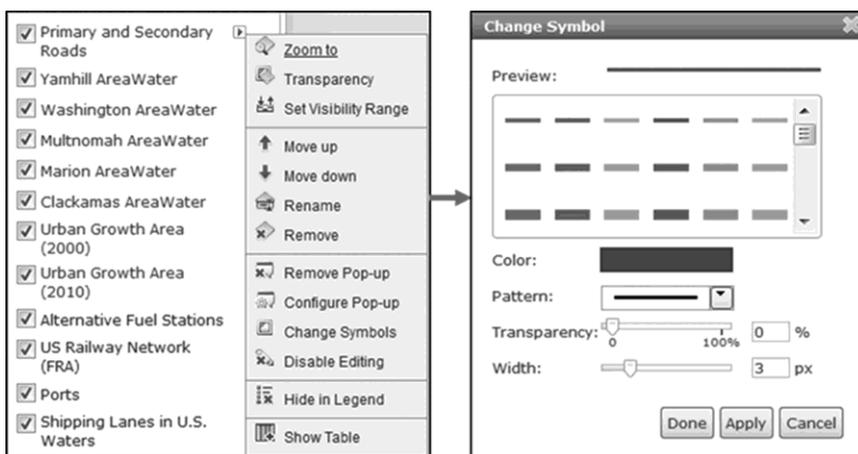


Рисунок 7.3 (См. цветную вставку) Модификация символа основного и второстепенного пути.

Для проведения обширного анализа также требуются соответствующие экологические данные, которые можно найти в онлайн массивах данных GIS, таких как железнодорожные сети, парк и аттракционы, национальные парки и заправочные станции.

*Шаг 4. Модификация символов* - данный шаг заключается в модификации символов существующих слоев для лучшего отображения. Пользователи могут изменить символ Первичного и Вторичного пути на коричневую и сплошную линию шириной 3px (Рисунок 7.3), а также символ различных Зональных Водных слоев с различными синими цветами. Они также могут отображать слой "Зона прироста 2000" желтым цветом и слой "Зона прироста 2010" зеленым цветом. Установите все остальные слои по умолчанию. (См. цветную вставку)

*Шаг 5. Сохранить карту* - После создания карты пользователи могут сохранить ее, добавить описание и прикрепить теги к карте. Подробное описание и прикрепленные теги обеспечат контекст и помогут другим пользователям найти вашу карту. Карта хранится в разделе "My Content" учетной записи Esri пользователей. Пользователи могут выбирать, делиться ли ей в сети или нет (Рисунок 7.4).

*Шаг 6. Размещение и совместное использование* - ArcGIS Online Sharing позволяет пользователям размещать свои геопространственные карты, слои и инструменты в сообществе ArcGIS и обмениваться ими, а также создавать частные группы для обмена контентом, связанным с конкретным проектом или общей деятельностью.

Данный метод внедрения демонстрирует, как ArcGIS Online готов к использованию и доставке в качестве сервиса, а также знакомит с интуитивно понятными инструментами и богатой коллекцией базовых карт и данных для создания и публикации карт и приложений по требованию.



Рисунок 7.4 Карта хранится в разделе "My Content" учетной записи Esri.

## 7.3.2 Методы внедрения arcGIS for server

### 7.3.2.1 Единая картина обстановки при наводнении, муниципалитета г.Брисбен

Единая картина обстановки наводнения (COP) муниципалитета г. Брисбен (BCC), созданная с помощью ArcGIS Server на Amazon EC2, была разработана сотрудниками Esri Australia Pty. Ltd.<sup>1</sup> Данная система помогает агентствам реагировать на наводнения и помогает жителям пострадавших районов получать доступ к текущей информации о развивающейся ситуации. Подобные работы проводились во время многих стихийных бедствий в 2010 году, включая разлив нефти Deep Horizon в Мексиканском заливе и землетрясения у побережья региона Мауле в Чили и недалеко от Порт-о-Пренса, Гаити. GIS и ИТ-специалисты быстро создали веб-картографическое GIS-приложение в ответ на огромный спрос на информацию о каждом событии. Доступ к данным веб-сайтам показал, что, помимо аварийно-спасательных служб и государственных служащих, общественность также заинтересована в своевременных, точных интерактивных картах. Эта ситуация выявляет требования к параллельному доступу и вычислительным возможностям обработки.

Карта наводнений BCC компилировала данные о наводнениях по всему Брисбену, пострадавшему от стихийного бедствия, - например, пик наводнений, закрытые дороги и центры эвакуации - на карту, чтобы обеспечить исчерпывающий сценарий наводнения в реальном времени. Важным аспектом карты наводнения BCC является функция, которая позволяла пользователям включать и выключать информационные слои по мере необходимости, такие как повреждение имущества и расположение центров эвакуации.

<sup>1</sup> См. Esri, ArcWatch на <http://www.esri.com/news/arcwatch/0611/feature2.html>.

Во время наводнений команда Esri Australia постоянно обновляла карту наводнений ВСС, используя информацию, полученную от местных и государственных органов власти, а также от аварийных бригад на местах.

Облачная GIS является ключом к успеху этого приложения, потому что это тесно интегрированная система компонентов и контента, которую можно развернуть вне помещения, в течение нескольких часов и можно управлять удаленно. При возникновении серьезной чрезвычайной ситуации у специалистов по GIS не будет времени или ресурсов, чтобы отключить существующее оборудование и программное обеспечение и перепрофилировать их для поддержки одного веб-приложения, обслуживающего десятки тысяч посетителей ежедневно. Пользователям необходим доступ к резервной машине, которая настроена и готова к работе в течение нескольких минут. Им также нужны огромные корпоративные ресурсы, которые могут мгновенно справляться с резкими скачками спроса, а также автоматически уменьшать масштаб по мере уменьшения веб-трафика.

Данный пример использования является одним из многих успешных примеров использования "облачных" GIS в интересах управления чрезвычайными ситуациями. Такого рода приложения или услуги эффективно решают проблему передачи информации о каждой чрезвычайной ситуации людям, вовлеченным в нее за короткий период времени.

### **7.3.2.2 Программа просмотра парков штата Пенсильвания**

Pennsylvania State Parks Viewer - это новая интерактивная карта Департамента охраны природы и природных ресурсов Пенсильвании для поиска информации о государственном парке Пенсильвании. Как и многие другие приложения, она предоставляет собой различные базовые карты, такие как Национальная географическая карта мира, Карта улиц мира и Карта аэрофотосъемки. Существуют также слои, предлагающие другую географическую информацию, например, государственные границы, туристические регионы, границы округов и водосборные бассейны.

В данном приложении есть три режима для переключения: Государственные парки, государственные леса и геология. Каждый режим предоставляет различные виды информации о парках и лесах.

Если родители хотят взять своих детей в поход или на пикник, они могут выбрать поиск по Государственным паркам (рис. 7.5) и выбрать несколько каталогов, таких как Camping, Education Program, Picnicking, Sightseeing (Кемпинг, Образовательная программа, Пикник и Достопримечательности). Затем они могут получить несколько результатов по рекомендуемым паркам с их названиями, площадью и местоположением. Если выбрать один из них, то подробная информация (рис. 7.6) появится на крупномасштабной карте вместе с направлениями от начального местоположения.

Данное приложение также имеет некоторые полезные инструменты для пользователей, чтобы легко извлекать данные, добавлять данные в ArcMap и Google Earth, создавать PDF-файлы и печатать. Другие функции включают в себя измерение расстояния, широты/долготы координат, аннотации и прогноз погоды. Пользователи также могут делать закладки, обозначения и запросы о направлениях движения.

Вышеописанный пример использования демонстрирует, что помимо поддержки управления чрезвычайными ситуациями, облачные GIS могут также помочь людям планировать досуг с удобством.

122 Манжу Ю, Пинде Фу, Наньинь Чжоу и Цзичжэ Ся

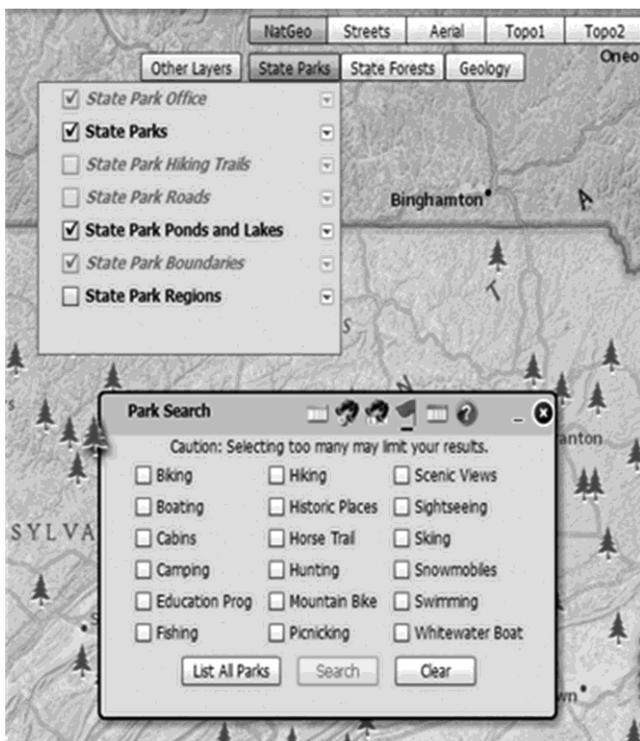


Рисунок 7.5 Поиск парков в режиме State Parks.

### 7.3.3 Краткое изложение раздела

В настоящем разделе представлены три варианта использования ArcGIS Online и ArcGIS for Server. Такие варианты использования демонстрируют, как облачная GIS может (1) использовать виртуализированные вычислительные ресурсы для предоставления веб-сервисов (например, картографические сервисы и геосервисы), (2) справляться с резкими скачками спроса и (3) предоставлять предварительно установленные и настроенные виртуальные машины, так что сервер будет готов к использованию в течение нескольких минут.

## 7.4 КРАТКОЕ ИЗЛОЖЕНИЕ

В данной главе представлено несколько практических примеров облачной GIS. В разделе 7.2 представлены продукты пакета ArcGIS в облаке: ArcGIS Online, ArcGIS for Server, ПО GIS как услуга и Мобильный GIS-сервис. В разделе 7.3 показаны три варианта использования для дальнейшего ознакомления с различными аспектами облачной GIS, такими как региональный анализ, управление чрезвычайными ситуациями и рекомендации инструментов для организации отдыха.

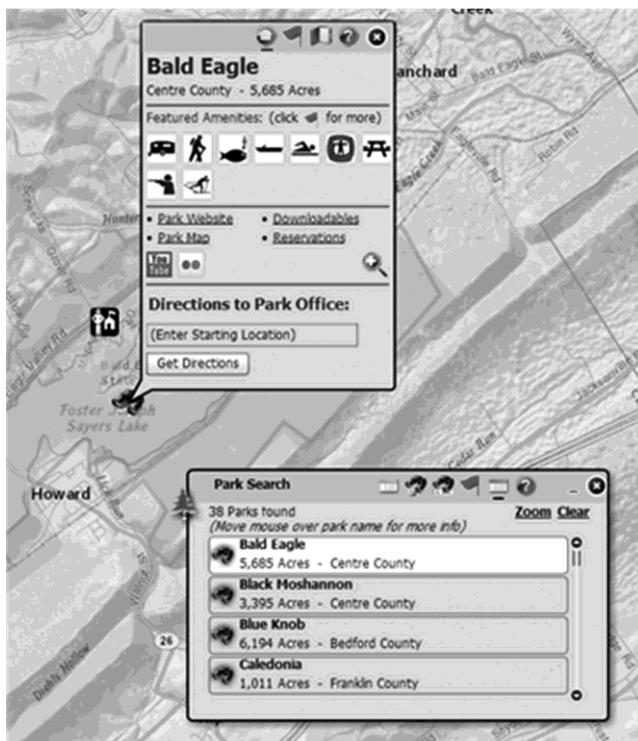


Рисунок 7.6 Подробная информация об одном объекте.

Такие практические примеры дают практический опыт использования облачных GIS и создания потенциальных решений в области GIS для задач XXI века.

## 7.5 ПРОБЛЕМЫ

1. Какие приложения могут использовать преимущества облачных GIS?
2. Каковы четыре компонента ArcGIS в облаке? Что делает их эксклюзивными?
3. Как вы думаете, ArcGIS Online уже является SaaS-приложением? Почему?
4. Какие функции ArcGIS for Server отличают его от локальных серверов?
5. Какие функции ArcGIS Online отличаются от ArcGIS Desktop?

## СПИСОК ЛИТЕРАТУРЫ

- ArcNews. 2011/2012. Opening Up Health Reform: Riverside County's Public Health Department Takes a New Look at Grant Applications. Winter. <http://www.esri.com/news/arcnews/winter1112/articles/opening-up-health-reform.html> (accessed March 29, 2013).
- Bhat, M. A., R. M. Shah, and B. Ahmad. 2011. Cloud computing: A solution to geographical information systems (GIS). *International Journal on Computer Science and Engineering* 3, no. 2: 594–600.
- Chappell D. 2010. GIS in the Cloud: The Esri Example. Esri White Paper. <http://www.esri.com/library/whitepapers/pdfs/gis-in-the-cloud-chappell.pdf> (accessed March 29, 2013).
- Georelated.com. 2012. Geocloud: ESRI ArcGIS Online Reviewed. <http://www.georelated.com/2012/10/cloud-web-mapping-service-api-review.html> (accessed March 29, 2013).
- Kouyoumjian, V. 2010. GIS in the Cloud: The New Age of Cloud Computing and GIS. Esri White Paper. <http://www.esri.com/library/ebooks/gis-in-the-cloud.pdf> (accessed March 29, 2013).
- Mann, K. 2011 (Spring). Cloud GIS. Esri. <http://www.esri.com/news/arcuser/0311/cloud-gis.html> (accessed March 29, 2013).
- Richardson, K. 2012. ArcGIS Online Helps Wall Street Network Do Business in the Cloud after Superstorm Sandy: From Reactive to Proactive. *ArcWatch: GIS News, Views, and Insights*. <http://www.esri.com/news/arcwatch/> (accessed March 29, 2013).

## Глава 8

---

# Информационный сервис GEOSS с поддержкой облачных технологий

*Кай Лю, Дуглас Неберт, Цюньин Хуан, Цзичжэ Ся и Чжэньлун Ли*

---

GEOSS (Глобальная система систем наблюдения за Землей)<sup>1</sup> Информационный сервис (CLH) представляет собой комплексную систему управления данными и является основным компонентом GEOSS, который обеспечивает возможности сбора и поиска для геопространственных компонентов и сервисов, зарегистрированных в GEOSS. В данной главе рассказывается, как включить облачную среду CLH с развертыванием и настройками, необходимыми для запуска CLH в облачном сервисе Amazon EC2.

### 8.1 ИНФОРМАЦИОННЫЙ СЕРВИС GEOSS: ИСХОДНЫЕ ДАННЫЕ И СЛОЖНОСТИ

#### 8.1.1 Исходные данные

Различные системы наблюдения Земли (EO) контролируют планету Земля и ежедневно генерируют большие объемы данных EO. Наблюдения предоставляют базовые данные о Земле с конкретными временными снимками, траекториями и событиями. Такие данные полезны для поддержки различных областей социальной выгоды (SBA), включая сельское хозяйство, биоразнообразие, климат, бедствия, экосистемы, энергетику, здоровье, воду и погоду, определенные межправительственной группой по наблюдению за Землей (GEO) (GEO 2009). Признавая ценность данных EO и требований к глобальному распространению, GEO была сформирована для создания общей инфраструктуры GEOSS (GCI), которая помогает управлять, интегрировать, получать доступ и совместно использовать глобальные данные EO для решения мировых и региональных проблем (Christian 2005).

CLH - это двигатель, который управляет всем GCI. Он был разработан и поддерживается Центром интеллектуальных пространственных вычислений для науки о воде и энергии (CISC) Университета Джорджа Мейсона в сотрудничестве с Геологической службой США (USGS). CLH создает мост между поставщиками данных EO и конечными пользователями GEOSS путем (1) сбора метаданных данных EO из нескольких источников данных с помощью механизмов сбора и (2) предоставления возможностей локального и удаленного поиска, позволяющих пользователям обнаруживать данные EO (Liu et al. 2011).

---

<sup>1</sup> См. Group on Earth Observations на <http://www.earthobservations.org/index.shtml>.

## 8.1.2 Сложности

В качестве примера пространственных веб-порталов (Yang et al. 2007) CLH разработан для поддержки доступа во всем мире к ресурсам пространственных данных. При разработке CLH учитываются несколько проблем:

- **Большие данные.** Большие данные - сложная задача в CLH по отношению к трем V:
  1. *Volume (Объем)* — GEOSS - это всемирная система, содержащая большой объем метаданных для огромного количества данных EO, собранных из широкого распределения каталогов в GCI. После того, как поставщики данных EO зарегистрируют свои каталоги данных или услуги в реестре GEOSS, CLH будет собирать такие метаданные постоянно с временным интервалом (например, один час или один день) в зависимости от частоты обновления источников данных. К январю 2013 года в CLH было зарегистрировано 104 каталога и 120 000 записей метаданных. Вышеописанные данные большие и быстро растут.
  2. *Velocity (Скорость)* — Частое обновление наборов данных и метаданных.
  3. *Variety (Диапазон)* — Форматы / стандарты метаданных (FGDC CSDGM<sup>1</sup> Metadata Standards, Dublin core,<sup>2</sup> и ISO-19139<sup>3</sup>) и протоколы доступа, например, Сервис каталогов для Интернета (CSW),<sup>4</sup> Поиск / получение по URL (SRU),<sup>5</sup> и Really Simple Syndication (RSS),<sup>6</sup> разные.
- **Пространственно-временной поиск и полнотекстовый поиск** — Возможность поиска - самая важная функция для CLH. Однако в CLH нелегко обеспечить высокопроизводительные функции поиска, поскольку объем данных огромен, а каждая запись содержит как пространственно-временную, так и текстовую информацию. Сочетание пространственно-временного поиска и полнотекстового поиска резко увеличит вычислительную сложность. Метаданные в CLH фактически являются пространственно-временными метаданными, которые не только хранят буквальное описание данных EO, но также связаны с пространственно-временной информацией. Чтобы обнаружить применимые данные из геопространственных метаданных в CLH, пользователям необходимо использовать «Что» для поиска информации по описанию, «Где» чтобы найти географическую протяженность, и «Когда» для указания временной информации.

<sup>1</sup> См. Federal Geographic Data Committee на <http://www.fgdc.gov/metadata/csdgm/>.

<sup>2</sup> См. Dublin Core Metadata Initiative на <http://dublincore.org/>.

<sup>3</sup> См. ISO на [http://www.iso.org/iso/catalogue\\_detail.htm?csnumber=32557](http://www.iso.org/iso/catalogue_detail.htm?csnumber=32557).

<sup>4</sup> См. OGC на <http://www.opengeospatial.org/standards/cat>.

<sup>5</sup> См. SRU на <http://www.loc.gov/standards/sru/specs/transport.html>.

<sup>6</sup> См. RSS 2.0 Specification на <http://feed2.w3.org/docs/rss2.html>.

- *Одновременный доступ*—Последние достижения в области распределенной обработки географической информации (Yang and Raskin 2009) и популяризация веб- и беспроводных устройств позволяют одновременно получать доступ к геопространственным системам большому числу конечных пользователей (Goodchild и др. 2007). Имея доступ по всему миру, CLH имеет массивный одновременный доступ с течением времени. Большие одновременные запросы создают большую нагрузку для CLH как на аппаратное, так и на программное обеспечение, что может привести к длительному времени отклика у пользователей. Если время ответа превышает две секунды, пользователи могут чувствовать себя разочарованными (Nah 2004). Как оперативное приложение, к которому обращаются пользователи во всем мире, для CLH особенно важно обеспечить возможность доступа к высокопроизводительным данным

Для решения таких задач Amazon Web Services (AWS) использует различные преимущества облачных сервисов, такие как Amazon EC2 с масштабируемой вычислительной мощностью и Amazon stable Elastic Block Store (EBS) с высоким уровнем ввода-вывода и масштабируемым механизмом. Благодаря этому, CLH может эффективно обрабатывать одновременный доступ в часы пиковых нагрузок. Данные преимущества будут рассмотрены в разделе 8.2.2.

## 8.2 РАЗВЁРТЫВАНИЕ И ОПТИМИЗАЦИЯ

### 8.2.1 Последовательность выполняемых действий при развертывании

На рисунке 8.1 показано, как развернуть CLH на Amazon EC2, где серым цветом отмечены шаги, требующие особого внимания. Пользователям облака необходимо иметь учетную запись для развертывания приложений на EC2. Шаги по созданию учетной записи можно найти в Главе 4, Раздел 4.3.1. После регистрации учетной записи общие шаги будут следующими:

- *Шаг 1.* Авторизация доступа к сети - как описано в Главе 4, Раздел 4.3.1, данный шаг заключается в авторизации доступа к сети путем открытия соответствующих портов. В этом случае открывается порт 22 для обеспечения связи между локальным сервером и экземплярами EC2, а также порт 80 для обеспечения доступа CLH через веб-браузер.
- *Шаг 2.* Запустите экземпляр-CLH - это веб-приложение, которое размещается на сервере Tomcat и запускается на Linux-машине с пространственной базой данных. Поэтому для запуска экземпляра Amazon EC2 выбрано публичное приложение Amazon Machine Image (AMI) с PostgreSQL 8.4 и PostGIS 1.5.
- Шаги 3, 4 и 6. Шаги 3, 4, 5 и 6 на рисунке 8.1 используются для настройки экземпляра. Шаги 3, 4 и 6 являются необязательными, но они делают систему более надежной при большем объеме хранения. Данные шаги рассмотрены в Разделе 8.2.2.1.

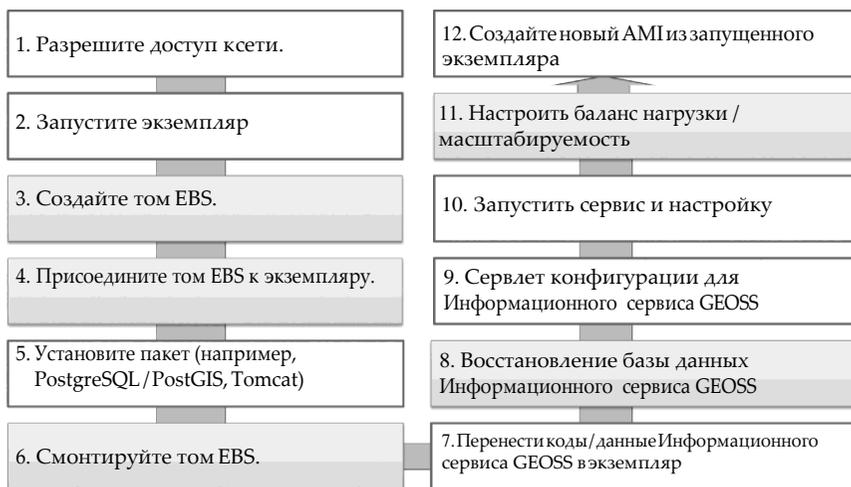


Рисунок 8.1 Процесс развертывания CLN на Amazon EC2 (серые поля обозначают шаги, требующие особого внимания).

Таблица 8.1 Матрица совместимости между различными версиями PostgreSQL и PostGIS \* Версия PostgreSQL PostGIS 1.3 PostGIS 1.4 PostGIS 1.5 PostGIS 2.0

*Trunk*

8.4	Да	Да	Да	Да
9.0	Нет	Нет	Да	Да
9.1	Нет	Нет	Да	Да

\*См. PostgreSQL на <http://trac.osgeo.org/postgis/wiki/UsersWikiPostgreSQLPostGIS>.

- Шаг 5. Пакет установки — На шаге 5 необходимо установить и настроить PostgreSQL / PostGIS и Tomcat. PostgreSQL / PostGIS - это программное обеспечение базы данных, в котором хранятся записи пространственных метаданных CLN, а Tomcat - это контейнер сервлетов, который используется для размещения CLN. Перед установкой PostgreSQL / PostGIS должны быть установлены зависимые пакеты, такие как Proj (библиотека cartographic projection),<sup>1</sup> GEOS (Geometry Engine-Open Source),<sup>2</sup> и libxml (синтаксический анализатор XML C и инструментарий),<sup>3</sup>. Один важный аспект, о котором должны знать потребители, заключается в том, что существуют проблемы несовместимости между различными версиями PostgreSQL и PostGIS (Таблица 8.1), а также PostGIS и GEOS (Таблица 8.2). В этом случае для запуска экземпляра используется Ubuntu AMI с PostgreSQL 8.4 и PostGIS 1.5. Следовательно, потребители экономят время на установку пакетов.

<sup>1</sup> См. Proj. 4 на <http://trac.osgeo.org/proj/>.

<sup>2</sup> См. GEOS на <http://trac.osgeo.org/geos/>.

<sup>3</sup> См. the XML C parser and toolkit of Gnome на <http://www.xmlsoft.org/>.

Таблица 8.2 Матрица совместимости версий GEOS и PostGIS

GEOS версия	PostGIS 1.3	PostGIS 1.4	PostGIS 1.5	PostGIS 2.0 Trunk
Отсутствует	Да (не рекомендовано)	Нет	Нет	Нет
2.2	Да (не рекомендовано)	Нет	Нет	Нет
3.0	Нет	Да (не рекомендовано)	Да	Да
3.1	Нет	Да (не рекомендовано)	Да	Да
3.2	Да	Да	Да	Да (не рекомендовано)
3.3	Да	Да	Да	Да

После настройки базы данных потребители могут настроить Tomcat с помощью следующих команд:

```
root@ip-10-189-149-104:/root$ cd /opt
root@ip-10-189-149-104:/opt$ wget http://www.
poolsabovoground.com/apache/tomcat/tomcat-6/v6.0.26/
bin/apache-tomcat-6.0.26.tar.gz
root@ip-10-189-149-104:/opt$ tar -zxvf apache-
tomcat-6.0.26.tar.gz
```

Чтобы Tomcat работал правильно, также необходимо установить и настроить Java Development Kit (JDK) (Глава 5).

- Шаг 7. Перенос кода и данных CLH. Как передать большой объем данных в облако, подробно описано в главе 5, раздел 5.3.
- Шаг 8. Восстановить базу данных - после переноса приложений CLH в экземпляр и восстановления данных в базе данных для экземпляра, следующие коды можно использовать для восстановления базы данных. Первая строка используется для смены владельца файла дампа базы данных (geoss.Sql) на PostgreSQL; вторая строка используется для переключения на пользователя PostgreSQL; третья строка используется для создания новой базы данных: geoss; четвертая строка используется для восстановления базы данных.

```
root@ip-10-189-149-104:/mnt$ chown postgres:postgres
geoss.sql
root@ip-10-189-149-104:/mnt$ su postgres
bash-3.2$ createdb geonetwork
bash-3.2$ psql geonetwork < geoss.sql
```

- Шаг 9. Сконфигурируйте сервлет для CLH. Для запуска Tomcat на этом этапе создается виртуальный пользователь Tomcat. Следующие коды показывают конфигурацию файла /etc/rc.local, позволяющую CLH запускаться автоматически при загрузке системы.

The screenshot shows the 'SYSTEM CONFIGURATION' page of the GEOSS Clearinghouse. It is divided into several sections: 'SITE', 'SERVER', 'INTRANET', and 'METADATA SEARCH RESULTS'. Each section contains input fields for configuration parameters. The 'SITE' section has 'Name' and 'Organization' both set to 'GEOSS'. The 'SERVER' section has 'Host' set to 'ec2-50-19-223-225.compute-1.amazonaws.com' and 'Port' set to '80'. The 'INTRANET' section has 'Network' set to '127.0.0.1' and 'Netmask' set to '255.0.0.0'. The 'METADATA SEARCH RESULTS' section has 'Maximum selected records' set to '1000'. The page header includes the GEOSS logo and the text 'GROUP ON EARTH OBSERVATIONS' and 'GEOSS Clearinghouse'. A navigation bar at the top contains links for 'Home', 'Administration', 'Contact us', 'Links', 'About', and 'Help'.

Рисунок 8.2 Конфигурации CLH.

```
sudo -u tomcat /opt/tomcat/bin/startup.sh
```

- *Шаг 10.* Запустите службу и настройте приложение. Если база данных и Tomcat настроены правильно, пользователи должны иметь доступ к CLH через веб-браузер после запуска Tomcat. Однако необходимы дополнительные настройки, в том числе установка URL-адреса для удаленного поиска (подробно в Разделе 8.3.2) и изменение пароля для администратора. На рисунке 8.2 показаны интерфейсы настройки URL-адреса для удаленного поиска. Страница конфигурации находится в разделе конфигурации системы администрирования после входа в CLH. Пользователям необходимо изменить URL-адрес сервера на фактическое доменное имя, предоставленное облачными провайдерами.
- *Шаг 11.* Настройте балансировку нагрузки или масштабируемость. Данный шаг необязателен для развертывания CLH в облачном сервисе, но он сделает систему более гибкой и масштабируемой. Балансировка нагрузки и масштабируемость описаны в разделах 8.2.2.2 и 8.2.2.3.
- *Шаг 12.* Создание AMI из запущенного экземпляра - наконец, новый AMI может быть создан на основе запущенного экземпляра CLH (см. Главу 4, раздел 4.3, чтобы узнать, как создать AMI).

## 8.2.2 Специальные соображения

### 8.2.2.1 Резервное копирование данных

Данные, файлы журналов и приложения могут быть скопированы на отдельный том EBS для резервного копирования. Как инструмент командной строки, так и консоль AWS Web Management Console могут использоваться для создания и прикрепления тома EBS к экземпляру CLH.

## Информационный сервис GEOSS с поддержкой облачных технологий 131

Том EBS монтируется в каталог PostgreSQL. Такой отдельный том имеет два преимущества: (1) он может быть использован для восстановления CLH из тома в случае сбоя текущего экземпляра, и (2) том может быть любого размера от 1 ГБ до 1 ТБ.

Следующие шаги показывают, как создавать и использовать объем EBS.

- *Шаг 1.* Создайте том EBS с нуля без содержимого в веб консоли и убедитесь, что выбранная зона тома EBS совпадает с зоной экземпляра CLH (рисунок 8.3).
- *Шаг 2.* Прикрепите том к активному экземпляру. Следуйте инструкциям по использованию `/dev/sdf` для первой точки монтирования (рисунок 8.3). Если необходимо прикрепить несколько томов, следуйте алфавитной последовательности для каждого тома, например, `/dev/sdg`, `/dev/sdh` и т.д.
- *Шаг 3.* Войдите в экземпляр и переместите раздел данных PostgreSQL на том EBS. После входа в экземпляр, необходимо:
- (1) создать файловую систему на новом томе с помощью команды `mkfs`; (2) убедиться, что службы Tomcat и PostgreSQL выключены должным образом; (3) создать каталог на загрузочном томе EBS для резервного копирования (например, `/mnt/datavol_1`); (4) смонтировать том `/dev/sdh` в новой точке монтирования; и, наконец, (5) создать резервную копию данных, журнальных файлов и приложения в новом хранилище.

```
[root@ ip-10-189-149-104~] mkfs -t ext3 /dev/sdh#  
make a file system  
[root@ip-10-189-149-104~] mkdir /mnt/datavol_1  
[root@ip-10-189-149-104~] mount /dev/sdh  
/mnt/datavol_1/
```

Вышеприведенная команда показывает команды создания и прикрепления тома EBS (предполагая, что доменное имя экземпляра `ip-10-189-149-104`, а учетная запись пользователя - "root").

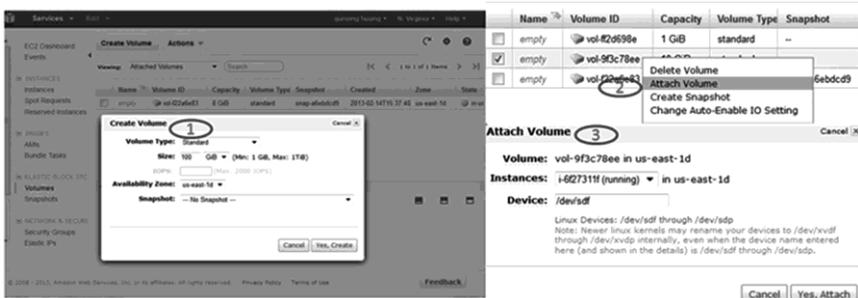


Рисунок 8.3 Создание нового тома EBS и прикрепление к экземпляру.



Рисунок 8.4 Настройка сервиса балансировки нагрузки.

### 8.2.2.2 Балансировки нагрузки

После успешного запуска экземпляра Amazon EC2 потребители могут настраивать расширенные сервисы для экземпляра, такие как балансировка нагрузки и возможность калибровки. AWS предоставляет балансировщик нагрузки<sup>1</sup> для балансировки нагрузки запросов пользователей на нескольких экземплярах EC2. Поскольку CLN может рассчитывать на одновременную обработку запросов от огромного числа пользователей, служба балансировки нагрузки будет полезна для балансировки нагрузки трафика системы. В разделе «Сеть и безопасность» пользователи могут щелкнуть «Балансировщик нагрузки», чтобы начать настройку службы балансировки нагрузки (рис. 8.4).

В этом случае при использовании служб балансировки нагрузки используется только одна копия базы данных, что означает, что все экземпляры будут обращаться к одной и той же базе данных, настроенной на первом экземпляре.

### 8.2.2.3 Автоматическое масштабирование

Геопространственные приложения могут испытывать ежечасную, суточную или еженедельную изменчивость доступа пользователей, что требует различных вычислительных возможностей в разное время для минимизации затрат и удовлетворения требований к производительности. Масштабируемые сервисы EC2 могут упростить геопространственные приложения, эластично удовлетворяя вычислительные требования. AWS предоставляет автоматическое масштабирование<sup>2</sup> для реализации такой функции.

<sup>1</sup> См. Amazon Web Services на <http://aws.amazon.com/elasticloadbalancing/>.

<sup>2</sup> См. Amazon Web Services на <http://aws.amazon.com/autoscaling/>.

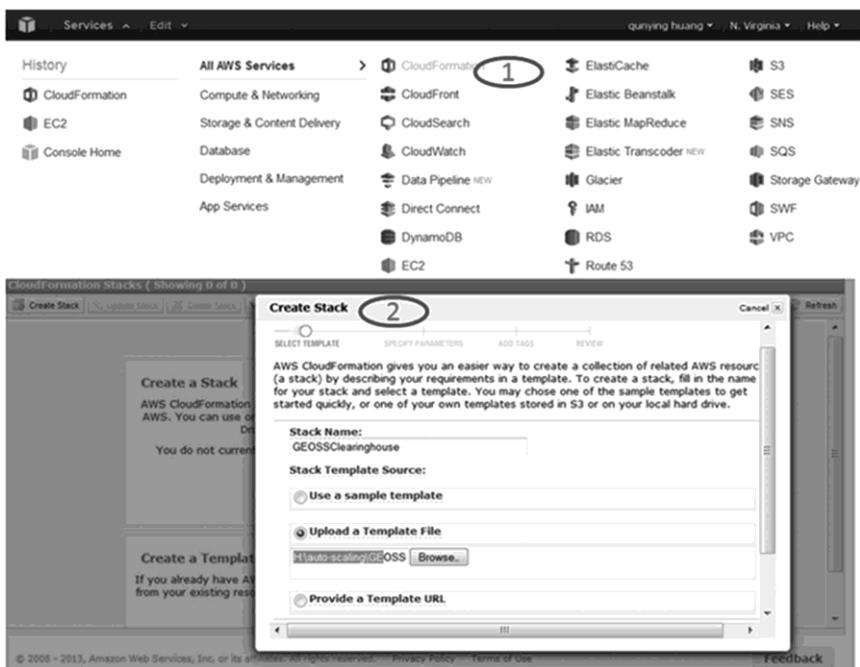


Рисунок 8.5 Использование сервиса CloudFormation для настройки возможности автоматического масштабирования через веб-консоль.

Пользователи облака также могут использовать сервис CloudFormation через веб-консоль (рисунок 8.5).

Пользователи также могут настроить службу автоматического масштабирования через командную строку.

```
cfn-create-stack GEOSSClearinghouse --template-file  
GEOSSClearinghouse.template --region us-east-1  
--awsaccesskey=FAKEKEY --awssecretkey=FAKEKEY2  
--parameters="KeyName=GeoNet; InstanceType=m1.large"
```

Однако независимо от того, какие сервисы или инструменты используются для настройки функции автоматического масштабирования, потребителям необходимо настроить шаблон (т.е. источник шаблона стека на рис. 8.4 и входные данные для файла шаблона параметров при использовании вышеупомянутой команды `cfn-create-stack`). В Приложении 8.1 приведен пример шаблона. Шаблон представляет собой файл JSON<sup>1</sup>, который определяет требования к инфраструктуре. В шаблоне наиболее важным объектом, который необходимо определить, является триггер ресурса. В образце шаблона триггер автоматического масштабирования основан на загрузке ЦП веб-серверов, как показано ниже.

<sup>1</sup>См. Introducing JSON на <http://www.json.org>.

```
"AutoScalingTrigger" : {
  "Type" : "AWS::AutoScaling::Trigger",
  "Properties" : {
    "MetricName" : "CPUUtilization",
    "Namespace" : "AWS/EC2",
    "Statistic" : "Average",
    "Period" : "300",
    "UpperBreachScaleIncrement" : "1",
    "LowerBreachScaleIncrement" : "-1",
    "AutoScalingGroupName" : { "Ref" : "WebServerGroup" },
    "BreachDuration" : "600",
    "UpperThreshold" : "90",
    "LowerThreshold" : "75",
    "Dimensions" : [ {
      "Name" : "AutoScalingGroupName",
      "Value" : { "Ref" : "WebServerGroup" }
    } ]
  }
},
```

Данные настройки добавят новый экземпляр, когда средняя загрузка ЦП за 300-секундный период превысит 90% (до параметра Autoscaling MaxSize, настроенного в шаблоне), и удалит экземпляр после средней загрузки ЦП за 300-секундный период меньше 75% (до параметра AutoScaling MinSize). Помимо CPUUtilization, потребители также могут использовать другие значения параметра MetricName, такие, как Latency и RequestCount. Задержка запускает масштабирование экземпляров на основе времени задержки ответа веб-сервиса, а RequestCount основан на количестве номеров одновременных запросов пользователей. Более подробную информацию о шаблоне можно найти на веб-сайте AWS.<sup>1</sup>

Моментальный снимок тома EBS создается автоматически каждые 5 минут для хранения кодов PostgreSQL / PostGIS, CLN и временных данных, чтобы новый масштабированный экземпляр мог присоединить последний том данных с синхронизированными данными. Система будет хранить только одну копию последнего тома данных (после создания нового тома EBS он заменит предыдущую версию) для экономии затрат, поскольку Amazon взимает плату за хранение на основе общего размера используемого тома.

### 8.2.3 Отличия от основных шагов в главе 5

CLN объединяет несколько сложных компонентов, включая управление базами данных (PostgreSQL / PostGIS), управление ресурсами метаданных и их сбор, а также предварительный просмотр геопространственных данных. Потенциальная интенсивность вычислений из-за управления большими данными и одновременных запросов дополнительно усложняет всю систему.

<sup>1</sup> См. Amazon Web Services на <http://aws.amazon.com/cloudformation/aws-cloudformation-templates/>.

## Информационный сервис GEOSS с поддержкой облачных технологий 135

Во время развертывания CLN на Amazon EC2 мы использовали несколько специальных стратегий, чтобы сделать его полностью функциональным, включая резервное копирование базы данных и настройку балансировки нагрузки на сервер и автоматическое масштабирование.

### 8.3 ДЕМОНСТРАЦИЯ СИСТЕМЫ

Доступ к CLN с поддержкой облака можно получить в Информационном центре GEOSS по адресу <http://ec2-50-19-223-225.compute-1.amazonaws.com/geonetwork>. В настоящем разделе показано, как обнаруживать наборы геопространственных данных и ресурсы, используя как локальный поиск (через интерфейс CLN), так и удаленный поиск (через стандартные интерфейсы веб-служб CLN вместо веб-интерфейса).

#### 8.3.1 Локальный поиск

Функция локального поиска позволяет пользователям находить записи на основе заголовка, ключевого слова, пространственного положения и временной информации. Результаты можно отсортировать по разным правилам (например, по релевантности, дате, заголовку, алфавитному порядку, популярности и рейтингу). Кнопки для инструментов доступности динамически добавляются к каждой панели записи в соответствии с типами ресурсов (например, «Интерактивная карта» будет добавлена в WMS). Пользователи также могут использовать интегрированные инструменты визуализации (на базе OpenLayers<sup>1</sup>) для взаимодействия со службами данных. На рисунке 8.6 показаны результаты поиска по глобальным показателям Суммы годовых осадков NDVI» за период с 1980 по 2013 год.

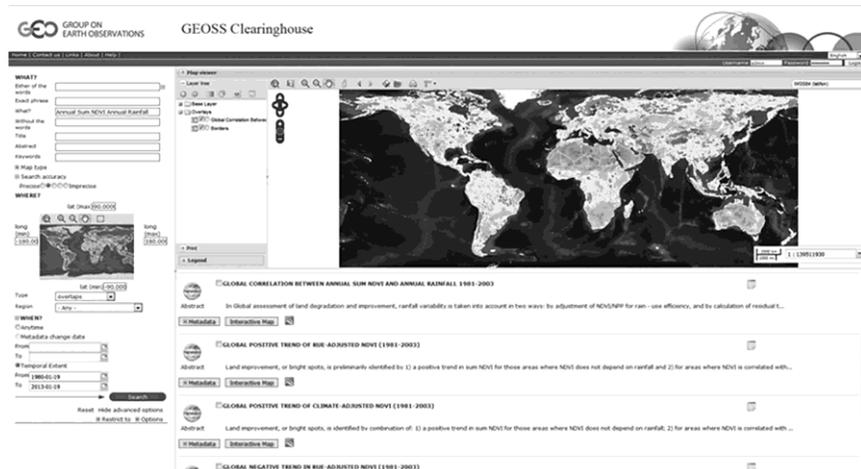


Рисунок 8.6 (См. цветную вставку.) Результаты поиска по запросу «Annual Sum NDVI Annual Rainfall».

<sup>1</sup>См. OpenLayers на <http://openlayers.org/>.

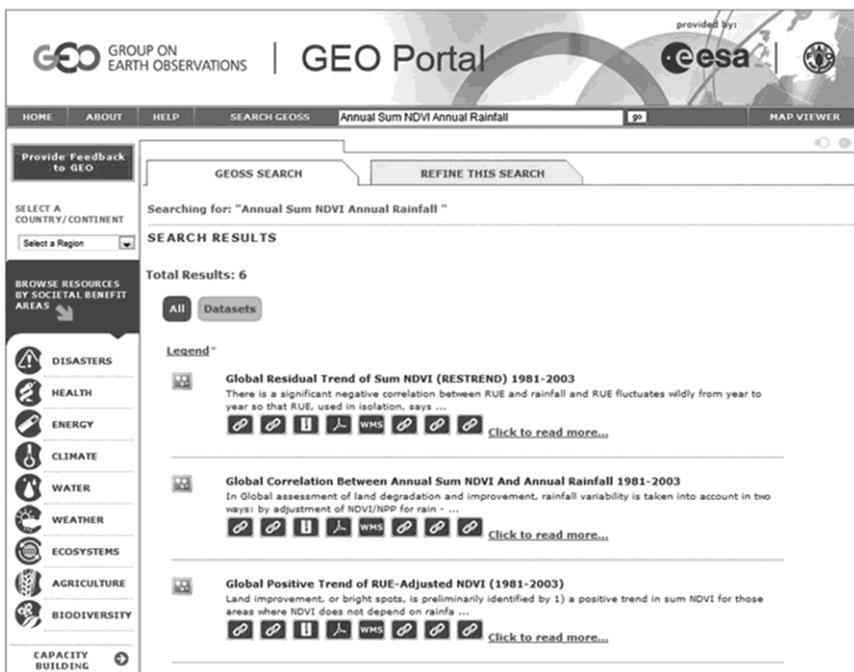


Рисунок 8.7 (См. цветную вставку) Результаты поиска по запросу «Annual Sum NDVI Annual Rainfall» на портале GEO

### 8.3.2 Удаленный поиск

CLH поддерживает различные протоколы для удаленного поиска данных ЕО. Это позволяет глобальным пользователям настраивать свои приложения для использования записей ЕО от GEOSS. Используя возможности удаленного поиска, пользователи могли отправлять поисковые запросы от своих клиентов в CLH. Протоколы поиска, поддерживаемые CLH, включают CSW, SRU и RSS. На рисунке 8.7 показаны результаты поиска CSW относительно CLH из веб-интерфейса GEO Portal.

## 8.4 ЗАКЛЮЧЕНИЕ

Облачная обработка данных предлагает как экономические, так и технические преимущества для поддержки CLH.

## Информационный сервис GEOSS с поддержкой облачных технологий 137

### 8.4.1 Экономические преимущества

Одно из самых больших преимуществ состоит в том, что традиционный сервер больше не требуется для размещения CLH. Ежемесячные расходы на электричество, систему охлаждения, физическое оборудование и системного администратора для обслуживания сервера сохраняются. В таблице 8.3 показано сравнение затрат на использование сервисов AWS и локального сервера (Huang et al. 2011). Сравниваются фиксированная (разовая) стоимость и ежемесячная стоимость в каждом году за 10-летний период.

Таблица 8.3 Сравнение стоимости сервисов AWS и локального сервера \*

Стоимость	Модель аренды	Amazon EC2	Локальный сервер
Фиксированная (единовременная) стоимость		Отсутствует	Около \$2,000 на покупку сервера
Ежемесячная стоимость сети		1 доллар за передачу данных в месяц	Нет (входит в плату за обслуживание)
Хранение		\$22/месяц	Нет (включено в первую покупку)
Ц			
Вычислительные мощности	Не выбрано	\$255/месяц	~ \$17 (при условии, что сервер можно использовать в течение 10 лет, и тогда ежемесячная стоимость будет $2000 / (10 * 12)$ )
	Выбрано	\$57/месяц (\$676 при резервировании большого экземпляра при интенсивной загрузке на три года); \$13/месяц (1028 долларов при бронировании на 3 года)	
Стоимость обслуживания (охлаждение, система, системный администратор, обслуживание, помещение и т.д.)		Отсутствует	\$200/в месяц (Предполагается, что 100 долларов за охлаждение, сеть и плату за комнату и 100 долларов за оплату системному администратору за проверку и обслуживание сервера)
В год	Не выбран	\$3,300	\$2,604
итого	Выбран	\$432 (экземпляр арендован на 3 года)	

Примечание: Стоимость облачной обработки данных основана на стандартном большом экземпляре Amazon EC2 с двумя ядрами ЦП для размещения CLH. Если выбраны более продвинутое типы экземпляров и настроены передовые облачные сервисы (например, масштабируемость), стоимость вычислений увеличится.

\*По сравнению с 27 марта 2013 г.

Ежегодная стоимость сервисов AWS составляет 432 доллара США при резервировании стандартного большого экземпляра с 2 ядрами ЦП на трехлетний срок (всего 1028 долларов США за экземпляр на три года по состоянию на 27 марта 2013 г.); годовая стоимость локального сервера составляет 2604 доллара. Это экономит около 83,6% в год при развертывании и размещении CLN в облачных сервисах.

## 8.4.2 Технические преимущества

Облачный сервис (Amazon EC2) обеспечивает масштабируемость для поддержания формата CLN при массивном одновременном доступе. Amazon EC2 также предлагает высоконадежную среду для CLN, поскольку EC2 работает в проверенной сетевой инфраструктуре и центрах обработки данных Amazon. Внедрение CLN на Amazon EC2 свидетельствует о том, что облачные вычисления обеспечивают экономически эффективный подход к размещению и эксплуатации геопространственных приложений в высокомасштабируемой и надежной вычислительной среде.

## 8.5 КРАТКОЕ ИЗЛОЖЕНИЕ

В данной главе рассказывается, как использовать облачную обработку данных для поддержки CLN. Раздел 8.1 знакомит с предысторией и проблемами CLN. Раздел 8.2 демонстрирует развертывание CLN в облаке. Раздел 8.3 иллюстрирует пользовательские сценарии CLN, а в разделе 8.4 завершается глава кратким описанием экономических и технических преимуществ развертывания CLN в облачных сервисах.

## 8.6 ПРОБЛЕМЫ

1. Каковы общие шаги по развертыванию центра обмена информацией GEOSS в облаке? Каковы отличия от общих шагов, описанных в Главе 5?
2. Как переносить данные на подключаемый том Amazon EBS?
3. Какие услуги компьютерного облака можно использовать для балансировки нагрузки системы? Обсудите, как их использовать.
4. Какие масштабируемые сервисы предоставляет AWS? Какими пользоваться?
5. На примере Информационного сервиса GEOSS расскажите о технических преимуществах геонаучных приложений, основанных на облачных технологиях.

## ПРИЛОЖЕНИЕ 8.1 ШАБЛОН СОЗДАНИЯ ФУНКЦИИ АВТОМАТИЧЕСКОГО МАСШТАБИРОВАНИЯ

```
{  
  "AWSTemplateFormatVersion" : "2010-09-09",  
  "Description" : "Create a load balanced, auto scaled CLN  
    Web site. ",
```

## Информационный сервис GEOSS с поддержкой облачных технологий 139

---

```
"Parameters" : {
  "InstanceType" : {
    "Description" : "Type of EC2 instance to launch",
    "Type" : "String",
    "Default" : "m1.large"
  },
  "WebServerPort" : {
    "Description" : "The TCP port for the Web Server",
    "Type" : "String",
    "Default" : "80"
  },
  "KeyName" : {
    "Description" : "The EC2 Key Pair to allow SSH access
      to the instances",
    "Type" : "String"
  }
},

"Mappings" : {
  "AWSInstanceType2Arch" : {
    "m1.large" : {"Arch" : "64"},
    "m1.xlarge" : {"Arch" : "64"}
  },
  "AWSRegionArch2AMI" : {
    "us-east-1" : {"64" : "xxxxxx"},
    "us-west-1" : {"64" : " xxxxxx "},
    "eu-west-1" : { "64" : " xxxxxx "},
    "ap-southeast-1" : {"64" : " xxxxxx "},
    "ap-northeast-1" : {"64" : " xxxxxx "}
  }
},

"Resources" : {
  "WebServerGroup" : {
    "Type" : "AWS::AutoScaling::AutoScalingGroup",
    "Properties" : {
      "AvailabilityZones" : {"Fn::GetAZs" : ""},
      "LaunchConfigurationName" : {"Ref" : "LaunchConfig"},
      "MinSize" : "1",
      "MaxSize" : "3",
      "LoadBalancerNames" : [{"Ref" : "ElasticLoadBalancer"}]
    }
  },

  "LaunchConfig" : {
    "Type" : "AWS::AutoScaling::LaunchConfiguration",
    "Properties" : {
      "KeyName" : {"Ref" : "KeyName"},

```

```
"ImageId" : {"Fn::FindInMap" : ["AWSRegionArch2AMI",
                                {"Ref" : "AWS::Region"},
                                {"Fn::FindInMap" :
["AWSInstanceType2Arch", {"Ref" : "InstanceType"},
                                "Arch"]}],
"UserData" : {"Fn::Base64" : {"Ref" :
"WebServerPort"}},
"SecurityGroups" : [{"Ref" : "InstanceSecurityGroup"}],
"InstanceType" : {"Ref" : "InstanceType"}
}
},
"CPUBasedTrigger" : {
  "Type" : "AWS::AutoScaling::Trigger",
  "Properties" : {
    "MetricName" : "CPUUtilization",
    "Namespace" : "AWS/EC2",
    "Statistic" : "Average",
    "Period" : "300",
    "UpperBreachScaleIncrement" : "1",
    "LowerBreachScaleIncrement" : "-1",
    "AutoScalingGroupName" : {"Ref" : "WebServerGroup"},
    "BreachDuration" : "600",
    "UpperThreshold" : "90",
    "LowerThreshold" : "75",
    "Dimensions" : [{
      "Name" : "AutoScalingGroupName",
      "Value" : {"Ref" : "WebServerGroup"}
    }]
  }
},
"ElasticLoadBalancer" : {
  "Type" : "AWS::ElasticLoadBalancing::LoadBalancer",
  "Properties" : {
    "AvailabilityZones" : {"Fn::GetAZs" : ""},
    "Listeners" : [{
      "LoadBalancerPort" : "80",
      "InstancePort" : {"Ref" : "WebServerPort"},
      "Protocol" : "HTTP"
    }],
    "HealthCheck" : {
      "Target" : {"Fn::Join" : [ "", [ "HTTP:", {"Ref" :
"WebServerPort"}, "/" ] ]},
      "HealthyThreshold" : "3",
      "UnhealthyThreshold" : "5",
      "Interval" : "30",
      "Timeout" : "5"
    }
  }
}
```

## Информационный сервис GEOSS с поддержкой облачных технологий 141

---

```
    }  
  },  
  "InstanceSecurityGroup" : {  
    "Type" : "AWS::EC2::SecurityGroup",  
    "Properties" : {  
      "GroupDescription" : "Enable SSH access and HTTP access  
                           on the inbound port",  
      "SecurityGroupIngress" : [{  
        "IpProtocol" : "tcp",  
        "FromPort" : "22",  
        "ToPort" : "22",  
        "CidrIp" : "0.0.0.0/0"  
      }],  
      {  
        "IpProtocol" : "tcp",  
        "FromPort" : {"Ref" : "WebServerPort"},  
        "ToPort" : {"Ref" : "WebServerPort"},  
        "CidrIp" : "0.0.0.0/0"  
      }  
    ]  
  }  
},  
"Outputs" : {  
  "URL" : {  
    "Description" : "The URL of the website",  
    "Value" : {"Fn::Join" : [ "", [ "http://", {"Fn::GetAtt" :  
      [ "ElasticLoadBalancer", "DNSName" ] ] ] ] }  
  }  
}
```

## СПИСОК ЛИТЕРАТУРЫ

- Christian, E. 2005. Planning for the Global Earth Observation System of Systems (GEOSS). *Space Policy* 21, no. 2: 105–109.
- GEO. 2009–2011 Work Plan [online], 2009. [http://www.earthobservations.org/documents/work%20plan/geo\\_wp0911\\_rev2\\_091210.pdf](http://www.earthobservations.org/documents/work%20plan/geo_wp0911_rev2_091210.pdf) (accessed January 4, 2013).
- Goodchild, M. F., M. Yuan, and T. J. Cova. 2007. Towards a general theory of geographic representation in GIS. *International Journal of Geographical Information Science* 21, no. 3: 239–260.
- Huang, Q., D. Nebert, C. Yang, and K. Liu. 2011. GeoCloud Project Report—CLH [online]. <http://www.fgdc.gov/initiatives/geoplatform/geocloud/reports/fgdc-geocloud-project-report-geonetwork.pdf> (accessed March 4, 2013).
- Liu, K., C. Yang, W. Li, Z. Li, H. Wu, A. Rezgui, and J. Xia. 2011. The CLH High Performance Search Engine. *The 19th International Conference on Geoinformatics*. June 24–26, Shanghai, China.

- Nah, F. F. H., X. Tan, and S. H. Teh. 2004. An empirical investigation on end-users' acceptance of enterprise systems. *Information Resources Management Journal (IRMJ)* 17, 3: 32–53.
- Yang, P., J. Evans, M. Cole, N. Alameh, S. Marley, and M. Bambacus. 2007. The emerging concepts and applications of the spatial Web portal. *Photogrammetric Engineering and Remote Sensing* 73, no. 6: 691.
- Yang, C. and R. Raskin. 2009. Introduction to distributed geographic information processing research. *International Journal of Geographical Information Science* 23, no. 5: 553–560.

## Глава 9

---

# Climate@Home с облачной поддержкой

*Цзин Ли, Чжэньлун Ли, Мин Сунь, Цюньбин Хуанг, Кай Лю и Майра Бамбакус*

---

Проект Climate@Home - это инициатива, направленная на создание виртуального суперЭВМ для поддержки моделирования климата. Облачные сервисы используются для поддержки управления массовыми выходными данными моделей и вычислительными ресурсами, а также для облегчения эффективного анализа выходных данных. В этой главе рассказывается о развертывании системы Climate@Home в облачных сервисах.

### 9.1 Climate@home: ИСХОДНЫЕ ДАННЫЕ И СЛОЖНОСТИ

#### 9.1.1 Исходные данные

Моделирование климата реконструирует прошлое и предсказывает климатические условия в будущем посредством количественной оценки взаимодействия между атмосферой, сушей, океаном и морским льдом с числовыми описаниями (Skamarock and Klemp 2008). Моделирование климата характеризуется как интенсивностью вычислений, так и интенсивностью данных. С одной стороны, климатические модели могут генерировать массивные наборы данных для описания климатологической и геофизической динамики. С другой стороны, чтобы избежать систематических ошибок, вызванных конфигурациями модели, необходимо несколько прогонов (Schmidt et al. 2006). Для завершения этих прогонов требуются значительные объемы вычислительных ресурсов. В качестве пилотного исследования проект Climate@Home ищет альтернативный подход для удовлетворения вычислительных требований моделирования климата и внесения вклада в исследование глобального климата. Основная цель этого проекта - создать виртуальный суперкомпьютер за счет использования вычислительных ресурсов граждан, которые могут поддерживать моделирование нескольких моделей. Модель E<sup>1</sup> Института космических исследований имени Годдарда NASA выбрана в качестве модели климата для демонстрационных целей. Учитывая основную цель проекта, основная задача состоит в том, чтобы разработать вычислительную систему и систему управления данными, которая служит комплексным решением для интеграции ресурсов, предоставленных гражданами, и содействия научному анализу.

---

<sup>1</sup> См. NASA на <http://www.giss.nasa.gov/tools/modelE/>.

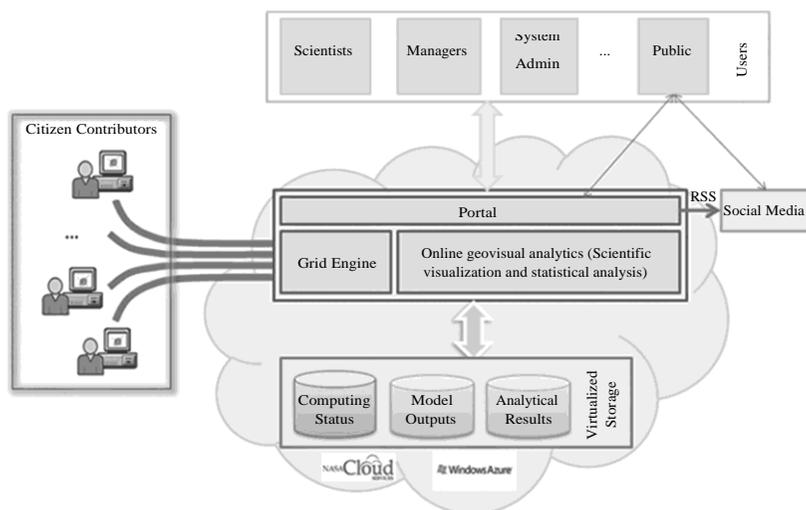


Рисунок 9.1 Архитектура системы Climate@Home.

Данная система должна иметь следующие функции: (1) совместная вычислительная среда, в которой моделирование моделей может быть отправлено на вычислительные устройства; (2) Хранилище данных, где хранятся наборы данных из нескольких источников, включая выходные данные модели, аналитические результаты и состояние ресурсов; (3) интерактивный визуальный аналитический компонент для облегчения анализа ранних научных исследований и состояния системы; и (4) набор защищенных интерфейсов онлайн-доступа к трем вышеупомянутым функциям. На рисунке 9.1 показана архитектура Climate @ Home. Двумя основными компонентами являются: (1) пространственный веб-портал (Ян и др., 2007), объединяющий несколько функций, и (2) сервер BOINC<sup>1</sup> - служит механизмом сетки, который управляет запуском моделей и наборами данных.

- *Пространственный веб-портал*—Учитывая сложность архитектуры системы и множество функций, предлагаемых пользователям разных уровней (например, ученым и добровольцам), пространственный веб-портал разработан для интеграции всех компонентов внешнего интерфейса с функциями внутреннего интерфейса. Общедоступные и частные интерфейсы портала позволяют ученым, техническому персоналу и разработчикам ресурсов безопасно получать доступ к этим компонентам. Для создания пространственного веб-портала Drupal<sup>2</sup> используется в качестве контейнера для интеграции этих компонентов и функций. Drupal - это система управления контентом с открытым исходным кодом (CMS), которая предоставляет пользователям гибкие интерфейсы для настройки сайта и принятия пакетов, предоставленных разработчиками сообщества.

<sup>1</sup>См. BOINC на <http://boinc.berkeley.edu/>.

<sup>2</sup>См. Drupal на <http://drupal.org/>.

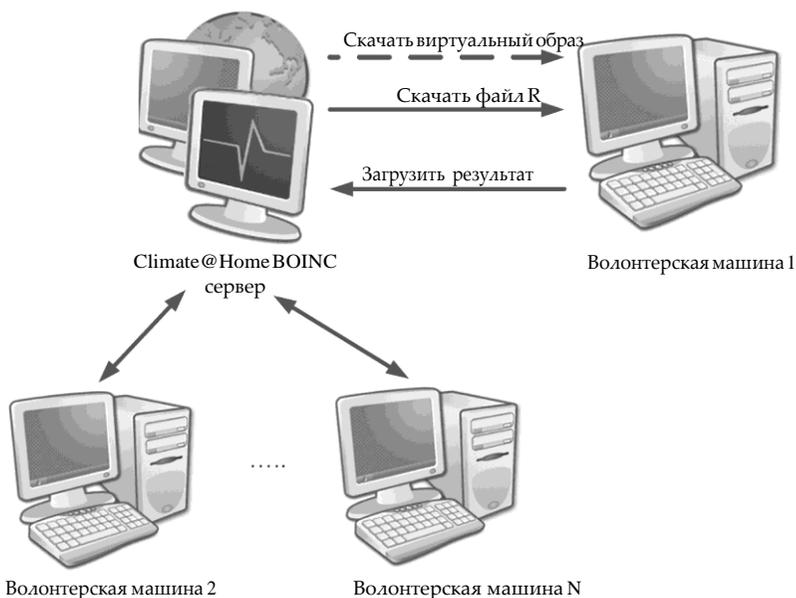


Рисунок 9.2 Рабочий процесс моделирования Climate@Home на основе BOINC.

Геовизуальный аналитический портлет и портлет управления ресурсами на основе онлайн-картографических веб-приложений (Bing Maps<sup>1</sup> и Google Earth<sup>2</sup>) интегрированы с порталом. Вышеуказанные два портлета поддерживают геовизуальный анализ климатических данных и визуальное управление предоставленными вычислительными ресурсами (Sun et al. 2012).

- Сервер BOINC используется в качестве механизма сетки, который отвечает за диспетчеризацию задач моделирования ModelE, сбор выходных данных модели и координацию с клиентами BOINC (рисунок 9.2). Каждое вычислительное устройство от граждан-участников рассматривается как клиент BOINC. BOINC может работать с различными платформами, такими как Windows, Linux/x86 и MacOS/X (Anderson 2004). Поскольку ModelE - это климатическая модель на основе Linux, VirtualBox от Oracle (Anderson 2010) выбран в качестве программного обеспечения виртуализации для создания виртуальных машин Linux на клиентских устройствах. Виртуальный образ загружается на клиента от граждан, впервые вносящих вклад. Сервер BOINC постоянно отправляет исправления модели клиенту. После завершения прогона модели на стороне клиента выходные данные модели преобразуются в файлы данных NetCDF<sup>3</sup> и загружаются на сервер BOINC.

<sup>1</sup> См. Bing Maps на <http://www.bing.com/maps/>.

<sup>2</sup> См. Google Earth на <http://www.google.com/earth/index.html>.

<sup>3</sup> См. Unidata на <http://www.unidata.ucar.edu/software/netcdf/>.

## 9.1.2 Сложности

Для разработки системы необходимо решить следующие задачи.

- *Создание масштабируемого и гибкого хранилища данных для облегчения управления массивными наборами данных.* Хранилище данных должно предлагать две основные функции. Во-первых, должен иметь возможность архивировать наборы климатических данных большого объема. Используя ModelE в качестве примера, объем данных ежемесячных результатов моделирования за десять лет из 300 прогонов модели составляет около 2,5 терабайт (Sun et al. 2012). Во-вторых, конфигурации хранилища данных должны динамически корректироваться в соответствии с шаблонами операций с данными, такими как загрузка, удаление и выгрузка данных. Размер склада должен увеличиться, когда большинство запусков будет ближе к завершению, чтобы заархивировать результаты модели дохода, загруженные свычислительных устройств граждан. Следовательно, следует создать масштабируемое и эластичное хранилище данных для управления большими климатическими данными.
- *Обработка распределенной передачи данных и одновременный доступ для поддержки онлайн-геовизуальной аналитики - система Climate@Home предоставляет онлайн-геовизуальный аналитический модуль.* Хотя несколько инструментов геовизуального анализа были разработаны для ученых для проведения статистического анализа больших объемов климатических данных (например, Santos et al. 2013), эти инструменты не обеспечивают достаточных функций для интерактивного анализа пространственно-временных климатических данных большого объема в наборах данных в сетевой среде. Частично это связано с тем, что эти инструменты не были разработаны для выполнения визуализации и анализа в веб-среде. Возможности отвечать на параллельные запросы и выполнять задачи визуализации для массивных наборов данных становятся основными узкими местами при миграции традиционных геовизуальных аналитических функций в сетевую среду. Следовательно, разработка механизма, который может обрабатывать распределенную передачу данных и одновременный доступ, имеет решающее значение для выполнения аналитического конвейера.
- *Обеспечение надежного и централизованного управления киберинфраструктурой для ресурсов.* Централизованное управление ресурсами обеспечивает динамическую доставку этих ресурсов в рамках общей сети. Эффективное централизованное управление повышает общую эффективность системы. Данный проект включает несколько типов ресурсов, таких как данные, прогоны моделей и вычислительные устройства. Неоднородность этих ресурсов увеличивает трудности выполнения консолидированного централизованного плана управления. Это также вызывает проблемы с надежностью и устойчивостью киберинфраструктуры, которая поддерживает централизованное управление.

Для решения описанных выше проблем пространственный веб-портал и сервер BOINC развернуты в облачных сервисах. Развертывание улучшает производительность системы и повышает надежность системы Climate@Home.

## 9.2 РАЗВЕРТЫВАНИЕ И ОПТИМИЗАЦИЯ

В данном разделе описывается общий рабочий процесс развертывания двух компонентов в Amazon EC2. Выделены особые конфигурации процедуры развертывания.

### 9.2.1 Рабочий процесс общего развертывания

#### 9.2.1.1 Развертывание пространственного веб-портала

Amazon EC2 используется в качестве облачной службы для развертывания (рисунок 9.3). Перед развертыванием требуется учетная запись AWS. Шаги по созданию учетной записи описаны в главе 4. После создания учетной записи потребители сначала входят в веб-консоль Amazon, а затем запускают развертывание. Следующие шаги демонстрируют общую процедуру развертывания пространственного веб-портала. Подробное описание шагов и команд см. В онлайн-видео Главы 9.

- *Шаг 1.* Авторизация доступа к сети - как описано в главе 4, раздел 4.3.1, на этом шаге разрешается требуемый доступ к сети путем открытия соответствующих портов. Должны быть открыты три порта: порт 22, порт 80 и порт 8080. Порт 22 используется для обеспечения доступа к экземпляру через Secure Shell (SSH). Порт 80 требуется для веб-сервера Apache2, порт 8080 используется сервером Tomcat.

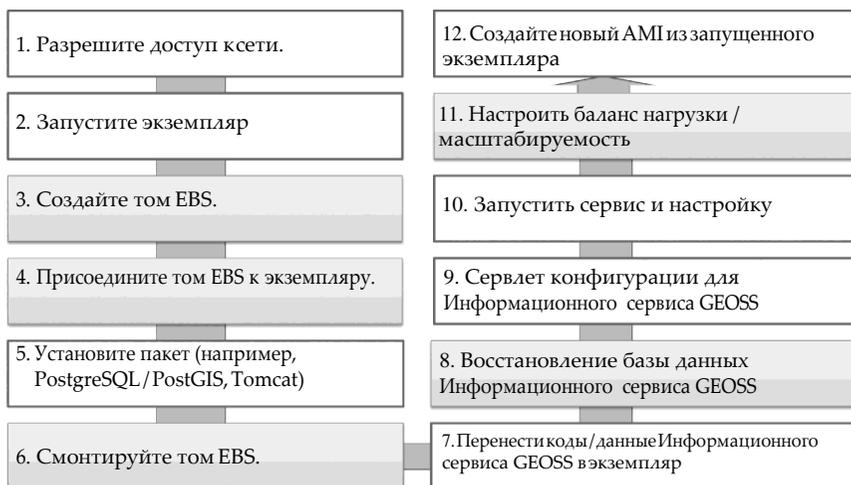


Рисунок 9.3 Процесс развертывания пространственного веб-портала Climate@Home на Amazon EC2. (Серые поля обозначают необязательные шаги, которые описаны в главе 8.)

- *Шаг 2.* Запуск экземпляра - общедоступный образ машины Amazon (AMI) с Ubuntu 12.04, поскольку ОС выбирается в качестве базового образа для создания экземпляра для дальнейшей настройки. Запуск инстанса на основе публичного AMI.
- Шаги 3, 4 и 6. Настройте эластичное блочное хранилище (EBS) - шаги 3, 4 и 6 необязательны. Подробные конфигурации обсуждались в главе 8, раздел 8.2.2.1.
- Шаг 5. Установите пакеты программного обеспечения. Несколько пакетов программного обеспечения устанавливаются и настраиваются в соответствии с требованиями пространственного веб-портала, включая веб-сервер Apache2, MySQL и сервер Tomcat. Установка веб-сервера Apache2 и MySQL описана в главе 5, раздел 5.3.1. Установка сервера Tomcat описана в главе 8, раздел 8.2..
- *Шаг 7.* Перенос исходных кодов и базы данных системы Climate@Home - исходные коды Drupal перемещаются в веб-папку веб-сервера Apache2. Геовизуальные аналитические портлеты перемещаются в веб-папку на сервере Tomcat. Файлы базы данных следует переместить в защищенную папку в экземпляре для последующего восстановления. База данных выгружается с существующего веб-сайта. Перенос данных и файлов с локального рабочего стола на облачный сервер можно найти в главе 4, раздел 4.3.

```
root@ip-10-189-149-104:/mnt$ mysqldump -u username  
-p climatehome > climatehome .sql # dump the database  
"climatehome" using the root account
```

- *Шаг 8.* Восстановить базу данных - для восстановления базы данных необходимы файлы базы данных. На этом шаге восстанавливаются базы данных как для главного портала, так и для гео-визуального аналитического портлета. Перед импортом файлов базы данных необходимо создать пользователя базы данных. В следующих командных строках показано, как восстановить базу данных MySQL из файла резервной копии базы данных (climhome.sql).

```
root@ip-10-189-149-104:/mnt$ mysql -u username -p #log  
in the mysql database using root  
  
mysql>create user 'climateuser'@localhost identified by  
'password' #create a user account (e.g. climateuser)  
for the portal  
  
mysql>create database climatehome #the name of the  
climate@home drupal database  
  
mysql> grant all privileges on climatehome .* to  
'climateuser'@localhost identified by 'password'  
#grant permission to the database
```

```
mysql>flush privileges
```

```
mysql>exit #quit mysql
```

```
root@ip-10-189-149-104:/mnt$ mysql -u username -p cli-  
matehome< climatehome.sql #import the climatehome  
database climatehome.sql on the cloud server
```

- *Шаг 9.* Запустите сервер Apache2 и Tomcat и настройте веб-портал. Следующие команды можно использовать для запуска серверов в экземпляре Ubuntu. Первая строка запускает веб-сервер Apache2. Чтобы запустить сервер Tomcat, можно перейти в каталог bin Tomcat и запустить вторую командную строку.

```
root@ip-10-189-149-104:/$ sudo service apache2 start  
root@ip-10-189-149-104:/Tomcat/bin/$ ./startup.sh
```

После запуска двух серверов к веб-порталу можно будет получить доступ по адресу [http://Your\\_VM\\_Public\\_DNS\(IP\)/climate@home/](http://Your_VM_Public_DNS(IP)/climate@home/). Портлет визуализации необходимо настроить на веб-портале для правильного отображения. Подробные инструкции по настройке и информацию для входа на веб-портал можно найти в онлайн-материалах главы 9

- *Шаг 10.* Настройте балансировку нагрузки или масштабируемость - это необязательный шаг, чтобы сделать систему более гибкой и масштабируемой. Конфигурация представлена в разделах 8.2.2.2 и 8.2.2.3.
- *Шаг 11.* Создание AMI из работающего экземпляра. Последним шагом развертывания является создание нового AMI на основе работающего экземпляра портала Climate @ Home. Раздел 4.3.1 (Глава 4) описывает создание AMI. Образ можно повторно использовать для создания нескольких виртуальных экземпляров.

### **9.2.1.2 Развертывание BOINC-сервера**

Основные этапы развертывания сервера Climate @ Home BOINC показаны на рисунке 9.4. Пожалуйста, обратитесь к онлайн-видео Главы 9 для получения подробных инструкций и команд, используемых в следующих шагах.

- *Шаг 1.* Разрешить доступ к сети - должны быть открыты два порта: порт 22 и порт 80. Порт 22 используется для доступа к экземпляру через Secure Shell (SSH). Порт 80 предназначен для HTTP.
- *Шаг 2.* Запуск экземпляра - общедоступный AMI с Ubuntu 12.04 в качестве ОС выбирается в качестве базового образа для создания экземпляра для дальнейшей настройки. Поскольку сервер BOINC управляет интенсивной и высокой одновременной передачей данных между клиентом и сервером, конфигурация оборудования экземпляра имеет решающее значение.

150 Цзин Ли, Чжэньлун Ли, Мин Сун, Цюньин Хуан, Кай Лю и Мира Бамбакус

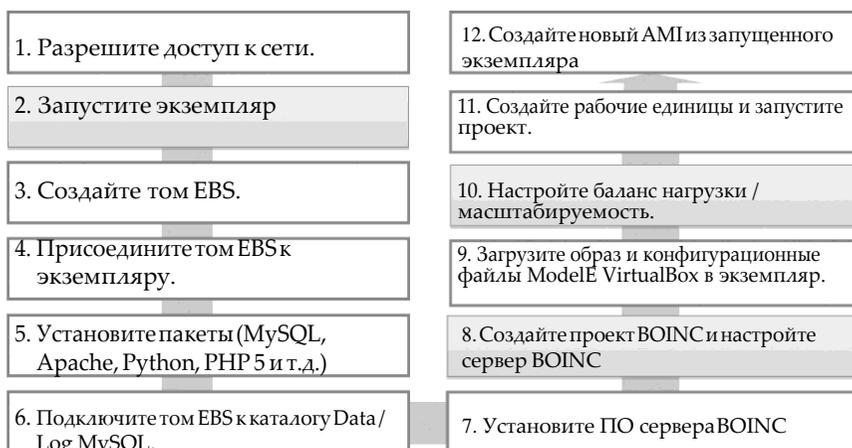


Рисунок 9.4 Процесс развертывания сервера BOINC на Amazon EC2. (Серые прямоугольники указывают на особые аспекты, описанные в Разделе 9.3.2.)

How to select an AMI for a BOINC server instance will be discussed in Section 9.2.2.

- Шаги 3, 4 и 6. Настройте EBS. При работе сервера Climate @ Home BOINC используется распределенный сервер FTPS для хранения загруженных результатов. Следовательно, EBS не является обязательным для сервера BOINC.
- Шаг 5. Установка пакетов программного обеспечения. Для работы BOINC необходимо несколько пакетов программного обеспечения, включая MySQL, веб-сервер Apache2, Python, php5 и другие.<sup>1</sup> Установка веб-сервера Apache2 и MySQL описана в главе 5, раздел 5.3.1. Для установки других пакетов, пожалуйста, обратитесь к онлайн-содержанию данной книги.
- Шаг 7. Установите программное обеспечение сервера BOINC. Подробные инструкции см. В онлайн-содержании данной книги.
- Шаг 8. Создайте проект BOINC и настройте сервер BOINC. По завершении установки сервера BOINC потребители должны создать тестовый проект BOINC, который распространяет приложения ModelE на вычислительные устройства, предоставленные гражданами.
- Шаг 9. Загрузите изображение ModelE VirtualBox и файлы конфигурации ModelE в экземпляр - на сервер должны быть загружены файлы двух типов. Образ VirtualBox ModelE помещается в папки приложений<sup>2</sup> сервера BOINC. Файлы конфигурации ModelE помещаются в папку загрузки проекта «climathome», созданного на шаге 8. См. Главу 4, раздел 4.3.1 для получения информации о загрузке файлов.

<sup>1</sup> См. BOINC Software Prerequisites на <http://boinc.berkeley.edu/trac/wiki/SoftwarePrereqsUnix>.

<sup>2</sup> См. BOINC Platforms на <http://boinc.berkeley.edu/trac/wiki/BoincPlatforms>.

- *Шаг 10.* Настройте балансировку нагрузки или масштабируемость - это необязательный шаг, чтобы сделать сервер BOINC более гибким и масштабируемым. Конфигурация представлена в Разделе 8.2.2.2 и Разделе 8.2.2.3.
- *Шаг 11.* Создайте рабочие единицы и запустите проект. Пользователи должны создать шаблоны ввода и вывода, чтобы начать проект.<sup>1</sup> В Приложении 9.1 показан пример шаблона ввода (E4M20a\_000040\_wu.xml), а в Приложении 9.2 показан шаблон вывода (climatehome\_re.xml). Используя два шаблона и файлы на шаге 9, потребители могут создать единицу работы и запустить проект, как показано ниже.

```
## Create work unit and start project
```

```
boincadm@ip-10-189-149-104:~/projects/climateathome $  
  bin/create_work -appname modelE -wu_name  
  E4M20a_000040e -wu_template templates/  
  E4M20a_000040_wu.xml -result_template templates/  
  climateathome_re.xml E4M20a_000040.R
```

```
boincadm@ip-10-189-149-104:~/projects/climateathome $  
  bin/start
```

- *Шаг 12.* Создайте АМІ из работающего экземпляра - на данном шаге создается новый АМІ на основе работающего экземпляра сервера BOINC для повторного использования. См. Главу 4, раздел 4.3 для создания АМІ

## 9.2.2 Специальные соображения

В соответствии с процедурами развертывания, сетевая связь и надежность виртуальных экземпляров имеют решающее значение для обеспечения успешной работы системы. При развертывании пространственного веб-портала и сервера BOINC на облачных сервисах следует уделять особое внимание следующим аспектам:

- *Конфигурация портала* - пространственный веб-портал может размещаться на различных экземплярах виртуальных машин. Для минимизации влияния серверов коммутационного хостинга рекомендуется два подхода: (1) использовать относительные пути вместо абсолютных и (2) использовать "локальный хост" в качестве учетной записи пользователя базы данных. Оба подхода обеспечивают безопасный доступ и разрешают конфликты при смене хостов.
- *Данные резервного копирования* и VMs-файлы данных MySQL хранятся на отдельном томе EBS для обеспечения постоянного хранения в случае сбоя экземпляра. В разделе 8.2 описывается процесс создания и прикрепления тома EBS к экземпляру портала Climate@Home.

<sup>1</sup>См. BOINC Input and Output Templates на <http://boinc.berkeley.edu/trac/wiki/JobTemplates>.

## 152 Цзин Ли, Чжэньлун Ли, Мин Сун, Цюньбин Хуан, Кай Лю и Мира Бамбакус

После того, как том EBS смонтирован в каталог данных и журналов MySQL, он может быть клонирован и использован для восстановления базы данных на втором экземпляре EC2. Том EBS может помочь восстановить другой портал Climate@Home в случае, если текущий запущенный портал в облаке развалится. Использование возможностей облака по запросу для хранения данных подробно описано в Главе 8, Раздел 8.2

- *Баланс нагрузки и автоматическое масштабирование* - для обработки массивных запросов система Climate@Home может воспользоваться преимуществами облачной коммутации по требованию. Баланс нагрузки и автоматическое масштабирование должны быть приняты во внимание. В разделе 8.2 главы 8 подробно описаны шаги по настройке баланса нагрузки и автомасштабирования для облачных сервисов.
- *Следует учитывать выбор АМІ для конфигурации BOINC сервер-сервер и аппаратные конфигурации виртуального экземпляра*: процессор, память и хранилище. В рамках системы Climate@Home сервер BOINC в основном выполняет функции планировщика задач, а запуски модели распределяются и выполняются на стороне клиента. Достаточно экземпляра с 8 гигабайтами (Гб) оперативной памяти и 2-ядерным процессором. Однако хранилище данных должно быть достаточно большим, чтобы терабайты - петабайты климатических наборов данных, которые загружаются с клиентской стороны. Amazon EBS, обеспечивающая высокодоступные, надежные и предсказуемые объемы хранения данных, должна использоваться в экземпляре EC2. Кроме того, в связи с массивной передачей данных на сервер BOINC предпочтительнее использовать экземпляр с высокой производительностью ввода-вывода через Интернет и высокой производительностью ввода-вывода в системе хранения данных.

### 9.2.3 Отличия от основных шагов, описанных в главе 5

По сравнению с общими шагами, рассмотренными в Главе 5, вся процедура развертывания системы Climate@Home в облачном сервисе имеет следующие характеристики:

- *Развертывание нескольких компонентов*. Данная система включает в себя сервер сетевого движка (BOINC-сервер) и пространственный Web-портал. Портал объединяет несколько сложных компонентов, таких как гео-аналитический портлет, интерфейсы для управления базами данных и CMS на основе Drupal. Настройка облачного сервера для размещения портала включает в себя конфигурацию нескольких типов программных пакетов.
- *Развертывание и настройка нескольких серверов*. Внутри пространственного веб-портала используются два различных контейнера сервлетов, которые представляют собой веб-сервер Apache2 и сервер Tomcat, для размещения двух различных компонентов. Вышеуказанные два контейнера могут вызывать конфликты, такие как расположение портов.

- Работа с вычислениями и интенсивность сети. Обработка интерактивной гео-визуальной аналитики на массивах данных является как коммуникационной, так и вычислительной интенсивностью. При развертывании необходимо учитывать масштабируемость и баланс нагрузки в ответ на массивный одновременный доступ. Кроме того, при выборе АМ необходимо учитывать эффективность обработки запросов.

### 9.3 ДЕМОНСТРАЦИЯ СИСТЕМЫ

В данном разделе описаны функции трех основных компонентов системы. Настоящий пространственный веб-портал, геовизуальный аналитический портлет и визуальный портлет управления ресурсами.

#### 9.3.1 Обзор пространственного веб-портала

Пространственный Web-портал представляет собой точку входа в проект для поддержки взаимодействия между системой и конечными пользователями (рис. 9.5). Он предлагает такие функции, как регистрация учетных записей пользователей, загрузка исполняемого файла VOINC и форумы для общения. Такие инструменты социальных сетей, как Facebook и Twitter, интегрированы для связи с порталом с целью обмена информацией с общественностью для просвещения и пропаганды. Использование двух основных порталов - портала визуального анализа и портала управления ресурсами - будет обсуждаться в следующих двух подразделах.

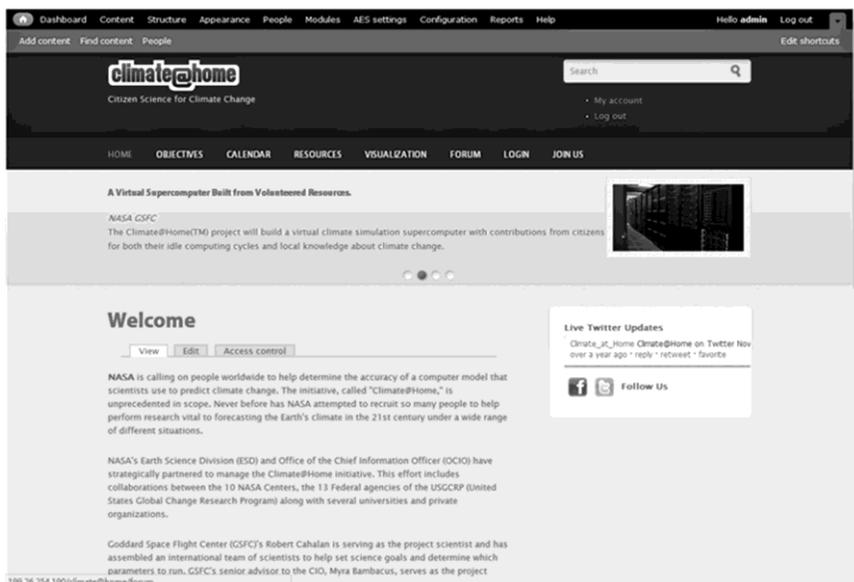


Рисунок 9.5 Главная страница пространственного веб-портала Climate@Home.

### 9.3.2 Геовизуальный аналитический портлет

Визуально-аналитический портлет обеспечивает ряд визуальных аналитических функций. Инструменты многомерной визуализации, включая NASA World Wind, Microsoft Bing Maps, Google Maps и Google Earth, интегрированы для облегчения визуальных исследований данных NetCDF. Операторы NetCDF (NCO) используются для поддержки функций статистического анализа, таких как расчет среднего значения, обнаружение аномалий и оценка качества моделирования. На рисунке 9.6 показано среднее десятилетнее значение сезонных (летних) осадков для двух выбранных областей с 1951 по 1960 год (Li и др., 2013).

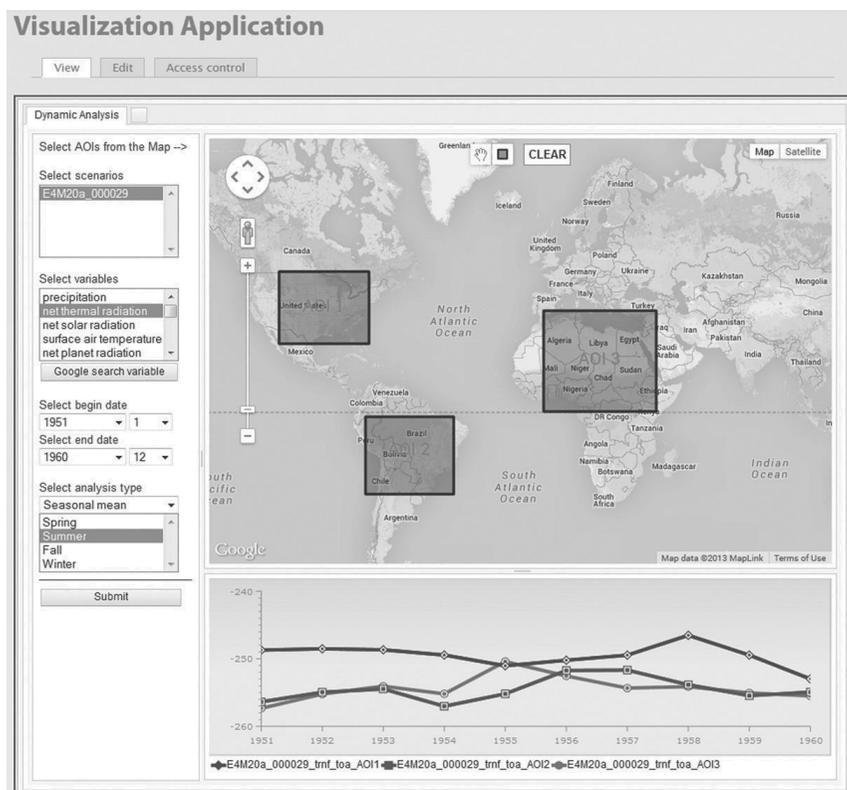


Рисунок 9.6 (См. цветную вставку) Анализ среднесезонных значений с использованием результатов моделирования.



Рисунок 9.7 (См. цветную вставку) Визуализация добровольных вычислительных узлов с помощью Google Maps.

### 9.3.3 the resource management portlet

Подобно визуально-аналитическому портлету, портлет управления ресурсами предназначен для мониторинга, управления и представления в реальном времени информации, связанной с запуском модели в визуальном виде. Типичными видами информации являются входные конфигурации ModelE, информация о пользователях, предоставляющих вычислительные ресурсы, состояние модели, запущенной на каждом клиентском вычислительном устройстве, и общее резюме модельных экспериментов. Информация о пользователе хранится в базе данных пространственного веб-портала при регистрации пользователя. Другая информация, связанная с вычислениями, собирается из базы данных сервера VOINC. На рисунке 9.7 показано распределение узлов добровольных вычислений по всему миру, а также конфигурация каждого узла и текущее рабочее состояние.

## 9.4 ЗАКЛЮЧЕНИЕ

Проект Climate@Home обеспечивает системное вычислительное решение и управление данными для содействия крупномасштабному и долгосрочному моделированию и анализу климата. В качестве двух основных компонентов системы пространственный веб-портал и сервер VOINC были развернуты для облачных сервисов. В таблице 9.1 приведена сводная информация об используемых при развертывании облачных сервисов. Облачная обработка данных расширяет возможности системы за счет создания надежной, масштабируемой и экономически эффективной киберинфраструктуры (Yang et al. 2011a).

156 Цзин Ли, Чжэньлун Ли, Мин Сун, Цюньин Хуан, Кай Лю и Мира Бамбакус

Таблица 9.1 Обзор облачных сервисов в системе Climate @ Home (стоимость оценена на основе облака AWS \*)

Вычис. сервис	Конфигурация	Сеть	Стоимость с AWS	Цель
Облачное хранение данных	10 Тб	C/O	\$0.080 за гигабайт в месяц (Amazon S3); \$0.095 за гигабайт в месяц (Amazon EBS)	Архивируйте выходные данные модели и результаты визуализации
Облачный веб-сервер 16 ГБ RAM	8-ядерный, 2,33 ГГц; 1G		0,299 \$ / час (аренда на 1 год) \$0,236 (аренда; 3 года); 0,5 \$ / час (почасовая оплата)	Разместите пространственный веб-портал
BOINC сервер 16G RAM	8-ядерный, 2,33 ГГц; 1G		0,299 \$ / час (аренда 1 год); 0,236 \$ (аренда 3 года); 0,5 \$ / час (почасовая оплата)	Сервер двигателя грид-машины

\*См. Cloud Computing Adoption Advisory Tool на <http://swp.gmu.edu:8080/ESIPCostModelProject/>.

- Использование облачного хранилища для архивирования наборов данных из моделирования и визуальной аналитики решает проблему масштабируемости управления "большими данными". Данный экономичный способ архивирования массивных данных с колебаниями объема данных (Zeng et al. 2009). Затраты на хранение результатов моделирования отличаются размером данных, продолжительностью использования и типами хранения (глава 6, раздел 6.2.2). Например, Amazon Simple Storage Service (S3) обычно взимает 0,095 долл. США/Гб в месяц (за Гб в течение одного месяца) за первые наборы данных объемом 1 Тб и 0,08 долл. Стоимость Amazon EBS относительно выше по сравнению с S3, потому что она может быть легко присоединена к экземпляру EC2 или отделена от него. Однако объем облачного хранилища данных может быть скорректирован в зависимости от объема выходных данных модели.
- Развертывание пространственного Web-портала в облачном сервисе управляет параллельным доступом к онлайн-портлету визуального геолого-аналитического анализа. Обеспечивая равновесие нагрузки и масштабируемость, "облачный" сервер может эффективно реагировать на массивные запросы (глава 17, раздел 17.3).

Многосайтовое развертывание может еще больше способствовать распределенному параллельному доступу, обеспечивая высокоскоростное сетевое соединение в глобальном масштабе. Кроме того, согласно анализу затрат, приведенному в главе 8, раздел 8.4, облачный хост веб-приложения будет экономить бюджет примерно на 57% ежегодно

- Развертывание сервера VOINC в облачной среде обеспечивает надежность, высокую производительность, безопасность и надежность управления симуляциями моделей. Сервер VOINC отвечает за управление вычислительными ресурсами и решение задач моделирования модели на рутинной основе. Надежность и устойчивость сервера VOINC имеют решающее значение при проведении огромного количества долгосрочных прогонов моделей. Большинство облачных сервисов гарантирует высокую степень доступности. Например, соглашение об уровне обслуживания Amazon EC2 (SLA) гарантирует доступность 99,95% для всех облачных центров обработки данных Amazon (глава 11, раздел 11.2.2).

Есть несколько вопросов, касающихся использования вычислительных технологий, которые заслуживают дальнейшего изучения:

- Разработайте в системе компонент оценки пользователей, который позволит пользователям, в том числе гражданским и ученым, тестировать функции и производительность системы. На основе статистики, полученной в результате оценки пользовательского опыта, исследователи могут лучше изучить систему и изменить существующие компоненты.
- Разработайте эффективную вычислительную структуру для использования разнородных вычислительных ресурсов, предоставленных гражданами. В дополнение к доступности вычислительных устройств, предоставленных гражданами, механизм энергосистемы должен распределять задачи на основе надежности, исторической производительности и сетевого подключения этих вычислительных устройств.
- Разработайте и реализуйте облачную стратегию управления распределенными данными. В данном проекте облачное хранилище использовалось для хранения наборов данных. Масштабируемость и эластичность облачных сервисов хранения еще полностью не исследованы.
- Изучите оптимальные решения для развертывания пространственного веб-портала, основанные на пространственно-временных принципах. Все сценарии развертывания, обсуждаемые в этой главе, являются относительно стандартными процедурами без учета пространственно-временных принципов, таких как шаблоны доступа пользователей. Такие принципы также являются критическими факторами, влияющими на производительность системы (Yang et al. 2011b).

## 9.5 КРАТКОЕ ИЗЛОЖЕНИЕ

В настоящей главе описывается разработка комплексной системы для облегчения моделирования климата с акцентом на то, как облачные сервисы могут использоваться для расширения возможностей системы. Раздел 9.1 представляет исходные данные,

158 Цзин Ли, Чжэньлун Ли, Мин Сун, Цюньбин Хуан, Кай Лю и Мира Бамбакус

---

сложности и компоненты системы Climate@Home. В разделе 9.2 описаны процедуры развертывания системы в облачных сервисах с упором на специальные процедуры и соображения. Демонстрации выбранных компонентов обсуждаются в Разделе 9.3. В разделе 9.4 суммированы преимущества использования облачных сервисов для поддержки системы, а также потребности будущих исследований

## 9.6 ПРОБЛЕМЫ

1. На примере системы Climate@Home перечислите основные компоненты системы, необходимые для поддержки моделирования и анализа климата в распределенной среде.
2. На примере разработки системы Climate@Home объясните, почему методы облачных вычислений важны для поддержки моделирования и анализа климата в распределенной среде.
3. Каковы основные этапы развертывания пространственного веб-портала в облачных сервисах? Какие особые соображения касаются сетевых конфигураций в процессе развертывания пространственного веб-портала?
4. Каковы основные этапы развертывания сервера BOINC в облачных сервисах? Какие особые соображения нужно учитывать по сравнению с общим развертыванием приложения в главе 5?
5. Каковы требования к аппаратной конфигурации виртуальной машины, выступающей в качестве сервера BOINC?
6. Каково использование хранилища гибких блоков (EBS) в облачной системе Climate@Home?
7. Перечислите облачные сервисы, используемые при развертывании системы Climate@Home.
8. Каковы оставшиеся проблемы при развертывании компонентов Climate@Home в облаке?

## ПРИЛОЖЕНИЕ 9.1 e4m20a\_000040\_Wu.xml

```
<!-- workunit template for the BOINC E4M20a_000040
      program -->
<file_info>
  <number>0</number>
</file_info>
<file_info>
  <number>1</number>
</file_info>
<workunit>
```

```
<file_ref>
  <file_number>0</file_number>
  <open_name>shared/boinc_app</open_name>
  <copy_file>1</copy_file>
</file_ref>
<file_ref>
  <file_number>1</file_number>
  <open_name>shared/E4M20a_000040.R</open_name>
  <copy_file>1</copy_file>
</file_ref>
<min_quorum> 1 </min_quorum>
<rsc_disk_bound> 1000000000 </rsc_disk_bound>
<rsc_fpop_est>144400000000000</rsc_fpop_est>
<rsc_fpop_bound>188800000000000</rsc_fpop_bound>
<credit>3072</credit>
</workunit>
```

## ПРИЛОЖЕНИЕ 92 Climateathome\_re.xml

```
<file_info>
  <name><OUTFILE_0/></name>
  <generated_locally/>
  <upload_when_present/>
  <max_nbytes>8000000000</max_nbytes>
  <url><UPLOAD_URL/></url>
</file_info>

<result>
  <file_ref>
    <file_name><OUTFILE_0/></file_name>
    <open_name>shared/out.tar.gz</open_name>
    <optional>1</optional>
    <no_validate>1</no_validate>
    <copy_file/>
  </file_ref>
</result>
```

## СПИСОК ЛИТЕРАТУРЫ

- Anderson, D. P. 2004. BOINC: A system for public-resource computing and storage. In *Grid Computing. Proceedings of 5th IEEE/ACM International Workshop*, pp. 4–10.
- Anderson, D. P. 2010. Volunteer Computing with BOINC. In Cerin, C., and Fedak, G., eds. *Desktop Grid Computing*. Boca Raton, FL: CRC Press/Chapman & Hall, pp. 3–25.

- Li, Z., C. Yang, M. Sun et al. 2013. A High Performance Web-Based System for Analyzing and Visualizing Spatiotemporal Data for Climate Studies. In S. Liang, X. Wang, C. Claramunt, eds. *Web and Wireless Geographical Information Systems*, pp. 190–198. Heidelberg: Springer Berlin.
- Santos, E., J. Poco, Y. Wei et al. 2013. UV-CDAT: Analyzing climate datasets from a user's perspective. *Computing in Science & Engineering* 15, no. 1: 94–103.
- Schmidt, G. A., R. Ruedy, J. E. Hansen et al. 2006. Present-day atmospheric simulations using GISS ModelE: Comparison to *in situ*, satellite, and reanalysis data. *Journal of Climate* 19, no. 2: 153–192.
- Skamarock, W. C. and J. B. Klemp. 2008. A time-split nonhydrostatic atmospheric model for weather research and forecasting applications. *Journal of Computational Physics* 227, no. 7: 3465–3485.
- Sun, M., J. Li, C. Yang et al. 2012. A Web-based geovisual analytical system for climate studies. *Future Internet* 4, no. 4: 1069–1085.
- Taylor, K. E. 2001. Summarizing multiple aspects of model performance in a single diagram. *Journal of Geophysical Research: Atmospheres (1984–2012)* 106, no. D7: 7183–7192.
- Yang P., Evans J., Cole M., Alameh N., Marley S., and Bambacus M. 2007. The emerging concepts and applications of the spatial Web portal. *Photogrammetry & Remote Sensing* 73, no. 6: 691–698.
- Yang, C., M. Goodchild, Q. Huang et al. 2011a. Spatial cloud computing: How can the geospatial sciences use and help shape cloud computing? *IJDE* 4, no. 4: 305–329.
- Yang C., H. Wu, Z. Li, Q. Huang, and J. Li. 2011b. Utilizing spatial principles to optimize distributed computing for enabling physical science discoveries. *Proceedings of National Academy of Sciences* 108, no. 14: 5498–5503.
- Zeng, W., Y. Zhao, K. Ou et al. 2009. Research on cloud storage architecture and key technologies. In *Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human*, pp. 1044–1048. ACM.

## Глава 10

---

# Прогнозирование пыльных бурь с облачной поддержкой

*Цюнбин Хуан, Цзичжэ Ся, Манжу Ю,  
Карл Бенедикт и Мира Бамбакс*

---

Прогнозирование пыльных бурь в реальном времени - типичное приложение для вычисления интенсивности, которое требует быстрого моделирования с высоким разрешением и большой географической территории. В настоящей главе рассказывается, как использовать облачные сервисы для поддержки моделирования и прогнозирования пыльных бурь.

## 10.1 МОДЕЛИРОВАНИЕ ПЫЛЬНЫХ БУРЬ: ИСХОДНЫЕ ДАННЫЕ И СЛОЖНОСТИ

### 10.1.1 Исходные данные

Пыльные бури возникают в результате сильных турбулентных ветровых систем, уносящих частицы пыли в воздух и снижающих видимость с нескольких миль до нескольких метров (Goudie and Middleton 1992). Глобальное изменение климата увеличило частоту и интенсивность пыльных бурь в последние десятилетия с негативными последствиями для окружающей среды, здоровья человека и материальных ценностей. Например, сильные пыльные бури обрушиваются на юго-запад Соединенных Штатов несколько раз в год и вызывают крупные автомобильные аварии, материальный ущерб, травмы и гибель людей (Shoemaker and Davis 2008). Негативное воздействие пыльных бурь побудило ученых разработать модели, чтобы лучше понять и предсказать распределение и интенсивность выбросов, осаждения и структуры пыли. С конца 1980-х годов несколько исследовательских групп разработали модели пыли, которые могут правильно предсказывать пространственно-временные закономерности, эволюцию и порядок величины концентрации, выбросов и осаждения пыли (Huang et al. 2013a). В дополнение к прогнозированию погоды и климата моделирование и прогнозирование пыльных бурь также можно использовать на практике (WMO 2012) для: (1) определения важных последствий качества воздуха для различных групп населения, от населения до промышленности; (2) помочь службам общественного здравоохранения в улучшении оперативного эпиднадзора за болезнями и раннем обнаружении; (3) повысить безопасность воздушного и дорожного движения и управление аварийными ситуациями; и (4) служить в качестве предупреждающей, консультативной и оценочной системы, чтобы дать инструкции для адаптивных действий, таких как изменение времени планирования, укрепление инфраструктуры и строительство ветрозащитных полос и защитных полос.

### 10.1.2 Сложности

Использование существующих моделей для прогнозирования пыльных бурь с высоким разрешением для больших географических областей создает несколько критических проблем:

- Моделирование явления пыльной бури - сложная задача и требует больших вычислительных ресурсов (Xie et al. 2010). Моделирование периодических явлений требует многократного повторения одного и того же набора численных расчетов с интервалом времени. На рисунке 10.1 показана типичная процедура для негидростатической мезомасштабной модели (NMM) - пылевой модели (Huang et al. 2013a), которая является популярной моделью прогнозирования пыльных бурь. Каждая итерация включает расчет 21 модуля. Для заданного размера области вычислительная сложность модели атмосферы зависит от того, где  $n$  - размерность грид, включая одно временное измерение, два горизонтальных измерения и одно вертикальное измерение (Baillie et al. 1995). Если временное измерение расширено, потребуется больше итераций. Удвоение географической области в горизонтальном направлении приведет к четырехкратному увеличению затрат на вычисления. Удвоение пространственного разрешения (например, с 10 км до 5 км) приведет к восьмикратному увеличению вычислительных затрат, поскольку для сохранения точности модели также потребуется удвоение шага по времени (Baillie et al. 1995). Высокопроизводительные вычисления используются для моделирования пыльной бури для больших географических областей с высоким разрешением (Huang et al. 2013a). NMM-dust (Huang et al. 2013a) проводят параллель на высокопроизводительные вычисления (HPC) путем декомпозиции домена на несколько поддоменов и распределения вычислительной нагрузки каждого поддомена на одно ядро ЦП как процесс. Ядро, обрабатывающее субдомен, должно взаимодействовать с соседними ядрами для синхронизации. Подпрограммы, показанные на рисунке 10.1 в серых прямоугольниках, при распараллеливании требуют интенсивного обмена данными и синхронизации (Huang et al. 2013a).

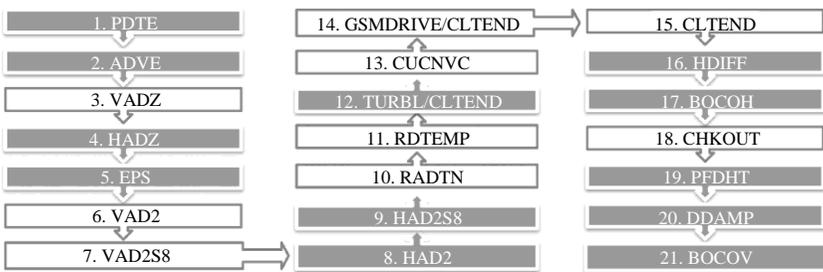


Рисунок 10.1 Вычислительные подпрограммы для модели NMM-dust. (Подпрограммы в серых прямоугольниках требуют связи и синхронизации.) (Из Huang et al. 2013a.)

## Прогнозирование пыльных бурь соблочной поддержкой 163

- Прогнозирование пыльных бурь - это критичная по времени задача, которую необходимо выполнить в ограниченные сроки. Например, для однодневного прогнозирования рекомендуется двухчасовой предел вычислений, чтобы результаты были точными (Drake and Foster 1995). Прогнозирование ограниченного географического района и / или разрешения обычно выполняется для завершения моделирования в отведенные сроки (Wolters, Cats, and Gustafsson 1995). Однако разрешение на уровне местности необходимо для прогнозирования пыльной бури, чтобы поддержать принятие чрезвычайных решений государственными учреждениями, например, для подготовки лекарств для общественного здравоохранения (Xie et al. 2010). На рисунке 10.2 показано время вычислений, необходимое для 24-часового прогнозирования для разных размеров области в горизонтальных направлениях, с одинаковыми вертикальными слоями (37 слоев) и пространственным разрешением (3 км). Более 12,6 часов требуется для прогнозирования размера домена  $10 \times 10$  градусов с использованием кластера НРС с 25 вычислительными узлами (Yang et al. 2011). Подсчитано, что для прогнозирования всего юго-запада США с размером области  $37 \times 20$  градусов потребуется около 93 часов. Такая вычислительная производительность неприемлема, потому что мы прогнозировали бы явление пыли в прошлом.
- Пыльные бури - разрушительные явления. Подсчитано, что общее время пыльных бурь за один год обычно составляло менее 90 часов и занимает менее 1% от одного года, если предположить, что каждая пыльная буря длится в среднем два часа (NOAA 2011). Следовательно, система прогнозирования таких событий будет ожидать разные требования к вычислениям и доступу в разное время года и даже в разные часы в течение дня.

Для решения таких проблем можно использовать облачную обработку данных. Облачные сервисы могут справиться с резкими скачками и разрушительными требованиями к вычислениям с помощью эластичных вычислительных ресурсов и ресурсов по требованию (Huang et al. 2013b). Кроме того, с развитием технологий облачной обработки данных, таких как виртуализация, и общим развитием облачной инфраструктуры, облачные сервисы готовы поддерживать приложения с интенсивными вычислениями.



Рисунок 10.2 Время выполнения прогнозов для различных географических доменов. (Из Huang et al. 2013b.)

Например, Amazon EC2 предлагает экземпляры кластера с сетевым подключением 10 Гбит / с. Каждый экземпляр кластера имеет минимум 23 ГБ ОЗУ и два четырехъядерных процессора с тактовой частотой 2,93 ГГц. Такие конфигурации оборудования и сети намного лучше, чем вычислительные среды grid с гетерогенными вычислительными ресурсами и подключением к глобальной сети (WAN), и даже лучше, чем большинство частных однородных кластерных конфигураций HPC. Это делает облачную обработку данных новой выгодной вычислительной парадигмой для решения научных задач, которые требуют больших вычислительных ресурсов и традиционно требуют специального высокопроизводительного кластера (Rehr et al. 2010).

## 10.2 РАЗВЕРТЫВАНИЕ И ОПТИМИЗАЦИЯ

### 10.2.1 Общий рабочий процесс

Общие шаги по развертыванию модели пыльной бури на Amazon EC2 следующие (рисунок 10.3):

*Шаг 1.* Авторизация доступа к сети. Авторизация доступа к сети была представлена в главе 4. Порт 22 открыт для обеспечения связи между локальным сервером и экземплярами EC2 через Secure Shell (SSH).

*Шаг 2.* Запуск экземпляра - запуск экземпляра в качестве головного узла задачи моделирования пыльной бури (для справки см. Главу 4, раздел 4.3). Выбран экземпляр кластера с восемью ядрами ЦП и 23 ГБ памяти.

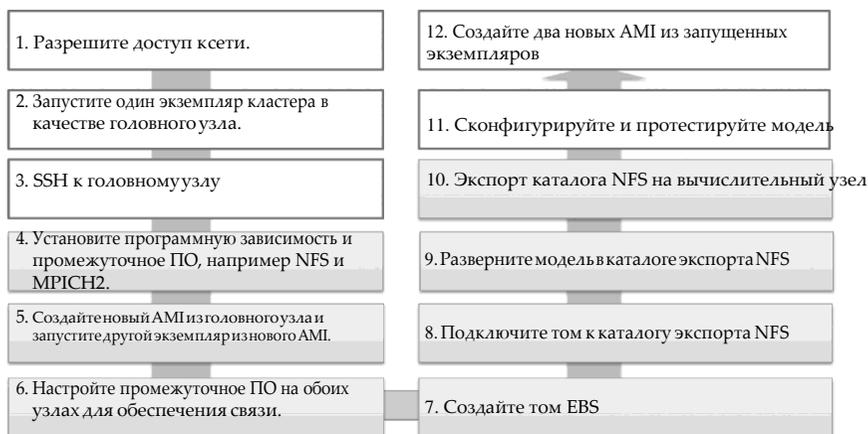


Рисунок 10.3 Процесс развертывания модели пыльной бури на Amazon EC2. (Серые прямоугольники указывают на то, что этот шаг требует особого внимания.)

*Шаг 3.* SSH к экземпляру. Используйте инструмент SSH для доступа к экземпляру, созданному на шаге 2. Доступ к экземпляру подробно обсуждался в главе 4, раздел 4.3.

*Шаг 4.* Установите зависимости программного обеспечения. Для выполнения модели пыльной бури необходимо, чтобы сетевая файловая система (NFS) и MPICH2 были предварительно установлены на головном узле и вычислительных узлах. NFS позволяет всем другим вычислительным экземплярам совместно использовать программный пакет и данные модели. Используя NFS, нам нужно только настроить модель на головном узле, в то время как другие вычислительные экземпляры могут совместно использовать эту среду без установки и настройки среды запуска модели. MPICH2 - одна из самых популярных реализаций интерфейса передачи сообщений (MPI). В следующей командной строке описан процесс установки NFS и MPICH2 в экземпляре Amazon EC2 с CentOS 5.6 в качестве операционной системы

```
[root@domU-head~] yum install gcc gcc-c++ autoconf
automake
[root@domU-head~] yum -y install nfs-utilsnfs-utils-
lib sytem-config-nfs #install NFS
[root@domU-head~] tar -zvxf mpich2-1.5.tar.gz
download MPICH2 package and unzip
[root@domU-head~] mkdir /home/clouduser/mpich2-
install # create an installation directory
[root@domU-head~] cd mpich2-1.5.tar.gz
[root@domU-head~] ./configure --prefix=/home/
clouduser/mpich2-install
[root@domU-head~] make # Build MPICH2
[root@domU-head~] make install # Install MPICH2
```

После установки NFS для настройки и запуска NFS можно использовать следующие команды.

```
[root@domU-head ~] mkdir /headMnt # create a NFS
export directory
[root@domU-head ~] echo "/headMnt *(rw) " >> /etc/
exports
[root@domU-head ~] exportfs -ra
[root@domU-head ~] service nfs start #start up NFS
```

*Шаг 5.* Создайте AMI и вычислительный узел. Создайте AMI на основе настроенного экземпляра на шаге 4. Используйте данный новый AMI для запуска одного экземпляра в качестве вычислительного узла.

*Шаг 6.* Сконфигурируйте головной узел и вычислительный узел - на этом этапе устанавливается сетевое соединение между головным узлом и вычислительными узлами. Сначала используйте SSH для доступа к головному узлу и добавьте IP-адрес вычислительного узла в список узлов.

<sup>1</sup>См. MPICH2 на <http://www.mcs.anl.gov/research/projects/mpich2staging/goodell/index.php>.

Затем создайте открытый ключ в головном узле. Наконец, используйте SSH для доступа к каждому вычислительному узлу и скопируйте открытый ключ головного узла в список авторизованных ключей.

```
[root@domU-headMnt ~] vi /etc/hosts #access to the
hosts list of the head node
[root@domU-headMnt ~] ssh-keygen -t rsa #create a
public key at the head node

[root@domU-computing~] mkdir -p /root/.ssh/
[root@domU-computing ~] scp root@domU-head: /root/.
ssh/id_ras.pub /root/.ssh/ #copy the public key
from the head node to the computing node
[root@domU-computing ~] cat /root/.ssh/id_ras.pub >>
/root/.ssh/authorized_keys
```

*Шаг 7.* Создайте том EBS и присоедините его к головному узлу. Информацию о создании нового тома EBS и его присоединении к работающему экземпляру см. В главе 8, раздел 8.3.

*Шаг 8.* Подключите том к каталогу экспорта NFS - см. Главу 8, раздел 8.3, чтобы узнать о подключении тома к экземпляру.

*Шаг 9.* Разверните модель в каталоге экспорта NFS. Перенесите модель пыльнойбури (см. Главу 5, раздел 5.3.1 для передачи больших объемов данных между локальным сервером и экземплярами EC2) в каталог экспорта NFS головного узла, которым является /headMnt. в этом случае.

*Шаг 10.* Экспорт каталога NFS на вычислительный узел - на данном этапе устанавливается соединение NFS между головным и вычислительным узлами, что позволяет вычислительному узлу совместно использовать комплекты пакетов из головного узла. Используйте SSH для доступа к вычислительному узлу и смонтируйте том в каталог NFS. В приведенной ниже команде показан пример монтирования тома вычислительного узла в каталог NFS. domU-head - доменное имя головного узла. / headMnt - это путь к файлу каталога NFS в головном и вычислительном узлах. После выполнения команды для монтирования каталога NFS от головного узла к вычислительному узлу потребители облака могут перейти в каталог / headMnt и проверить, доступна ли модель на вычислительном узле.

```
[root@domU-computing ~] mkdir //headMnt # create the
directory
[root@domU-computing ~] mount -t nfs -o rw domU-head:/
headMnt //headMnt #Mount the volume to the NFS export
directory
```

## Прогнозирование пыльных бурь с облачной поддержкой 167

*Шаг 11.* Настройте и протестируйте модель - создайте файл хоста со списком IP-адресов или доменных имен, которые могут запускать задачи MPI (например, `mpd.Hosts`), и добавьте частный IP-адрес каждого вычислительного узла в этот файл, как показано ниже (включая два вычислительных узла).

```
199.26.254.161 # node1
199.26.254.162 # node2
```

На головном узле запустите соединение MPI. Приведенная ниже команда показывает, как запустить сервис MPICH2. Параметр `-n 3` - это количество вычислительных узлов плюс головной узел, а `mpd.hosts` - файл конфигурации машины. В приведенной ниже команде также показан пример выполнения сценария модели: `run_test.sh` - сценарий выполнения модели, а `out.log` - файл журнала для хранения информации о выполнении модели

```
[root@domU-head~] mpdboot -n 3 -f mpd.hosts -v #
start up MPICH2
[root@domU-xxx~] ./run_test.sh >& out.log & # Run the
model
```

*Шаг 12.* *Создайте АМІ из работающего экземпляра* - наконец, два новых АМІ могут быть созданы отдельно на основе работающего головного узла и вычислительного узла (см. Главу 4, раздел 4.3, чтобы узнать, как создать АМІ).

### 10.2.2 Специальные соображения

Помимо общих этапов развертывания, для моделей пыльных бурь с высокими требованиями к вычислительной мощности характерны несколько конфигураций.

- Настройка виртуальной среды кластера для поддержки прогнозирования пыльной бури с облачными сервисами, виртуальная среда кластера должна быть установлена. Как введено в главе 5, раздел 5.3, один или несколько виртуальных экземпляров следует использовать с установленными промежуточным программным обеспечением, и выполненными с возможностью формирования виртуального кластера. В данной главе MPICH2 используется в качестве промежуточного слоя (Глава 5 использует Condor), чтобы разослать задачи работающей модели и собирать результаты. Чтобы сэкономить время на конфигурирование вычислительных узлов, изображение из головного узла на шаге 5 (рис. 10.3) может быть построено для запуска вычислительного узла. Кроме того, накладные расходы на связь можно значительно сократить, если все экземпляры запускаются в одних и тех же центрах обработки данных и на физически близких машинах. Следовательно, потребители облака должны создать группу размещения и запустить несколько экземпляров кластера в группу размещения (рисунок 10.4). Группа размещения кластера - это логическая группа экземпляров кластера в одном центре обработки данных ЕС2.



Рисунок 10.4 (См. цветную вставку) Создайте группу размещения и поместите экземпляры в одну группу.

Группа размещения не может охватывать несколько центров обработки данных. Таким образом, потребители могут логически группировать экземпляры в кластеры в одном центре обработки данных. Это позволяет многим приложениям НРС получить полную пропускную способность и производительность сети с малой задержкой, необходимые для тесной связи между узлами. Amazon рекомендует потребителям облака запускать минимальное количество экземпляров, необходимое для поддержки приложений в группе размещения в одном запросе на запуск. Если потребители запускают только несколько экземпляров и попытаются добавить больше экземпляров в группу размещения позже, они могут получить ошибку «Недостаточно емкости», поскольку отсутствуют физические ресурсы для создания виртуальных машин в той же стойке НРС. Если потребители все же получают сообщение об ошибке «Недостаточно емкости», остановите и перезапустите экземпляры в группе размещения и попробуйте снова добавить экземпляры.

Пользователи облака также могут использовать следующую командную строку для создания группы размещения (например, VirtualCluster) и размещения всех экземпляров (например, двух экземпляров в данном случае в одной группе).

```
PROMPT>ec2-create-placement-group VirtualCluster -s  
cluster  
PROMPT>ec2-run-instances ami-xxx -n 2 --placement-  
group VirtualCluster
```

- Слабосвязанное вложенное моделирование и облачная обработка данных. Во время теста слабосвязанной вложенной модели (Huang et al. 2013a) большой домен сокращается на несколько небольших областей, которые идентифицируются грубой моделью ETA-8bin. Для одновременного запуска этих 18 областей интереса (AOI) необходимо задействовать большое количество вычислительных ресурсов, каждая из которых требует прогнозирования с высоким разрешением.

<sup>1</sup> См. AWS Elastic Cloud Compute на [http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using\\_cluster\\_computing.html](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using_cluster_computing.html).

Гибкие экземпляры кластера Amazon EC2 используются для полной реализации подхода вложенного моделирования. Это связано с тем, что на платформе Amazon EC2 можно создать несколько виртуальных кластеров, при этом каждый виртуальный кластер будет запускать один субрегион напрямую (подробности о производительности описаны в разделе 10.3.3).

- *Автоматическое масштабирование* — Поскольку это не веб-приложение, метод запуска автоматического масштабирования ресурсов через CloudFormation (глава 8, раздел 8.3) не подходит. Пользователю облака придется писать сценарии с API EC2, чтобы запускать больше экземпляров для обработки вычислений, при необходимости.

### 10.2.3 Краткое изложение отличий от общих шагов в главе 5

По сравнению с общими шагами, описанными в главе 5, вся процедура развертывания модели пыльной бури в облачном сервисе имеет следующие характеристики:

- *Управление интенсивностью вычислений.* Данная модель пыльной бури требует использования нескольких виртуальных экземпляров для одновременного запуска моделирования. Следовательно, виртуальная кластерная среда должна быть настроена с установленным промежуточным программным обеспечением MPICH2 и правильно настроено на головном узле и вычислительных узлах.
- *Обработка интенсивности взаимодействия* - следует создать группу размещения (рисунок 10.4), и все экземпляры для виртуального кластера должны находиться в одной группе, чтобы уменьшить накладные расходы на связь.

## 10.3 ДЕМОНСТРАЦИЯ

### 10.3.1 Пыльные бури в г.Финикс

Штат Аризона является домом для одних из самых впечатляющих проявлений пыльных бурь в Соединенных Штатах. Опасные пыльные бури обрушиваются на штат несколько раз в год и вызывают многочисленные экономические потери (Shoemaker and Davis 2008). Самая крупная из этих пыльных бурь называется хабубом, наиболее распространенным в центральных пустынях Аризоны в течение летнего сезона, с пиком повторяемости в конце июля и начале августа. В городе Финикс, штат Аризона, с июня по сентябрь в среднем происходит около трех хабубов в год (Shoemaker and Davis 2008). На рис. 10.5 показан классический хабуб, который прошел через область Финикса 1 июля 2007 года. В этой главе это событие используется в качестве примера, чтобы показать, как модель пыльной бури может имитировать явление пыльной бури и как облачные вычисления могут помочь в выполнении такого моделирования.



*Рисунок 10.5 (См. цветную вставку)* Фотография хабуба, поразившего Финикс, штат Аризона, 1 июля 2007 г. (любезно предоставлено Osha Gray Davidson, Mother Jones, Сан-Франциско, Калифорния, <http://www.motherjones.com/blue-marble/2009/09/australias-климат-хаос.>)

### 10.3.2 Результат моделирования

На рисунке 10.6 показан временной интервал в 3 часа ночи 2 июля 2007 г., полученный ETA-8bin (модель с низким разрешением) (Huang et al. 2013a) и моделями пыли NMM с различным пространственным разрешением. Результаты модели показывают аналогичные закономерности для области пыльной бури и результатов NMM-dust с разрешением 3 км, которые собрали гораздо более подробную информацию о концентрациях пыли.

### 10.3.3 Эффективность

Гибкие экземпляры кластера Amazon EC2 используются для полной реализации подхода слабосвязанного вложения (Huang et al. 2013b). При использовании метода размещения со слабой связью ETA-8bin выполняет быстрое прогнозирование с низким разрешением (22 км) для выявления потенциальных горячих точек с высокой концентрацией пыли. NMM-dust будет выполнять прогнозирование с высоким разрешением (3 км) для гораздо меньших горячих точек параллельно, чтобы сократить вычислительные требования и время вычислений.

В данном случае 18 АОИ идентифицируются грубой моделью ETA-8bin. На рис. 10.7 показано распределение этих 18 АОИ по ширине и длине. Замечено, что большинство из них распределяются в пределах  $2 \times 2$  градуса. Таким образом, 18 экземпляров кластера запускаются из АМІ, настроенного в соответствии с шагами, описанными в разделе 10.2, и каждый экземпляр отвечает за обработку моделирования для одного АОИ.

На рис. 10.8 показано время выполнения для каждого экземпляра по 18 экземплярам Amazon, каждый из которых моделирует один регион АОИ для 24-часового прогноза. Результаты показывают, что большинство АОИ могут быть успешно выполнены в течение одного часа при 24-часовом прогнозировании.

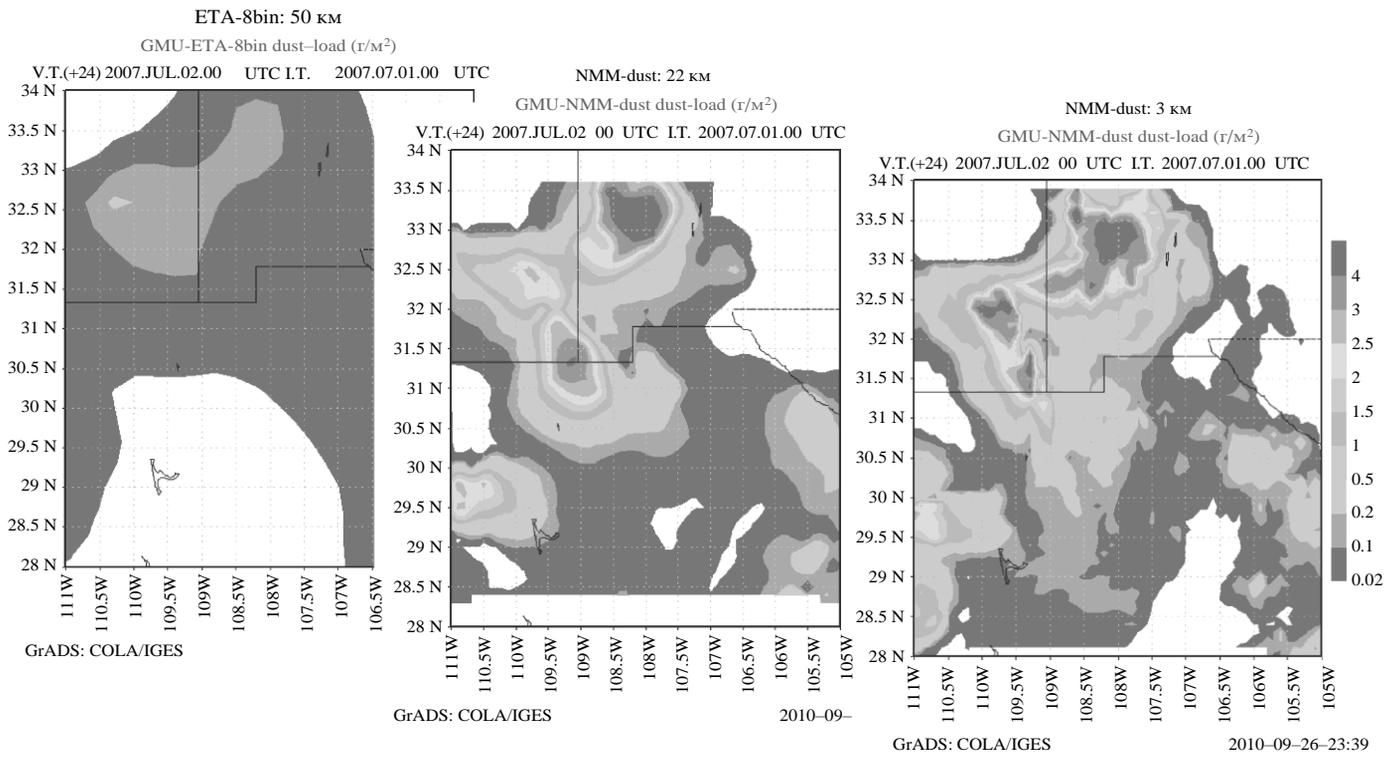


Рисунок 10.6 (См. цветную вставку) Сравнение результатов моделирования ETA-8bin и NMM-пыль АОИ 10, 11, 12 и 13 в 3 часа ночи 2

июля 2007г.

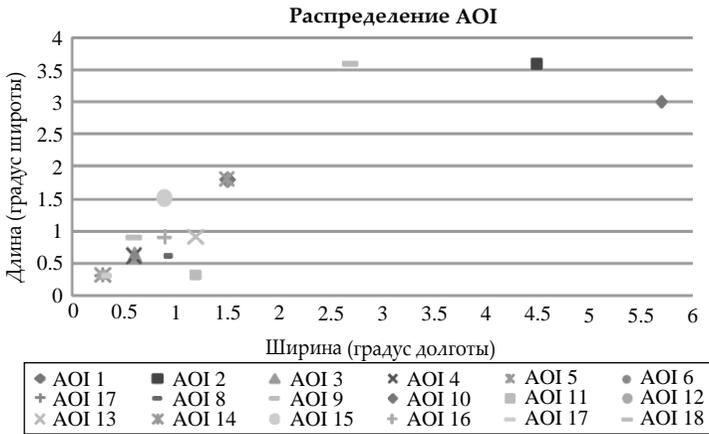


Рисунок 10.7 (См. цветную вставку) Распределение АОI, идентифицированное ETA-8bin.

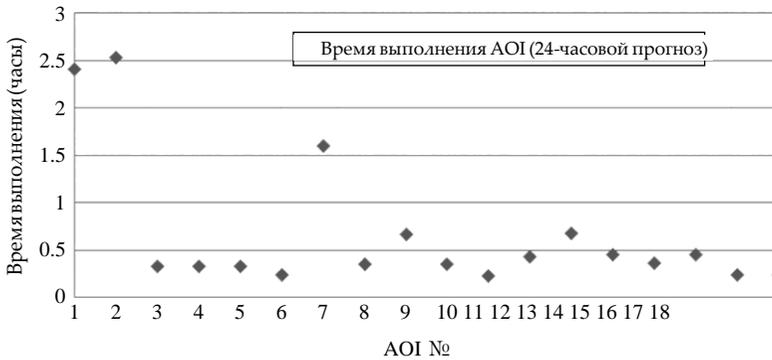


Рисунок 10.8 Восемнадцать кластерных экземпляров Amazon запускаются для запуска 18 АОI параллельно, причем каждый экземпляр моделирует один регион АОI для 24-часового прогноза.

Однако два АОI не могут быть успешно завершены в течение двух часов. Следовательно, необходимо интегрировать больше вычислительных ресурсов и оптимизаций, чтобы позволить вычислениям завершиться в течение 2 часов при 24-часовом ограничении по времени моделирования. Два и три экземпляра кластера используются для дальнейшего тестирования вычислимости этих двух подобластей. На рис. 10.9 показана производительность облачных сервисов Amazon EC2 при использовании двух и трех экземпляров для каждого виртуального кластера с оптимизацией для трехчасового прогнозирования и без для первого субрегиона. Замечено, что система не может достичь 0,25-часового ограничения по времени даже с тремя задействованными экземплярами кластера. Следовательно, такие два экземпляра следует оптимизировать за счет лучшего распараллеливания и стратегий планирования, включая сопоставление схожих карт и стратегии пространственно-временной оптимизации (Yang et al. 2011).

## Прогнозирование пыльных бурь с облачной поддержкой 173

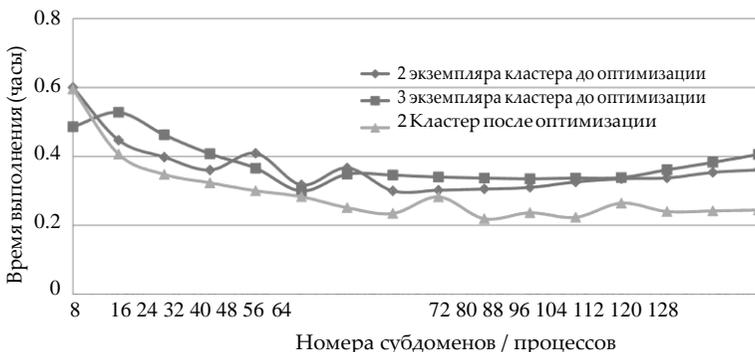


Рисунок 10.9 Сравнение производительности экземпляров кластера Amazon до и после оптимизации.

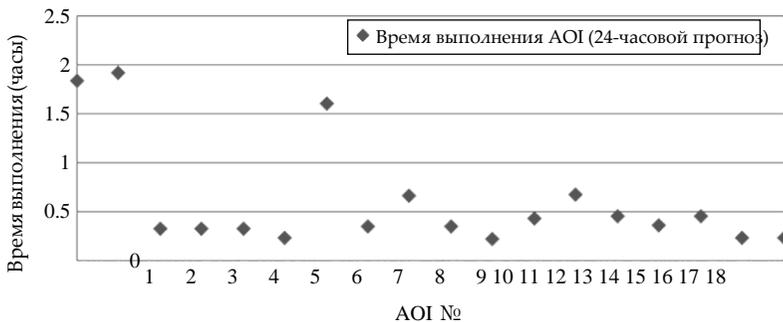


Рисунок 10.10 После оптимизации в облачном сервисе Amazon EC2 работают 18 субрегионов. Оба АОИ 1 и 2 используют два оптимизированных экземпляра кластера.

На рисунке 10.9 показано, что два оптимизированных экземпляра могут успешно завершить прогнозирование в течение 0,25 часа при использовании более 64 процессов. Наконец, с этими двумя экземплярами, запускающими 64 процесса, можно заметить, что два АОИ могут быть успешно завершены в течение двух часов (рисунок 10.10).

### 10.3.4 Экономическая эффективность

Для оперативной системы прогнозирования пыльных бурь обычно экземпляр EC2 резервируется для долгосрочного прогнозирования по гораздо более низкой цене по сравнению с экземпляром по запросу. Стоимость составляет всего 0,16 доллара в час для общего доступа пользователей и прогнозирования с низким разрешением. Когда происходят события пыльной бури, запускается большая группа экземпляров, чтобы реагировать на одновременный доступ пользователей и прогнозирование с высоким разрешением.

Было выявлено, что почасовая стоимость данного кластера намного выше, чем у облачного сервиса EC2, когда не происходит пыльная буря и требуется только один зарезервированный экземпляр (Huang et al. 2013b). В такой ситуации почасовые расходы кластера более чем в 896 раз превышают почасовые затраты на один зарезервированный экземпляр.

Как правило, всего в США происходит около 45 пыльных бурь в год (NOAA 2011). Годовая стоимость локального кластера примерно в 12,7 раз выше, чем у облачной службы EC2, если использовать 28 экземпляров EC2 (с 400 ядрами ЦП) для обработки требований высокого разрешения и одновременных вычислений в течение 48 часов. При гораздо меньших затратах облачный сервис может предоставить ту же вычислительную мощность, что и локальный кластер, в течение более короткого периода времени при реагировании на события пыльной бури. Эти результаты сравнения затрат показывают, что использование облачных вычислений является экономичным, поддерживая низкий доступ и прогнозирование разрешения, одновременно задействовав большие вычислительные мощности для выполнения прогнозирования с высоким разрешением для больших географических областей, когда это необходимо.

## 10.4 ЗАКЛЮЧЕНИЕ

Пыльные бури подвержены межгодовой изменчивости и являются типичными разрушительными событиями. Вычислительный пул для размещения оперативной системы прогнозирования пыльных бурь должен также поддерживать прерывистость работы за счет (1) масштабирования для ответа на потенциальные запросы высокого разрешения и одновременных вычислений, когда происходят пыльные бури, и (2) масштабирование, когда не происходят пыльные бури для экономии затрат и энергии. Облачные вычисления используются для поддержки прогнозов пыльных бурь:

- Возможность предоставить большой объем вычислительной мощности за несколько минут для удовлетворения разрушительных вычислительных требований для прогнозирования пыльных бурь (раздел 10.3.2).
- Экономически поддерживая низкие скорости доступа и модели с низким разрешением, при этом сохраняя возможность задействовать большой объем вычислительной мощности для выполнения прогнозирования с высоким разрешением для большого публичного доступа, когда это необходимо (раздел 10.3.3).

Более сложные конфигурации, такие как улучшенная конструкция файловой системы для снижения накладных расходов на ввод-вывод (Huang et al. 2013a), могут лучше поддерживать крупномасштабное моделирование, такое как прогнозирование пыльных бурь. В кластерной вычислительной архитектуре каждый вычислительный узел обычно предназначен для доступа к одному и тому же удаленному хранилищу данных для параллельного выполнения задач. В этом примере NFS используется для совместного использования хранилища и обеспечения синхронизации доступа к данным. Однако разные файловые системы могут иметь большое влияние на производительность операций ввода-вывода и приложений с интенсивным использованием данных. Таким образом, потребители облака могут изучить другие решения, такие как Параллельная виртуальная файловая система версии 2 (PVFS2) (Latham et al. 2010).

## 10.5 КРАТКОЕ ИЗЛОЖЕНИЕ

В настоящей главе обсуждается, как облачные сервисы могут поддерживать прогнозирование разрушительных событий, таких как пыльные бури, на примере Amazon EC2. В разделе 10.1 объясняются предпосылки и мотивация проведения прогнозирования пыльных бурь, а также представлены основные проблемы при прогнозировании пыльных бурь. Подробности настройки и развертывания модели пыльной бури в облачном сервисе EC2 описаны в Разделе 10.2. В разделе 10.3 показано, как облачный сервис может обеспечить прогнозирование пыльной бури на примере пыльной бури в г.Финикс 22 декабря 2009 г. В разделе 10.4 завершаются усовершенствования за счет использования облачных вычислений для поддержки такого крупномасштабного моделирования и прогнозирования, а также будущих направлений исследований.

## 10.6 ПРОБЛЕМЫ

1. Каковы вычислительные сложности для прогнозирования пыльных бурь?
2. Каковы общие шаги по развертыванию моделирования пыльной бури в облаке?
3. Какой тип экземпляров лучше подходит для прогнозирования пыльной бури; обычный экземпляр или экземпляр HPC? Почему?
4. Как настроить кластер виртуальных высокопроизводительных вычислений (HPC) для поддержки приложений синтенсивными вычислениями?
5. Как сервис Elastic Block Storage (EBS) используется для поддержки развертывания модели пыльной бури в облаке?
6. Как создать группу размещения для экземпляров HPC с помощью инструментов управления Amazon Web Console и командной строки? Для чего необходим данный шаг?
7. По сравнению с рабочим процессом развертывания общих приложений, представленным в главе 5, каковы особенности модели пыльной бури?
8. Почему облачные вычисления могут обеспечить экономическую эффективность?
9. Почему облачные вычисления являются хорошим решением для поддержки имитации разрушительного события (например, пыльной бури)?

## СПИСОК ЛИТЕРАТУРЫ

- Baillie, C. F., A. E. MacDonald, and S. Sun, 1995. QNH: A portable, massively parallel multi-scale meteorological model. *Proceedings of the 4th Int'l Conference on the Applications of High Performance Computers in Engineering*, June 19–21. Milan, Italy.
- Goudie, A. S. and N. J. Middleton. 1992. The changing frequency of dust storms through time. *Climatic Change* 20, no. 3: 197–225.

- Drake, J. and I. Foster. 1995. Introduction to the special issue on parallel computing in climate and weather modeling. *Parallel Computing* 21, no. 10: 1539–1544.
- Huang, Q., C. Yang, K. Benedict, A. Rezgui, J. Xie, J. Xia, and S. Chen. 2013a. Using adaptively coupled models and high-performance computing for enabling the computability of dust storm forecasting. *International Journal of Geographic Information Science* 27, no. 4: 765–784.
- Huang, Q., C. Yang, K. Benedict, S. Chen, A. Rezgui, and J. Xie. 2013b. Enabling dust storm forecasting using cloud computing. *International Journal of Digital Earth* 6, 4: 338–355.
- Latham, R. et al. 2010. A next-generation parallel file system for Linux cluster. *Linux World Magazine* 2, no. 1: 54–57.
- NOAA. 2011. Dust Storm Database [online]. [http://www4.ncdc.noaa.gov/cgi-win/wwcgi.dll?wwevent\\_storms](http://www4.ncdc.noaa.gov/cgi-win/wwcgi.dll?wwevent_storms) (accessed October 30, 2011 and 2013).
- Rehr, J. J., F. D. Vila, J. P. Gardner, L. Svec, and M. Prange. 2010. Scientific computing in the cloud. *Computing in Science & Engineering* 12, no. 3: 34–43.
- Shoemaker, C. and J. Davis. 2008. Hazardous Weather Climatology for Arizona, NOAA Technical Memorandum. 2008. <http://www.wrh.noaa.gov/wrh/tech-Memos/TM-282.pdf> (accessed March 10, 2013).
- WMO (World Meteorological Organization). 2012. WMO Sand and Dust Storm Warning Advisory and Assessment System (SDS-WAS): Science and Implementation Plan 2011–2015. [http://www.wmo.int/pages/prog/arep/wwrp/new/documents/SDS\\_WAS\\_implementation\\_plan\\_01052012.pdf](http://www.wmo.int/pages/prog/arep/wwrp/new/documents/SDS_WAS_implementation_plan_01052012.pdf) (accessed March 30, 2013).
- Wolters, L., G. Cats, and N. Gustafsson. 1995. Data-parallel numerical weather forecasting. *Science Program* 4, no. 3: 141–153.
- Xie, J., C. Yang, B. Zhou, and Q. Huang. 2010. High-performance computing for the simulation of dust storms. *Computers, Environment and Urban Systems* 34, no. 4: 278–290.
- Yang, C., H. Wu, Q. Huang, Z. Li, and J. Li. 2011. Using spatial principles to optimize distributed computing for enabling the physical science discoveries. *Proceedings of the National Academy of Sciences* 108, no. 14: 5498–5503.

## Часть IV

---

# Состояние и готовность облачной обработки данных

---

В данной части используются пять глав для систематического изучения готовности как коммерческих облачных сервисов, так и решений облачной обработки данных с открытым исходным кодом к поддержке геолого-геофизических приложений. В главе 11 представлены три облачных сервиса: Amazon EC2, Windows Azure и NASA Nebula. В главе 12 показаны методы и инструменты для проверки готовности трех облачных сервисов. В главе 13 рассматриваются четыре основных решения для облачной обработки данных с открытым исходным кодом. В главе 14 подробно рассматриваются методы и инструменты для тестирования вычислительной производительности решений. Наконец, в главе 15 сообщается о результатах GeoCloud, межведомственного проекта, возглавляемого Федеральным комитетом по географическим данным (FGDC), по изучению способов развертывания геонаучных приложений в облаке, а также о преимуществах и недостатках размещения приложений в облаке.



## Глава 11

---

# Облачные сервисы

*Чэнь Сю, Цзичжэ Ся, Цюньин Хуан и Мира Бамбакус*

---

В данной главе описаны три облачных сервиса «Инфраструктура как услуга» (IaaS): Amazon EC2, Windows Azure и NASA Nebula. Поставщики IaaS предлагают виртуальные машины (VM) пользователям облака для размещения их данных и приложений. Несколько критических факторов могут повлиять на способность поставщиков облачных сервисов эффективно предлагать услуги своим клиентам, такие как географическое присутствие, пользовательский интерфейс, автоматическое масштабирование и балансировка нагрузки, а также соглашение об уровне обслуживания (SLA).

### 11.1 ВВЕДЕНИЕ В ОБЛАЧНЫЕ СЕРВИСЫ

#### 11.1.1 Географическое присутствие

Промышленные компании являются новаторами в области реализации парадигмы сервисных вычислений. От раннего режима разделения времени до недавнего режима облачной обработки данных на основе виртуализации коммерческие компании внедрили передовые технологии совместного использования вычислений. Хотя облачный сервис все еще находится на ранней стадии своего жизненного цикла, как поставщики, так и потребители облачных услуг уверены в своем потенциале. Например, облачная обработка данных используется, чтобы помочь организациям выйти на более распределенные рынки, сократить инвестиции в оборудование и стать более гибкими для получения конкурентного преимущества (ISACA 2012).

Виртуализация позволяет географически распределить вычислительные мощности (глава 3) для пользователей облака, чтобы они могли совместно размещать свои услуги с конечными пользователями. У крупных поставщиков облачных сервисов обычно есть центры обслуживания в нескольких географических точках. Например, Amazon и Microsoft имеют несколько центров обработки данных по всему миру. Таким образом, пользователи облака будут иметь лучшую непрерывность обслуживания, например, за счет планирования более качественных услуг вычислений и хранения данных ближе к конечным пользователям или за счет быстрого перемещения услуг в другой центр в другом географическом месте, когда один центр обработки данных отключен. Ли и др. (2010) и Ян и др. (2011) демонстрируют, что географическое присутствие облачных провайдеров становится актуальным для выбора провайдеров услуг из-за пространственной близости между конечными пользователями и центром обработки данных.

## **11.1.2 Пользовательские интерфейсы и доступ к серверам**

Облачная обработка данных чрезвычайно сложна с технической точки зрения. Однако провайдеры облачных сервисов обычно сдерживают трудности, чтобы потребители облачных сервисов не имели дела с техническими деталями. Основная стратегия заключается в том, чтобы предоставить потребителям облака доступ к облачным сервисам через хорошо продуманные пользовательские интерфейсы, которые должны быть интуитивно понятными, независимо от сложности базовой технологии. Потребители облака также должны иметь возможность использовать облачные сервисы с помощью четко определенных процедур. Это достигается за счет использования единого представления управления, которое состоит из инструментов, помогающих клиентам пройти весь жизненный цикл использования облачной обработки данных от конфигурации и предоставления до управления и мониторинга (Scheier 2012). Такой подход потенциально делает облачные вычисления более удобными для пользователя и позволяет системным администраторам управлять большим пулом ресурсов (Solnik 2012).

## **11.1.3 Автоматическое масштабирование и балансировка нагрузки**

Автоматическое масштабирование позволяет приобретать или высвобождать вычислительные ресурсы в соответствии с вычислительными потребностями, как это предопределено конфигурациями масштабирования уровня обслуживания. Динамическое масштабирование обеспечивает механизм для поддержания производительности системы при изменении требований конечных пользователей (Bondi 2000). Средство балансировки нагрузки распределяет задачи приложения по доступным вычислительным ресурсам для достижения целей производительности. Балансировка нагрузки может оптимизировать использование доступных ресурсов, обеспечить лучшую вычислительную мощность и повысить общую надежность системы. Балансировщик нагрузки, программный или аппаратный, отвечает за предоставление каждому входящему запросу соответствующего вычислительного ресурса. Различные поставщики коммерческих облачных сервисов могут поддерживать разные типы сервисов масштабирования и балансировки нагрузки. В результате заказчики облачных вычислений должны тщательно оценивать решения от разных поставщиков, чтобы соответствовать своим вычислительным требованиям и надлежащим облачным сервисам.

## **11.1.4 Соглашение о гарантированном уровне обслуживания (SLA)**

SLA формально определяет контракт на обслуживание между поставщиком облачных сервисов и заказчиком облачных сервисов. Типичный SLA состоит из разделов, в которых четко изложены различные аспекты услуги, например, что такое услуга, как будет оцениваться производительность услуги, как спланировать антикризисное управление и как прекратить предоставление услуги. Обычно Соглашение о гарантированном уровне обслуживания обсуждается между поставщиками и клиентами, что делает его сложной, но важной частью контракта на облачную обработку данных.

В следующем разделе представлены три основных облачных сервиса. Подробная информация о трех услугах дана для четырех соображений, когда пользователь облака выбирает надлежащую услугу.

## 11.2 amazon WeB serVICes (aWs)

### 11.2.1 Архитектура

Amazon - один из крупнейших поставщиков облачных сервисов. Одна из услуг Amazon - Amazon EC2. AWS предлагает множество сервисов для реализации различных облачных возможностей (рисунок 11.1). В архитектуре AWS есть шесть уровней:

- *Уровень физической инфраструктуры.* Amazon неуклонно расширяет свою глобальную инфраструктуру, чтобы помочь своим клиентам добиться меньших задержек и более высокой пропускной способности. В настоящее время у Amazon есть центры обработки данных, расположенные в девяти регионах (рис. 11.2). AWS GovCloud (США) - это AWS

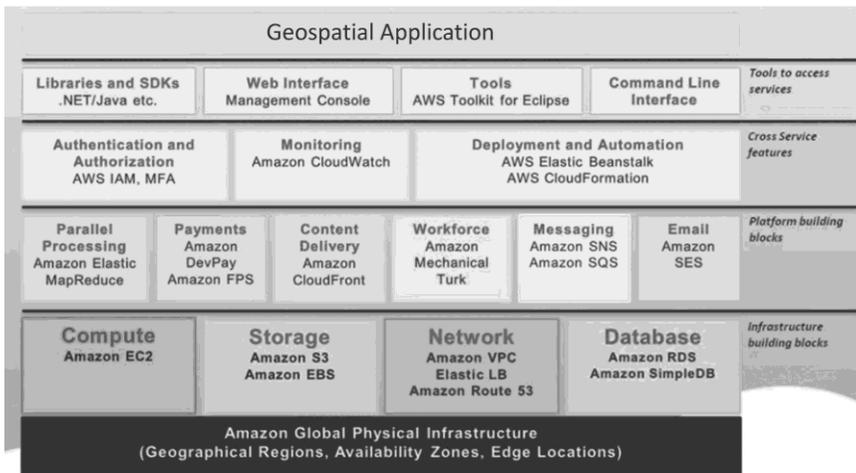


Рисунок 11.1 Архитектура AWS General Service. (Отредактировано по материалам Varia, J., 2011 г., <http://www.slideshare.net/AmazonWebServices/amazon-ec2-and-aws-elasticbeanstalk-introduction>, по состоянию на 28 апреля 2013 г.)

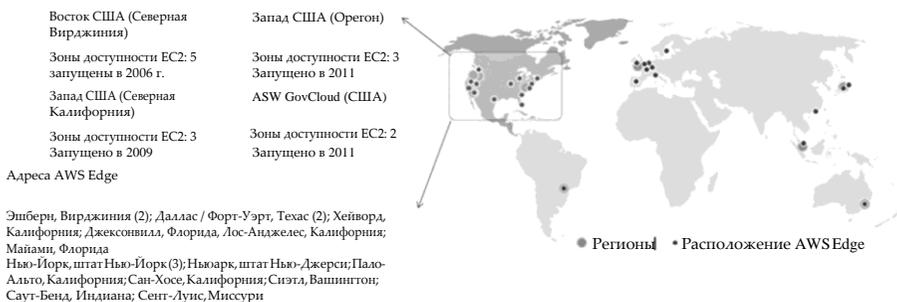


Рисунок 11.2 Глобальная инфраструктура Amazon. (Из Amazon AWS по адресу <http://aws.amazon.com/about-aws/globalinfrastructure/#reglink-na>.)

Регион разработан, чтобы позволить правительственным учреждениям и клиентам США переносить более чувствительные рабочие нагрузки в облако, соблюдая конкретные нормативные требования. Кроме того, сервисы Amazon CloudFront и Amazon Route 53 предлагаются в точках AWS Edge для доставки контента конечным пользователям с меньшими задержками (рисунок 11.2).

- *Уровень строительных блоков инфраструктуры* - уровень предоставляет различные типы IaaS, чтобы позволить клиентам облачной обработки данных получить доступ к базовым вычислительным ресурсам (например, вычислительной мощности, хранилищу и сети). Например, вычислительные мощности по запросу могут быть предоставлены сервисами Amazon EC2; неограниченное хранилище может быть предложено сервисами Amazon S3 и EBS, а сетевые функции предоставляются Amazon VPC, Elastic LB и Amazon Route 53.
- *Уровень строительных блоков платформы* - уровень предоставляет другую платформу как услуги (PaaS). Например, CloudFront<sup>1</sup> может помочь пользователям облака настраивать и доставлять контент в 21 периферийное местоположение по всему миру (рисунок 11.2); Сервисы обмена сообщениями позволяют потребителям облака легко и экономично программировать приложения с различными языками и API (например, Java, PHP, Python, обрабатывать огромное количество Ruby и .NET); Amazon Simple E-mail Service (SES) может быстро и экономично отправлять массовые и транзакционные электронные письма.
- *Межсервисный уровень* - уровень реализует расширенные облачные сервисы для управления, такие как безопасность, мониторинг и автоматизация.
- *Уровень инструментов и API*. На данном уровне доступны различные инструменты и API для доступа к базовому AWS.
- *Уровень приложений* - приложения можно переносить и развертывать на облачной платформе Amazon и настраивать для интеграции различных AWS для наилучшего обслуживания пользователей облака.

## 11.22 Общие характеристики eC2

### 11.22.1 Масштабируемость

EC2 предоставляет услугу автоматического масштабирования.<sup>2</sup> Автоматическое масштабирование позволяет потребителям увеличивать или уменьшать вычислительную мощность EC2 автоматически в соответствии с предопределенными условиями, такими как время отклика и загрузка ЦП. С помощью службы автоматического масштабирования потребители могут гарантировать, что количество используемых экземпляров EC2 плавно увеличивается во время пиков спроса для поддержания производительности и автоматически уменьшается, когда спрос падает, чтобы минимизировать затраты. Автоматическое масштабирование особенно хорошо подходит для приложений, которые испытывают временную изменчивость в использовании.

<sup>1</sup> См. AWS CloudFront на <http://aws.amazon.com/cloudfront/>.

<sup>2</sup> См. AWS Auto Scaling на <http://aws.amazon.com/autoscaling/>.

Кроме того, на межсервисном уровне, показанном на рисунке 11.1, есть две расширенные службы: Elastic Beanstalk и CloudFront, которые можно легко использовать для реализации масштабируемости

- Amazon CloudFront - это веб-сервис для доставки контента<sup>1</sup>. Он интегрируется с другими AWS, чтобы предоставить потребителям облака простой способ распространять контент среди конечных пользователей с низкой задержкой и высокой скоростью передачи данных.
- ElasticBeanstalk<sup>2</sup> дает клиентам облака еще более простой способ быстро развертывать и управлять приложениями в облаке AWS, не отказываясь от контроля над базовыми ресурсами AWS. AWS Elastic Beanstalk автоматически обрабатывает детали выделения емкости, балансировки нагрузки, автоматического масштабирования и мониторинга работоспособности приложений.

### **11.2.2 Совместимость**

С тех пор, как Amazon EC2 был пионером на рынке облачных услуг, его реализации обычно считались де-факто стандартами для различных облачных услуг. Следовательно, многие новые облачные провайдеры следуют интерфейсу Amazon EC2.

### **11.2.3 Развертывание и интерфейс**

AWS предоставляет множество инструментов и API для доступа к облачным сервисам и управления ими. Консоль управления AWS (рис. 11.3) - это простой и интуитивно понятный пользовательский веб-интерфейс, позволяющий потребителям получать доступ к AWS и управлять им. Amazon также предоставляет сопутствующее мобильное приложение для систем Android и iOS, позволяющее быстро просматривать ресурсы на лету.

### **11.2.4 Гипервизоры**

EC2 - это паравиртуализированная среда, основанная на Xen.3 Это одна из крупнейших инсталляций Xen, использующая сильно модифицированную и адаптированную версию Xen.

### **11.2.5 Надежность**

SLA Amazon EC2 гарантирует доступность 99,95% для всех регионов Amazon EC2. Все остальные сервисы AWS, такие как CloudFront, также предоставляют SLA с обязательством уровня обслуживания 99,9% доступности в целом. Например, Amazon EC2 предлагает высоконадежную среду для геопространственных приложений (подробности см. В главах 9 и 10), поскольку сервис работает в проверенной сетевой инфраструктуре и центрах обработки данных Amazon.

<sup>1</sup> См. AWS CloudFront на <http://aws.amazon.com/cloudfront/>.

<sup>2</sup> См. AWS Elastic Beanstalk beta на <http://aws.amazon.com/elasticbeanstalk/>.

<sup>3</sup> См. On Amazon EC2's Underlying Architecture на [http://openfoo.org/blog/amazon\\_ec2\\_underlying\\_architecture.html](http://openfoo.org/blog/amazon_ec2_underlying_architecture.html).

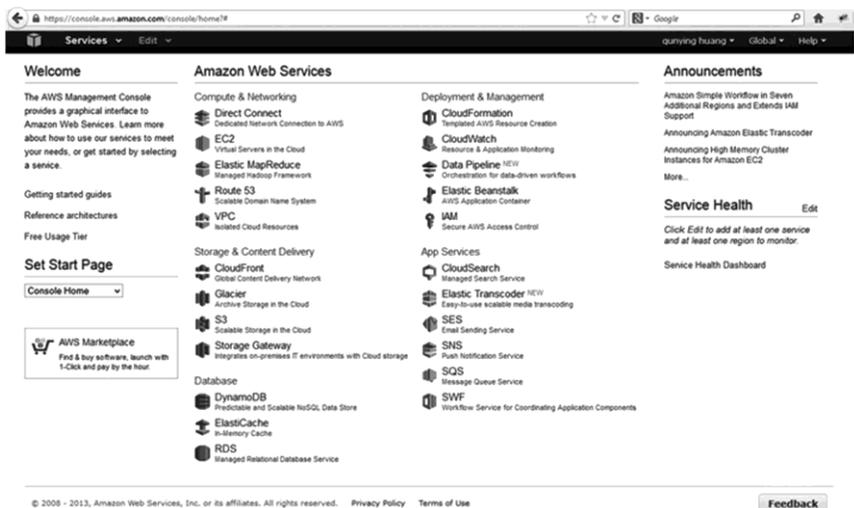


Рисунок 11.3 Консоль управления AWS.

### 11.2.2.6 Поддержка ОС

EC2 поддерживает операционные системы Windows и Linux, работающие на виртуальных машинах.

### 11.2.2.7 Стоимость

EC2 имеет гибкие и разнообразные ценовые модели, которые позволяют пользователям облака снижать затраты в зависимости от рабочих нагрузок. Стоимость рассчитывается на основе таких факторов, как модель клиента, регионы и использование вычислений, тип экземпляра и операционная система экземпляров. Модель арендатора включает в себя

(1) По запросу, (2) зарезервировано (Light, Medium, и Heavy на 1 год, 3 года) и (3) Спот.

- Модель экземпляров по требованию позволяет потребителям оплачивать вычислительные мощности почасово без долгосрочных обязательств. На рис. 11.4 показана стоимость экземпляров по запросу для Восточного региона США (раздел 11.2.1).
- Модель зарезервированных экземпляров дает потребителям возможность вносить небольшую единовременную оплату за каждый зарезервированный экземпляр и, в свою очередь, получать значительную скидку на почасовую оплату за этот экземпляр.
- Модель спотовых экземпляров позволяет потребителям делать ставки на вычислительные мощности Amazon EC2. Потребители облака просто используют запасные экземпляры Amazon EC2 и запускают их всякий раз, когда ставка превышает текущую спотовую цену, которая изменяется в реальном времени в зависимости от спроса и предложения.

Region: US East (N. Virginia) ▼		
	Linux/UNIX Usage	Windows Usage
<b>Standard On-Demand Instances</b>		
Small (Default)	\$0.065 per Hour	\$0.115 per Hour
Medium	\$0.130 per Hour	\$0.230 per Hour
Large	\$0.260 per Hour	\$0.460 per Hour
Extra Large	\$0.520 per Hour	\$0.920 per Hour
<b>Second Generation Standard On-Demand Instances</b>		
Extra Large	\$0.580 per Hour	\$0.980 per Hour
Double Extra Large	\$1.160 per Hour	\$1.960 per Hour
<b>Micro On-Demand Instances</b>		
Micro	\$0.020 per Hour	\$0.020 per Hour
<b>High-Memory On-Demand Instances</b>		
Extra Large	\$0.450 per Hour	\$0.570 per Hour
Double Extra Large	\$0.900 per Hour	\$1.140 per Hour
Quadruple Extra Large	\$1.800 per Hour	\$2.280 per Hour
<b>High-CPU On-Demand Instances</b>		
Medium	\$0.165 per Hour	\$0.285 per Hour
Extra Large	\$0.660 per Hour	\$1.140 per Hour
<b>Cluster Compute Instances</b>		
Quadruple Extra Large	\$1.300 per Hour	\$1.610 per Hour
Eight Extra Large	\$2.400 per Hour	\$2.970 per Hour
<b>High-Memory Cluster On-Demand Instances</b>		
Eight Extra Large	\$3.500 per Hour	\$3.831 per Hour
<b>Cluster GPU Instances</b>		
Quadruple Extra Large	\$2.100 per Hour	\$2.600 per Hour
<b>High-I/O On-Demand Instances</b>		
Quadruple Extra Large	\$3.100 per Hour	\$3.580 per Hour
<b>High-Storage On-Demand Instances</b>		
Eight Extra Large	\$4.600 per Hour	\$4.931 per Hour

Рисунок 11.4 Стоимость экземпляров по требованию для различных экземпляров ОС в восточном регионе США в феврале 2013 г. (Стоимость облачных услуг часто меняется).

Модель ценообразования на спотовые экземпляры дополняет модели ценообразования на экземпляры по требованию и зарезервированные экземпляры, обеспечивая потенциально наиболее экономичный вариант получения вычислительной мощности.

## 11.23 Основные пользователи и общие комментарии

### 11.23.1 Список основных клиентов

По данным 451 Group (ведущая синдицированная исследовательская, консультационная и профессиональная компания), Amazon заняла примерно 59% рынка облачных услуг, в то время как все другие поставщики облачных услуг разделили остальную часть 41% потребителей облачных услуг в 2011 году.



Рисунок 11.5 Основные клиенты Amazon EC2. (См. Amazon AWS по адресу <http://www.slideshare.net/AmazonWebServices/the-total-cost-of-non-ownership-in-the-cloud>.)

<sup>1</sup> Amazon EC2 в настоящее время обслуживает многие популярные интернет-сайты, которым доверяют многие предприятия и правительственные учреждения (например, NASA и IBM), и имеет множество клиентов из Ассоциации государств Юго-Восточной Азии (ASEAN) (рис. 11.5). Успешные приложения и предприятия, поддерживаемые EC2, можно найти на веб-сайте.<sup>2</sup>

### 11.2.3.2 Завершение<sup>3</sup>

AWS, запущенная в июле 2002 года, предоставляет онлайн-сервисы для веб-сайтов или клиентских приложений. В 2004 году Amazon предвидела, что будет выгодно предлагать своим клиентам услуги на уровне инфраструктуры. В результате Amazon S3 был запущен в марте 2006 года, а Amazon EC2 был построен в августе 2006 года, а инфраструктура Amazon и база разработчиков доступны по всему миру. Предлагая услуги в 2006 году и с тех пор, AWS стала лидером на рынке облачных вычислений благодаря раннему выходу на рынок, быстрым инновациям и гибким облачным сервисам. В июне 2007 года Amazon заявила, что более 330 000 разработчиков подписались на использование AWS. Как основная часть AWS, EC2 предоставляет вычислительные возможности для организаций и может поддерживать множество приложений (рисунок 11.5) (глава 5). В ноябре 2010 года Amazon переключила свой флагманский розничный веб-сайт на EC2 и AWS.

### 11.2.3.3 Обратная связь с общественностью

EC2 предлагает разнообразные облачные сервисы и имеет самую большую долю рынка облачных сервисов. Наибольшую озабоченность EC2 вызывает потенциальное отключение электричества, которое может вывести из строя весь региональный сервисный центр.

<sup>1</sup> См. Alog на <http://www.alog.com.br/en/noticias/infraestrutura-elastica-e-atraente-a-orcamento/>.

<sup>2</sup> См. AWS на <http://aws.amazon.com/solutions/case-studies/>.

<sup>3</sup> См. Amazon.com Basic Case Study на <http://www.slideshare.net/davinken/amazon-Web-services-history-overview-presentation>.

- 14 июня 2012 года центр обработки данных Amazon U.S. East, крупнейший центр обработки данных, был отключен от электроэнергии из-за суровых погодных условий. Heroku, Quora, Parse и Pinterest были среди сайтов, пострадавших от сбоя, наряду со многими небольшими компаниями, которые полагаются на Amazon как на источник вычислительной мощности вместо традиционного центра обработки данных.<sup>1</sup>
- 29 июня 2012 года в AWS вновь произошла сбой из-за сильных электрических бурь, в результате которых был причинен дискомфорт многим пользователям веб-сайтов.
- 22 октября 2012 года облачная инфраструктура Amazon EC2 на востоке США (центр обработки данных в Северной Вирджинии) вышла из строя, что повлияло на работу многих веб-сервисов, включая Reddit, Airbnb, Flipboard, GetGlue, Coursera и др.<sup>2</sup>
- 24 декабря 2012 года отключение EC2 сделало миллионы клиентов Netflix неспособными транслировать видео в ночь с высоким спросом для просмотра фильмов.

Такие сбои существенно влияют на веру клиентов в то, что они полагаются исключительно на технологии облачной обработки данных, и привели к тому, что некоторые из них покинули "облачные" сервисы (например, Whatsyourprice). Но сторонники Amazon говорят, что инциденты просто демонстрируют необходимость использования нескольких зон доступности.<sup>3</sup>

## 11.24 Сложность использования

AWS предлагает удобные веб-инструменты для легкого доступа и управления облачными сервисами (рис. 11.1). На рисунке 11.3 показаны сервисы AWS, доступные через веб-консоль. AWS также предоставляет SDK, IDE Toolkits и Command Line Tools для разработки и управления потребительскими приложениями.<sup>4</sup> С помощью SDK услуги AWS могут быть интегрированы в разработку приложений с использованием различных языков программирования (например, Java, Python, PHP, Ruby или .Net) или платформ (например, Android или IOS). Разработка AWS также может быть ускорена с помощью специализированных облачных инструментов IDE, интегрированных в среду разработки (например, Eclipse и Visual Studio). Кроме того, пользователи облачных сервисов могут управлять сервисами AWS из командной строки и автоматизировать управление сервисами с помощью скриптов.

<sup>1</sup> См. Information Week на <http://www.informationweek.com/cloud-computing/infrastructure/amazon-defended-after-june-14-cloud-outage/240002207>.

<sup>2</sup> См. Venture Beat на <http://venturebeat.com/2012/10/22/amazon-cloud-outage-takes-down-reddit-airbnb-flipboard-more/>.

<sup>3</sup> См. Information Week на <http://www.informationweek.com/cloud-computing/infrastructure/amazon-defended-after-june-14-cloud-outage/240002207>.

<sup>4</sup> См. AWS Tools на <http://aws.amazon.com/tools/>.

## 11.3 WinDoWs azure

### 11.3.1 Архитектура

Windows Azure - один из самых популярных облачных сервисов, предоставляемый Microsoft. Windows Azure предоставляет как PaaS, так и IaaS. На рис. 11.6 показана архитектура и поддерживаемые функции Windows Azure, в которую входят вычисления, службы данных, сети, службы приложений, коммерция, локальная среда разработки и портал управления, обеспечивающий доступ ко всем компонентам и управление ими.

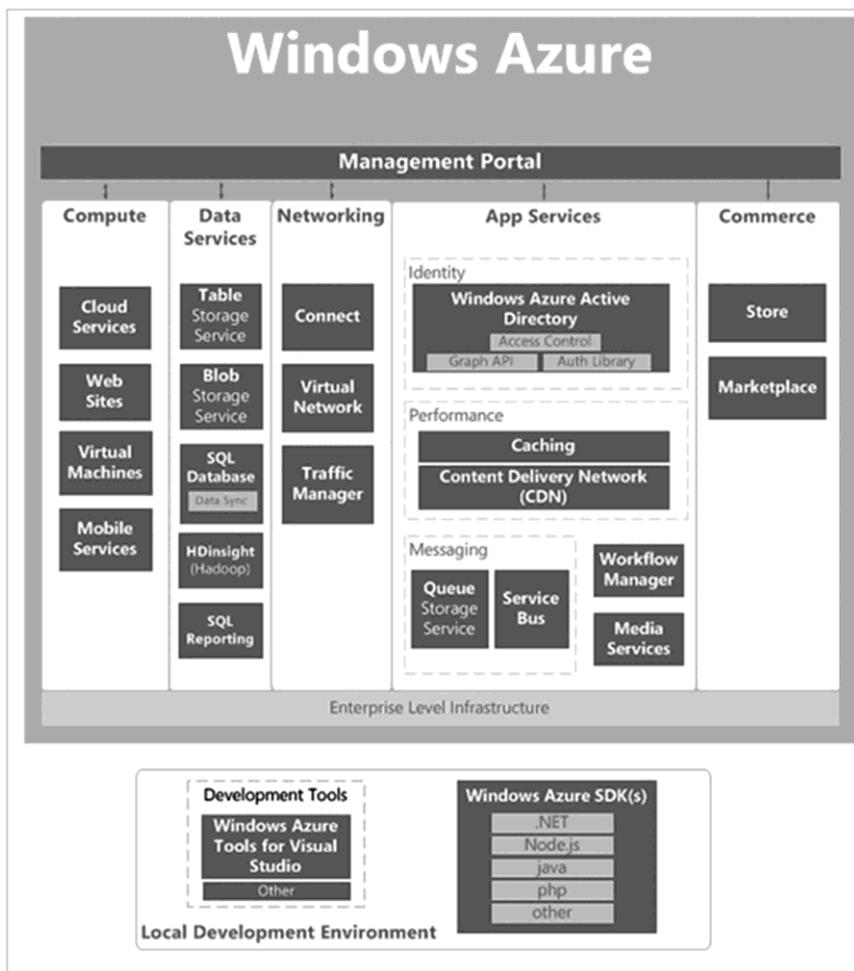


Рисунок 11.6 Архитектура и особенности Windows Azure. (См. Windows Azure по адресу <http://msdn.microsoft.com/en-us/library/windowsazure>.)

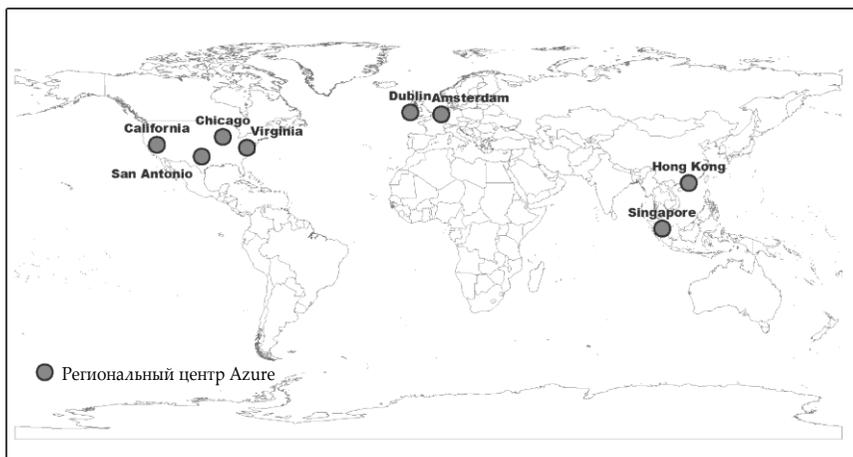


Рисунок 11.7 Глобальная инфраструктура Azure.

Windows Azure предоставляет вычислительные возможности через четыре вычислительных компонента: веб-сайты, облачные сервисы, виртуальные машины и мобильные службы. В настоящее время центры обработки данных расположены в восьми регионах (рис. 11.7), включая Восток США (Вирджиния), Запад США (Калифорния), Северный центральный регион США (Иллинойс), Южный Центральный регион США (Техас), Север ЕС (Ирландия), ЕС. Запад (Нидерланды), Восточная Азия (Гонконг) и Юго-Восточная Азия (Сингапур).

Windows Azure имеет независимую службу данных, которая предоставляет возможности для хранения, управления и составления отчетов. Службы данных включают в себя таблицу Windows Azure, BLOB-объекты и соответствующую базу данных SQL Azure. Сеть Windows Azure обеспечивает общие связи между различными компонентами. Службы приложений Windows Azure предоставляют различные модули, обеспечивающие идентификацию, производительность и обмен сообщениями в облачной платформе. Инструменты разработки и Windows Azure SDK обеспечивают локальную среду разработки для разработчиков.

## 11.3.2 Общие характеристики Windows azure

### 11.3.2.1 Масштабируемость

В Windows Azure вычислительные ресурсы можно отделить от хранилища для достижения независимой масштабируемости вычислений и хранилища. Данный механизм также обеспечивает изоляцию и мультиарендность. В 2012 году команда Windows Azure создала новую плоскую сетевую систему, которая могла бы создать плоское сетевое хранилище (FNS), охватывающее все центры обработки данных Windows Azure.

Таблица 11.1 Новая масштабируемая сетевая система (поколение 2) по сравнению со старой сетевой системой (поколение 1)

Хранение SKU	Узел хранения Скорость сети	Сеть между вычислениями и хранилищем	Балансировщик нагрузки	Устройство хранения, используемое для ведения журнала
Пок. 1	1 Гбит / с	Иерархическая сеть	Аппаратная нагрузка Балансировщик	Жесткие диски
Пок. 2	10 Гбит / с	Плоская сеть	Нагрузка ПО Балансировщик	Твердое состояние Диски (SSD)

Система FNS привела к увеличению пропускной способности сетевого подключения для поддержки виртуальных машин Windows Azure. Новая сетевая конструкция лучше поддерживает приложения для высокопроизводительных вычислений (HPC), которые требуют обширных коммуникаций и значительной полосы пропускания между вычислительными узлами. В таблице 11.1<sup>1</sup> показаны улучшения новой сетевой системы (Поколение 2) по сравнению с исходной сетевой системой (Поколение 1).

Текущее хранилище Azure может обеспечить следующую масштабируемость для одной учетной записи хранения, созданной после 7 июня 2012 г.

- Емкость - до 200 терабайт (ТБ)
- Транзакции - до 20 000 сущностей / сообщений / BLOB-объектов в секунду.
- Пропускная способность для геоизбыточной учетной записи хранения
- На входе (глава 6) - до 5 гигабит в секунду.
- На выходе (Глава 6) - до 10 гигабит в секунду.
- Пропускная способность для локальной избыточной учетной записи хранения
- На входе - до 10 гигабит в секунду.
- Выходной сигнал - до 15 гигабит в секунду.

### 11.3.2 Совместимость

Некоторые продукты Microsoft были интегрированы с Windows Azure, чтобы помочь пользователям облака лучше использовать облачные сервисы и управлять ими. Например, пользователи могут использовать Microsoft SQL Server<sup>2</sup> для доступа и работы (например, запроса, обновления, удаления таблиц) своей базы данных SQL Azure. Microsoft WebMatrix<sup>3</sup> и Visual Studio<sup>4</sup> можно использовать для удаленного доступа, обновления, развертывания и настройки приложений в Windows Azure.

<sup>1</sup> См. MSDN Blogs на <http://blogs.msdn.com/b/windowsazure/archive/2012/11/02/windows-azure-s-flat-network-storage-and-2012-scalability-targets.aspx>.

<sup>2</sup> См. Microsoft SQL Server на <http://www.microsoft.com/en-us/sqlserver/>.

<sup>3</sup> См. Microsoft WebMatrix на <http://www.microsoft.com/web/webmatrix/>.

<sup>4</sup> См. Visual Studio на <http://www.microsoft.com/visualstudio/>.

### **11.32.3 Развертывание и интерфейс**

Интерфейс Windows Azure включает в себя портал управления и мощные инструменты командной строки. Портал управления и инструменты командной строки просты в использовании и помогают потребителям и конечным пользователям облака управлять приложениями и облачными сервисами. Кроме того, несколько продуктов Microsoft (например, Microsoft Visual Studio и Microsoft WebMatrix) могут помочь в развертывании пользовательских приложений и облачных ресурсов и управлении ими. Windows Azure также имеет API службы на основе REST для управления и развертывания.

### **11.32.4 Гипервизоры**

Гипервизор Windows Azure разработан специально. Данный гипервизор оптимизирован на основе конкретной однородной среды центра обработки данных в Microsoft. В настоящее время Microsoft не предоставляет гипервизор Windows Azure как единый продукт для пользователей облака и конечных пользователей.

### **11.32.5 Надежность**

Windows Azure гарантирует надежность вычислений 99,95% и надежность экземпляра роли и хранилища 99,9%.<sup>1</sup> Сервис предоставляет несколько механизмов для обеспечения надежности. Например, он предоставляет механизм очередей, Azure Service Bus Queues, для повышения надежности системы и приложений.

### **11.32.6 Поддержка ОС**

В настоящее время Windows Azure поддерживает операционные системы Windows и Linux для виртуальных машин.

### **11.32.7 Стоимость**

Windows Azure не требует предоплаты. Стоимость рассчитывается на основе использования компьютеров. Стоимость включает в себя: (1) стоимость веб-сайта, (2) стоимость виртуальных машин, (3) стоимость облачных сервисов, (4) стоимость мобильных сервисов и (5) стоимость управления данными. Стоимость каждого компонента зависит от конфигурации облака. Планы оплаты Windows Azure включают в себя следующее: (1) текущие планы, (2) 6-месячные планы и (3) 12-месячные планы. Microsoft предоставляет официальный калькулятор стоимости<sup>2</sup> для Windows Azure, чтобы пользователи могли легко оценить стоимость. На рис. 11.8 показан пример использования калькулятора цен Windows Azure для оценки стоимости веб-сайта. Следует отметить, что цена Windows Azure динамично меняется. Например, 1 февраля 2013 г. стоимость Windows Azure увеличилась на 5%.

<sup>1</sup> См. IT Strategists на <http://www.itstrategists.com/Microsoft-Azure.aspx>.

<sup>2</sup> См. Windows Azure Pricing Calculator на <http://www.windowsazure.com/en-us/pricing/calculator>.

## No upfront costs. Pay only for what you use.



### Web Sites

Free   Shared   **Reserved**

Load balance up to 100 Web Sites across instances dedicated to your apps.  
Includes custom domain names, built-in FTP, Git, TFS, and Web Deploy support.



### SQL Database



### Bandwidth



### Support



Full Calculator

**\$248.85/mo**  
US Dollar (\$)

Рисунок 11.8 Калькулятор цен Windows Azure.

## 11.3.3 Основные пользователи и общие комментарии

### 11.3.3.1 Список основных клиентов

Windows Azure широко используется в различных отраслях и сообществах. Австралийский ритейлер Харви Норман использовал облачный сервис Windows Azure, чтобы снизить риски и капитальные затраты на сайте онлайн-сделок. Hogg Robinson Group расширила свою основную платформу для путешествий на мобильные устройства с помощью сервиса Windows Azure Mobile. Использование Windows Azure позволило сэкономить до 80% начальных затрат и повысило производительность на 25%. Вашингтонский университет переместил Michelangelo (систему самообслуживания) на Windows Azure, чтобы решить проблемы масштабируемости системы. На рисунке 11.9 показана часть основных клиентов Windows Azure.

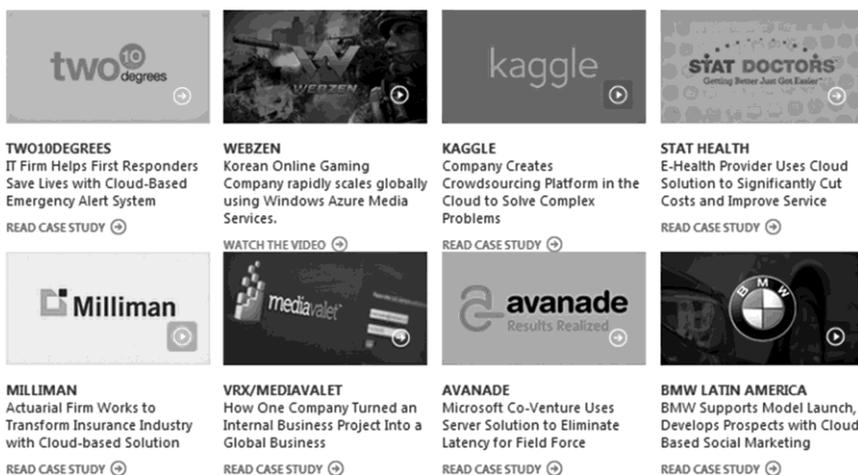


Рисунок 11.9 Основные клиенты Windows Azure. (См. Примеры использования Windows Azure на <http://www.windowsazure.com/en-us/home/case-studies/>.)

### 11.3.3.2 Завершение

Windows Azure была анонсирована в октябре 2008 года, после чего в марте 2009 года было объявлено о выпуске соответствующих данных SQL Azure. С момента своего создания Windows Azure несколько раз обновлялась для предоставления большего количества сервисов и функций. В октябре 2010 года в Windows Azure была добавлена полная поддержка IIS и дополнительные небольшие экземпляры. В декабре 2011 года была добавлена поддержка Traffic Manager, отчетов SQL Azure и планировщика HPC.

### 11.3.3.3 Обратная связь с общественностью

Windows Azure - один из самых популярных коммерческих облачных сервисов. Он предоставляет облачные сервисы как PaaS, так и IaaS. Windows Azure привлекает многих потребителей своей совместимостью со средой .NET. Стратегическое движение Windows Azure к сообществу разработчиков ПО с открытым исходным кодом потенциально увеличивает ее долю на рынке в будущем.

### 11.3.3.4 Сложность использования

Windows Azure предоставляет простой в использовании портал управления и мощные инструменты командной строки для управления приложениями и облачными службами. Сложность использования можно даже снизить, если клиенты будут использовать сервис PaaS Windows Azure, что избавляет их от необходимости развертывать всю вычислительную среду.

## 11.4 neBula

Nebula - это федеральная служба облачной обработки данных, обслуживаемая NASA. Nebula входит в список проектов с открытым исходным кодом. Однако, в отличие от обычных облачных решений с открытым исходным кодом, у Nebula была специальная миссия - поддерживать научные задачи NASA с помощью частного облака. Бета-версия Nebula была выпущена в сентябре 2010 года.

### 11.4.1 Архитектура

На рисунке 11.10 представлены архитектурные компоненты платформы Nebula. IaaS, предоставляемый Nebula, позволяет клиентам облачной обработки данных запускать вычислительные ресурсы в виде полнофункциональных виртуальных машин, подключенных к сети, по запросу. Уникальное требование к облачной обработке данных для поддержки научных задач - иметь дело с огромным количеством научных данных. Nebula предоставляет несколько сервисов хранения, которые можно использовать для хранения и извлечения больших объемов данных в любое время из любого места в Интернете.

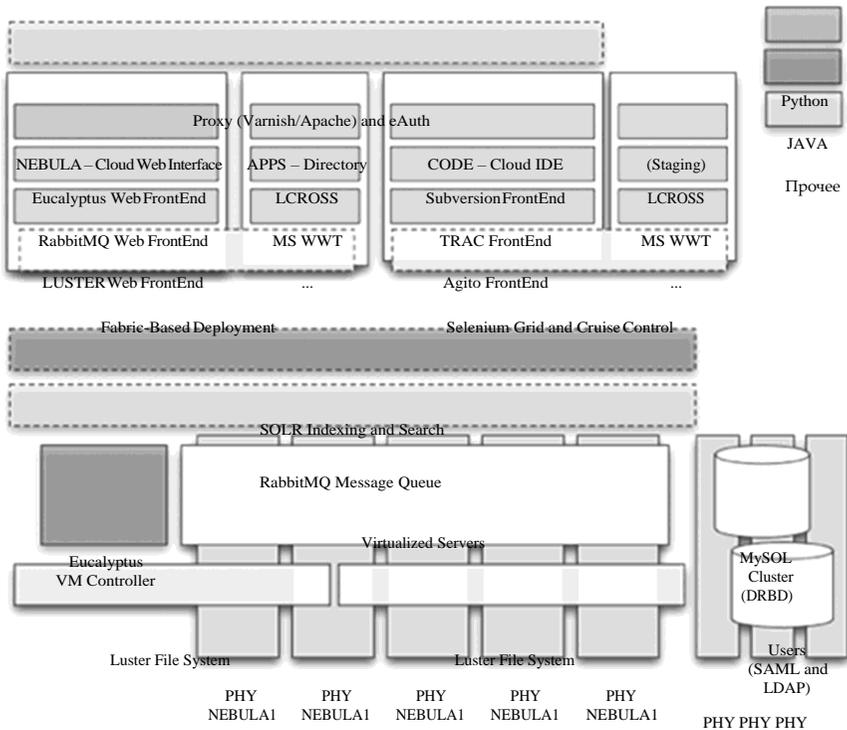


Рисунок 11.10 Архитектурные компоненты Nebula.

## **11.4.2 Общие характеристики nebula**

### **11.4.2.1 Масштабируемость**

Nebula предусматривала поддержку пользователей облака для научных исследований с помощью высокомасштабируемой, надежной, безопасной, быстрой и недорогой инфраструктуры. В конце концов, Nebula надеется получить возможность автоматического масштабирования, которая позволит клиентам автоматически масштабировать облачные сервисы вверх или вниз в соответствии с определенными условиями. Saini et al. (2012) продемонстрировали, что из-за ограничений приложений Nebula пока реализовала лишь ограниченную масштабируемость.

### **11.4.2.2 Совместимость**

С самого начала Nebula создавалась для предоставления услуг, совместимых с услугами Amazon EC2. Nebula использует Eucalyptus API для управления виртуальными машинами и обеспечивает совместимость с платформами Amazon EC2 и его клонами.

### **11.4.2.3 Развертывание и интерфейс**

Nebula предоставляет пользователям графический интерфейс и набор инструментов командной строки для управления виртуальными машинами, разрешениями сетевого доступа и т.д..

### **11.4.2.4 Гипервизоры**

Nebula использует гипервизор виртуальной машины на основе ядра (KVM).

### **11.4.2.5 Надежность**

На данный момент Nebula - это экспериментальный проект. Надежность может быть повышена по мере постепенного развития программного обеспечения.

### **11.4.2.6 Поддержка ОС**

Nebula поддерживает два дистрибутива Linux, Ubuntu и CentOS.

### **11.4.2.7 Стоимость**

Nebula - это программное обеспечение с открытым исходным кодом, которое не требует затрат на само программное обеспечение. Но затраты могут быть понесены операциями. Однако публичной информации об эксплуатационных расходах Nebula пока нет.

## **11.4.3 Основные пользователи и общие комментарии**

### **11.4.3.1 Список основных клиентов**

Nebula - одно из первых облачных решений, созданных для федерального правительства федеральным правительством. Белый дом был указан как первый заказчик, когда Nebula была запущена публично.

### **11.4.3.2 Завершение**

Nebula все еще находится на начальной стадии и претерпевает значительные улучшения. В нескольких новаторских проектах, в которых реализована услуга Nebula, сообщалось о хорошей производительности системы и удовлетворительных результатах.

### **11.4.3.3 Обратная связь общественностью**

Главный сотрудник по вопросам информации США высоко оценил Nebula за то, что она помогает NASA эффективно вовлекать общественность в научную деятельность.

### **11.4.3.4 Сложность использования**

Процедура настройки исходной среды Nebula и запуска экземпляров сложна. Но шаги по добавлению новых узлов для кластерного масштабирования просты. Поскольку Nebula все еще находится на ранней стадии, время от времени требуется ручное вмешательство, когда экземпляр не загружается.

## **11.5 ЗАКЛЮЧЕНИЕ**

Как два примера завершенных коммерческих облачных сервисов, Amazon EC2 и Windows Azure обеспечивают удовлетворительную поддержку таких функций, как масштабируемость и надежность. NASA Nebula в качестве частного облака во внутренней реализации получает хороший пользовательский опыт и демонстрирует многообещающую поддержку передовых научных вычислений. У облачных сервисов Amazon и Windows Azure есть сервисные центры в нескольких географических точках. Оба они работают на рынке облачных сервисов несколько лет и постоянно улучшают свои сервисные качества. NASA Nebula была разработана, чтобы оценить возможность внедрения модели облачной обработки данных для федеральных вычислений. По сравнению со своими коммерческими аналогами, Nebula обладает гибкостью для поддержки уникальных требований, таких как безопасность, от агентств. В таблице 11.2 представлены сопоставимые аспекты трех платформ. Масштабируемость, надежность и балансировка нагрузки оказывают значительное влияние на геолого-геофизические приложения.

Таблица 11.2 Сравнение соображений готовности для трех платформ

	<i>Amazon EC2</i>	<i>Windows Azure</i>	<i>NASA Nebula</i>	<i>Описание</i>
Географическое распределение центров	Центры в 9 географических регионах	4 центра в США, 2 центра в ЕС и 2 центра в Азии	400 аккаунтов Nebula в 9 центрах NASA, Лаборатории реактивного движения и штаб-квартире NASA	—
Масштабируемость	Сервис автомасштабирования	Масштабируемый, не зависящий от вычислений и хранения	Ограниченная масштабируемость	Динамическая настройка вычислительных ресурсов под задачи
Совместимость	Стандарты облачной платформы Amazon поддерживаются многими решениями с открытым исходным кодом	Информация не найдена	Совместим с Amazon EC2	Готовность продуктов к использованию различными облачными провайдерами
Развертывание интерфейса	Консоль управления AWS, мобильное приложение для Android	Портал управления, CLI, API на основе REST	Графический интерфейс и набор инструментов командной строки.	Установ. интерфейс
Гипервизоры	Xen	Специально разработан для надежности вычислений в Windows Azure	KVM	Поддерживаемые менеджеры виртуальных машин
SLA опред. Надежность	SLA гарантирует доступность 99,95% для всех регионов Amazon EC2.	99,95% и надежности экземпляра роли и хранилища 99,9%	Информация не найдена	Механизмы отказоустойчивости в облачной реализации

## 11.6 КРАТКОЕ ИЗЛОЖЕНИЕ

В настоящей главе подробно рассматриваются три облачных сервиса и даются вспомогательные знания для следующей главы. В разделе 11.1 представлены основные факторы, которые необходимо учитывать при выборе облачных сервисов. Раздел 11.2 знакомит с Amazon Web Services (AWS). Раздел 11.3 знакомит с Windows Azure. Раздел 11.4 знакомит с туманностью НАСА. Раздел 11.5 суммирует статус трех облачных сервисов.

## 11.7 ПРОБЛЕМЫ

1. При сравнении коммерческих облачных сервисов перечислите пять ключевых моментов.
2. Перечислите шесть уровней архитектуры Amazon AWS и кратко опишите их основные функции.
3. Каковы основные преимущества и основные проблемы Amazon AWS?
4. Перечислите основные компоненты Windows Azure и кратко опишите их основные функции.
5. Каковы основные преимущества и самые большие сложности Windows Azure?
6. В чем основные отличия NASA Nebula от двух других облачных сервисов в отношении их целевых клиентов?
7. Перечислите два других популярных облачных сервиса и обсудите преимущества использования облачных сервисов?

## СПИСОК ЛИТЕРАТУРЫ

- AWS. 2012. Overview of Amazon Web Services. [http://media.amazonwebservices.com/AWS\\_Overview.pdf](http://media.amazonwebservices.com/AWS_Overview.pdf) (accessed April 28, 2013).
- Bondi, A.B. 2000. Characteristics of scalability and their impact on performance. *ACM WOSP'00 Proceedings of the 2nd International Workshop on Software and Performance*, New York, NY.
- ISACA. 2012. Cloud Computing Market Maturity: Study Results. <https://downloads.cloudsecurityalliance.org/initiatives/collaborate/isaca/2012-Cloud-Computing-Market-Maturity-Study-Results.pdf> (accessed January 24, 2013).
- Li, A., X. Yang, S. Kandula, and M. Zhang. 2010. CloudCmp: Comparing Public Cloud Providers. *IMC'10*, November 1–3, 2010. Melbourne, Australia.
- Saini, S., S. Heistand, H. Jin, J. Chang, R. Hood, P. Mehrotra, and R. Biswas. 2012. An application-based performance evaluation of NASA's Nebula cloud computing platform. *2012 IEEE 14th International Conference on High Performance Computing and Communications*, pp. 336–343. June 25–27. Liverpool, UK.
- Scheier, R. L. 2012. Virtualization Management: A Single Pane of Glass. *CIO. IN*. <http://www.cio.in/article/virtualization-management-single-pane-glass> (accessed April 19, 2013).
- Solnik, R. 2012. Cloud Management through a Single Pane of Glass, Not a Kaleidoscope. <http://www.informationweek.com/cloud-computing/infrastructure/cloud-management-through-a-single-pane-o/240001622> (accessed April 28, 2013).

- Varia, J. 2011. Amazon EC2 and AWS Elastic Beanstalk Introduction. <http://www.slideshare.net/AmazonWebServices/amazon-ec2-and-aws-elastic-beanstalk-introduction> (accessed April 28, 2013).
- Yang, C., H. Wu, Q. Huang, Z. Li, and J. Li. 2011. Using spatial principles to optimize distributed computing for enabling the physical science discoveries. *Proceedings of the National Academy of Sciences* 108, 14: 5498–5503.



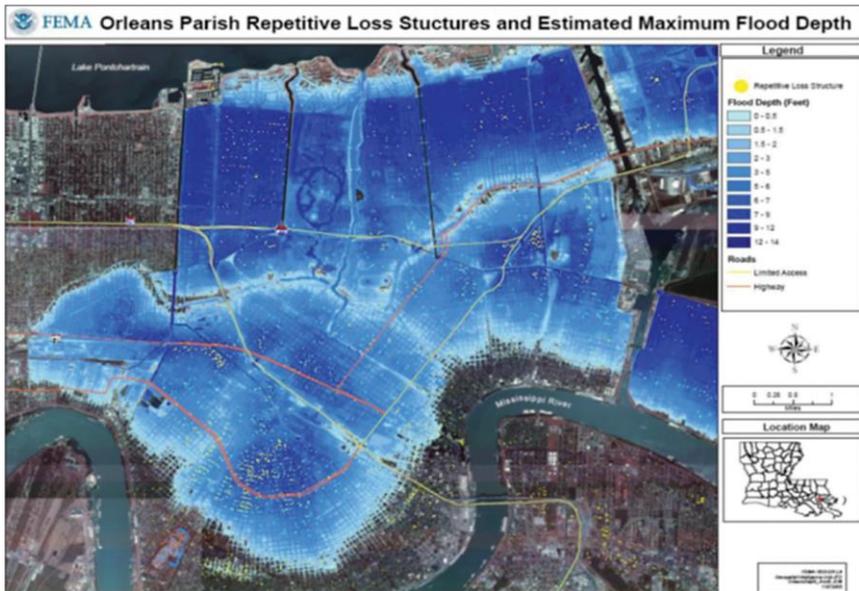


Рисунок 1.1 Карта глубины затопления Нового Орлеана после урагана Катрина. (С любезного разрешения Федерального агентства по чрезвычайным ситуациям [FEMA], Министерства внутренней безопасности.)

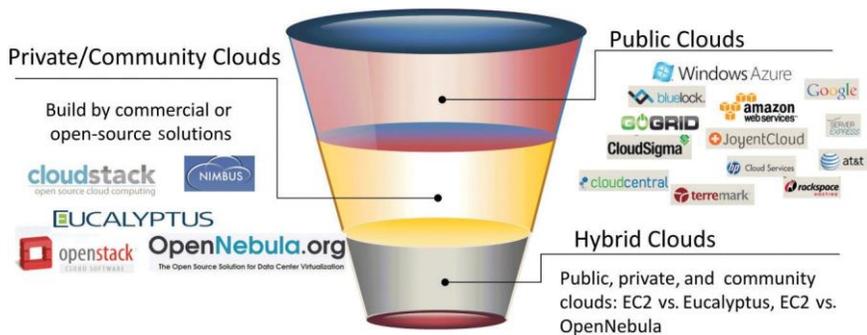


Рисунок 2.5 Типы облаков и программные решения.

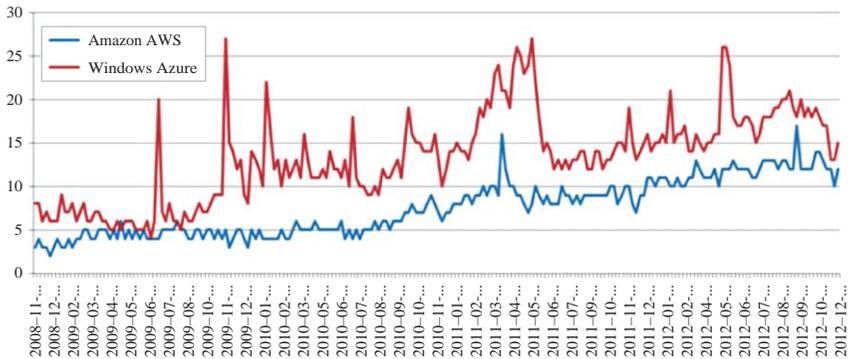


Рисунок 4.1 Тенденции поиска среди Amazon Web Services (AWS) и Windows Azure.



Рисунок 5.1 Общие шаги для облачных приложений геолого-геофизических исследований.

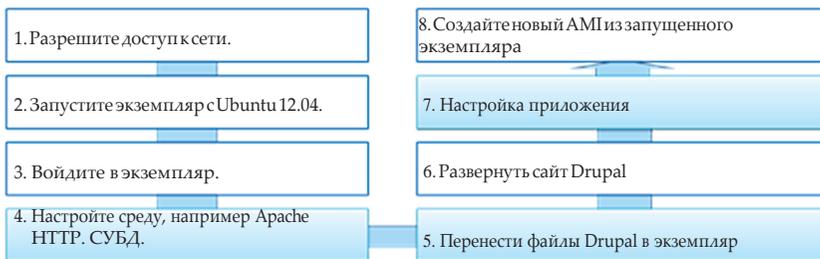
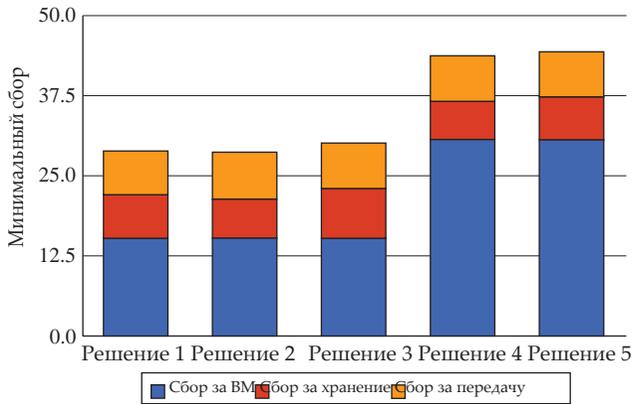


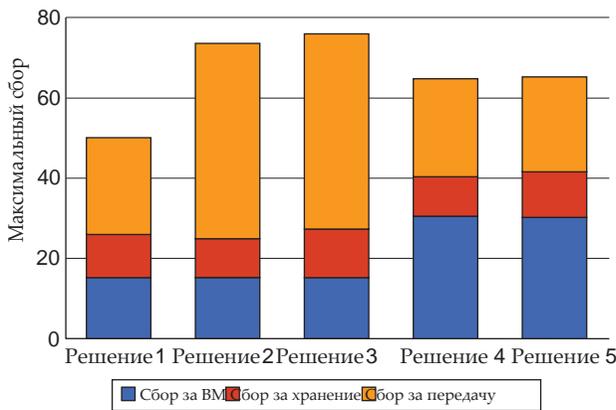
Рисунок 5.2 Процедура развертывания сайта Drupal на EC2 (синие квадраты указывают дополнительные шаги для развертывания).



Рисунок 5.9 Процесс настройки системы НРС для запуска интерполяции матрицы высот на ЕС2 (синие прямоугольники указывают дополнительные шаги для настройки виртуальной среды НРС).



(a)



(b)

Рисунок 6.5 (a) Минимальный сбор и (b) Максимальный сбор.

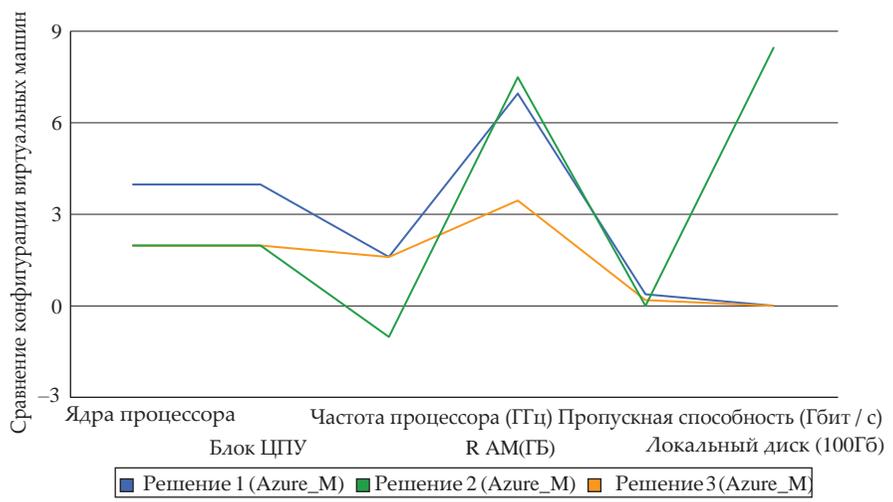


Рисунок 6.6 Сравнение конфигурации виртуальных машин.

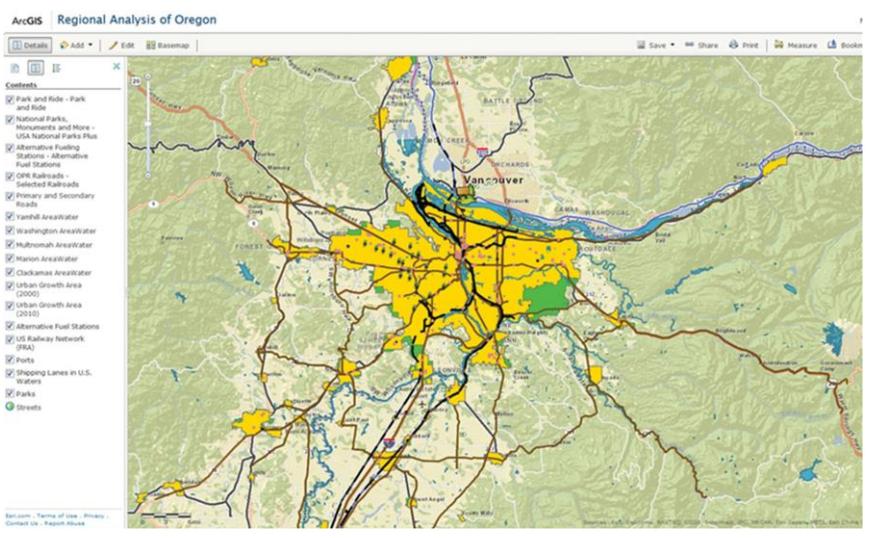


Рисунок 7.1 Региональный анализ карты Орегона.

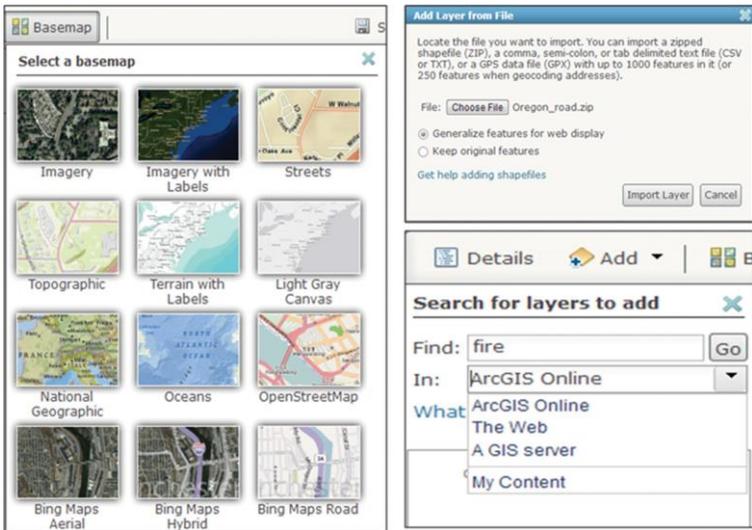


Рисунок 7.2 Базовые карты и данные, предлагаемые ArcGIS Online.

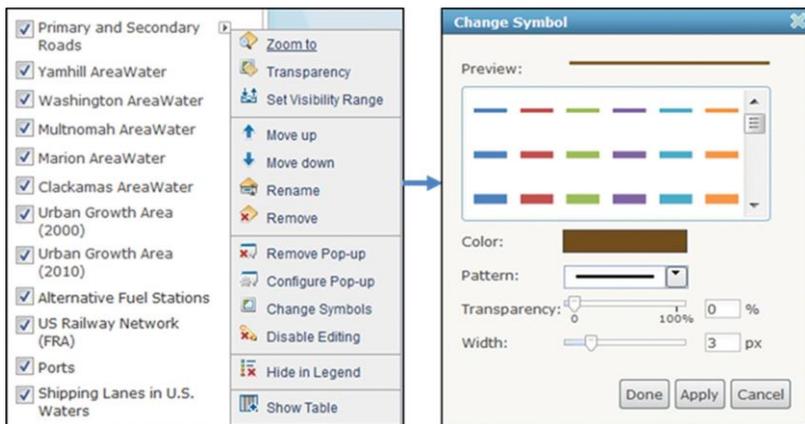


Рисунок 7.3 Модификация символа основной и второстепенной дорог.

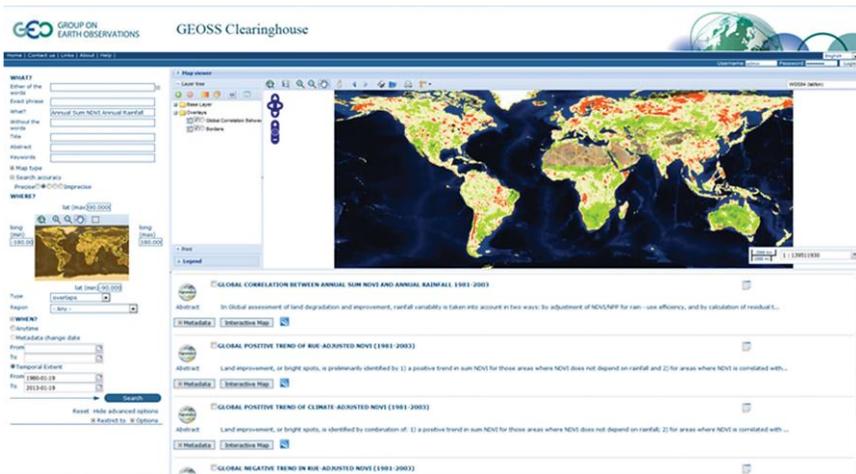


Рисунок 8.6 Результаты поиска по запросу «Annual Sum NDVI Annual Rainfall».

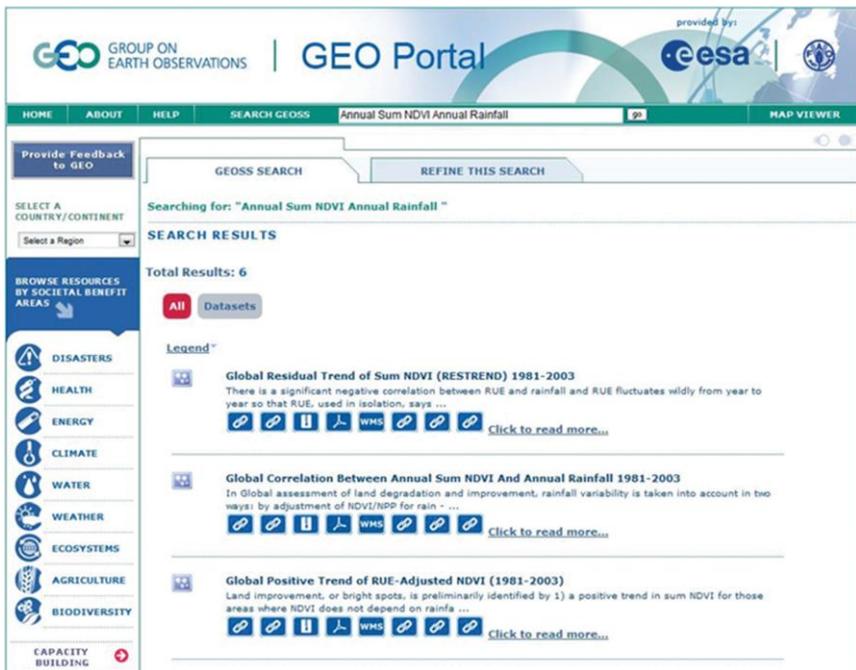


Рисунок 8.7 Результаты поиска по запросу «Annual Sum NDVI Annual Rainfall» на портале GEO.

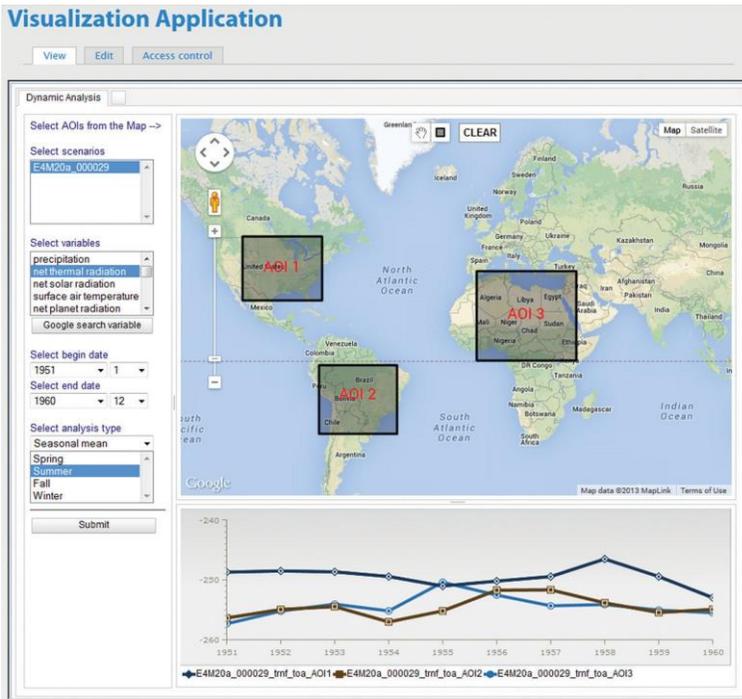


Рисунок 9.6 Анализ среднего сезонного значения с использованием результатов модели.

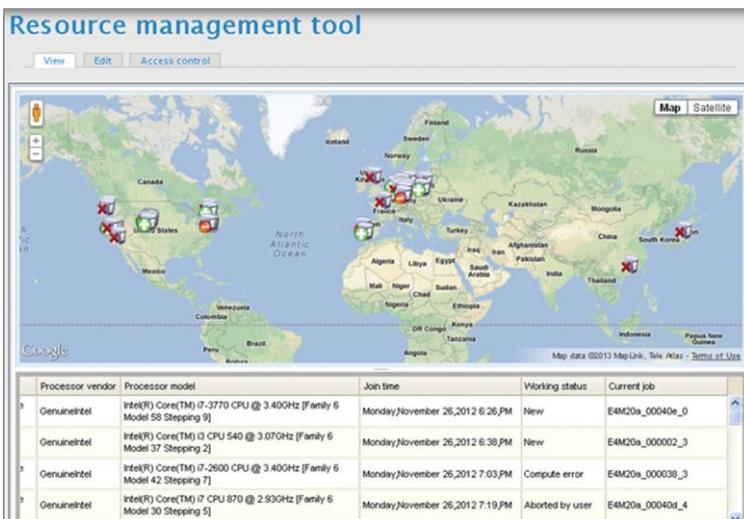


Рисунок 9.7 Визуализация добровольных вычислительных узлов с помощью Google Maps.

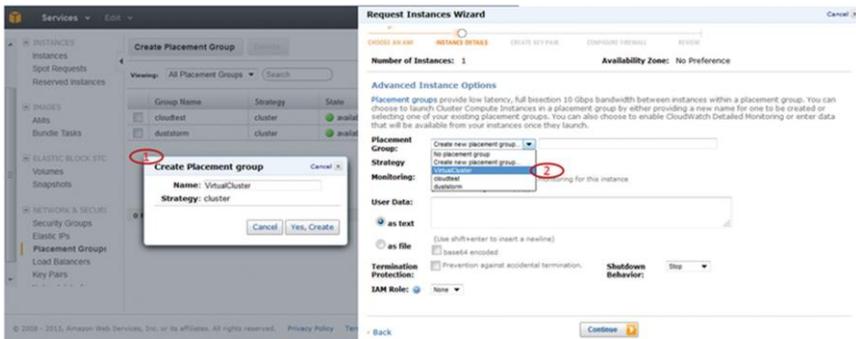


Рисунок 10.4 Создайте группу размещения и поместите экземпляры в одну группу.



Рисунок 10.5 Фотография хабуа, который поразил Финикс, штат Аризона, 1 июля 2007 г. (С любезного разрешения Osha Gray Davidson, Mother Jones, Сан-Франциско, Калифорния, <http://www.masterjones.com/blue-marble/2009/09/australias-климат-хаос.>)

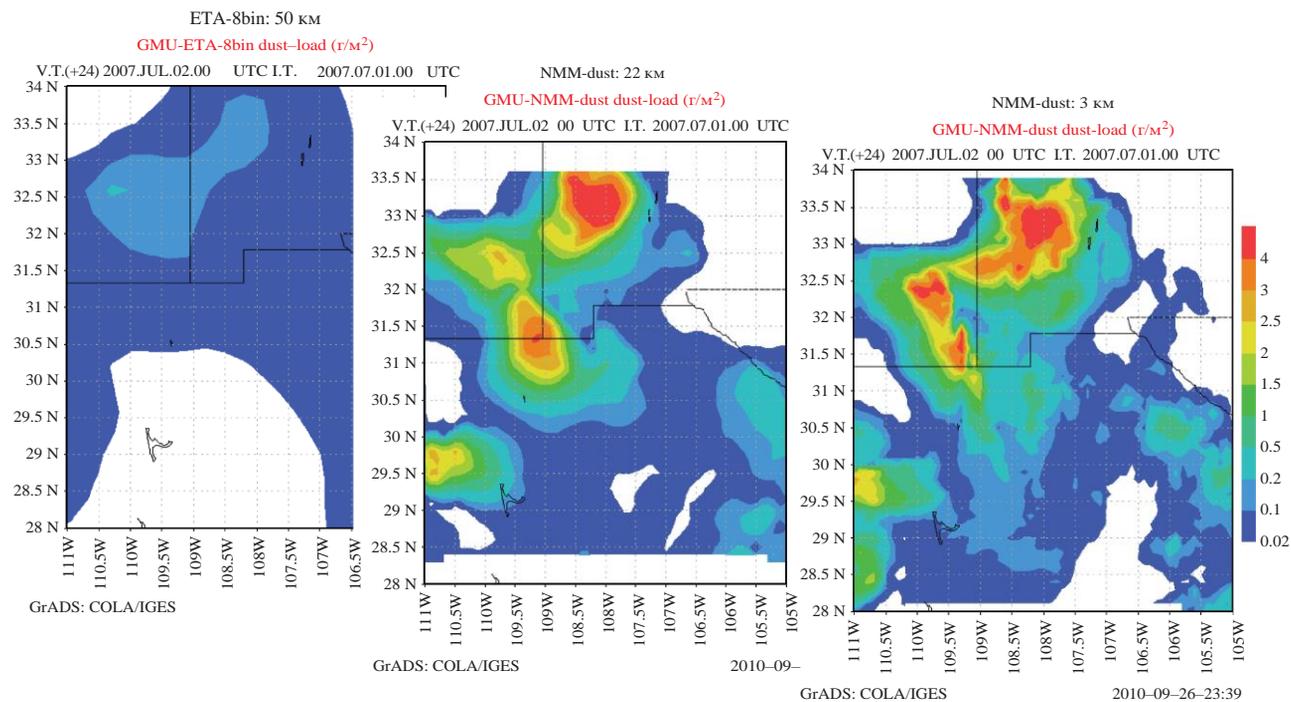


Рисунок 10.6 Сравнение результатов моделирования ETA-8bin и NMM-dust AOI 10, 11, 12 и 13 в 3 часа ночи 2 июля 2007 г.

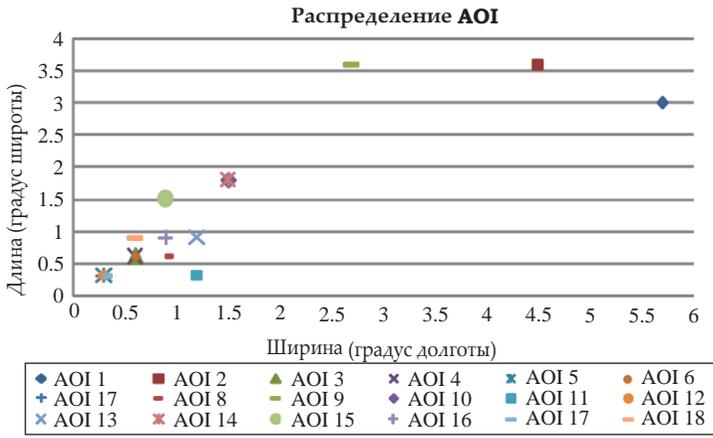


Рисунок 10.7 Распределение АОИ, определенное ETA-8bin.

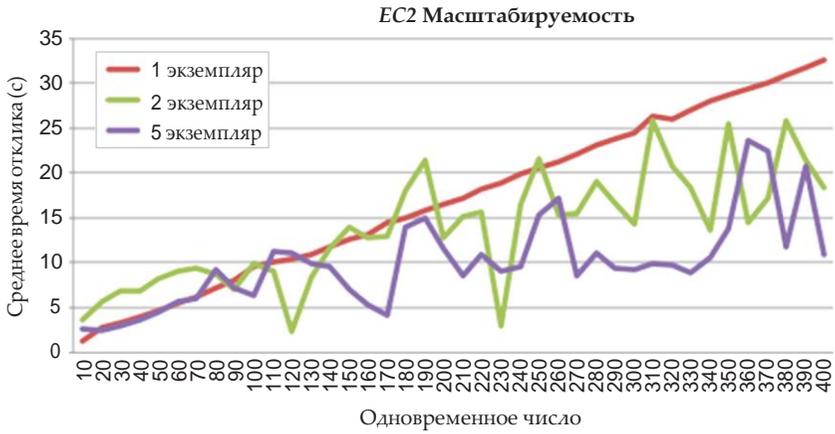


Рисунок 12.4 Масштабируемость EC2 до 1, 2 и 5 экземпляров.

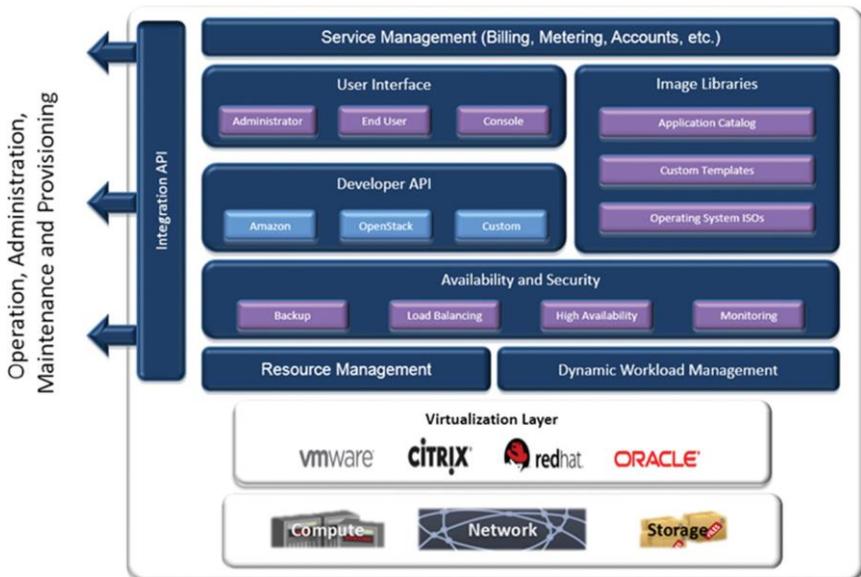


Рисунок 13.2 Архитектура CloudStack. (Взято из CloudStack Architecture на <http://www.slideshare.net/cloudstack/cloudstack-architecture>.)

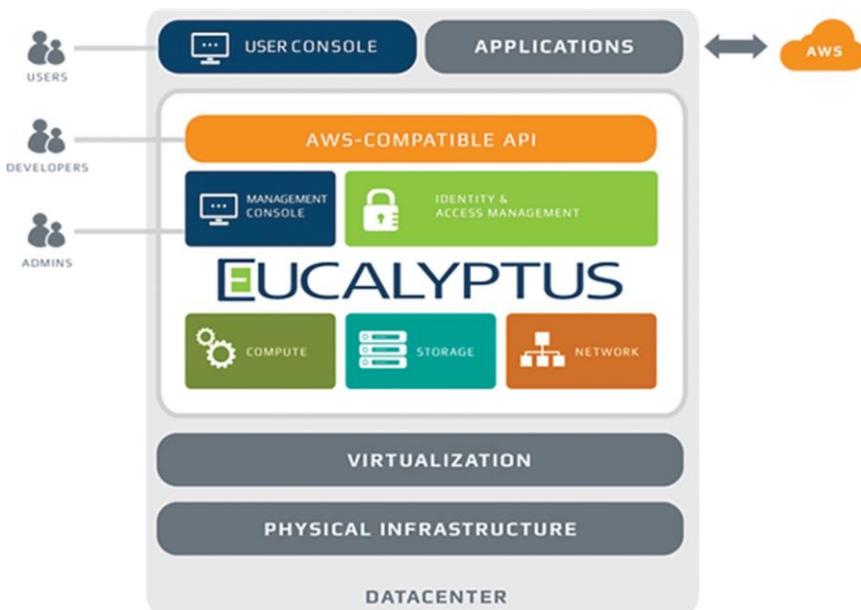
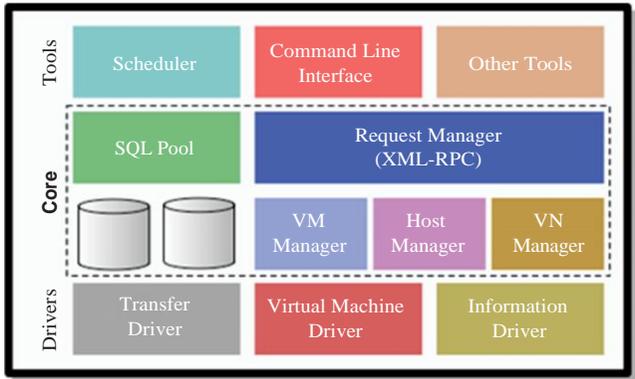
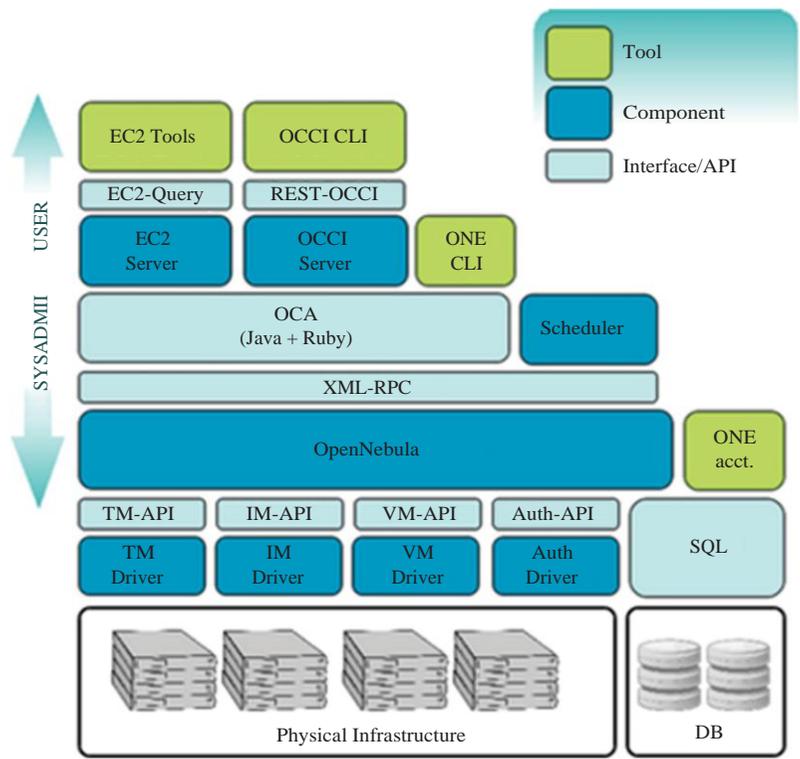


Рисунок 13.3 Архитектура Eucalyptus. (См. Eucalyptus на [http://en.wikipedia.org/wiki/File:Eucalyptus\\_Platform\\_Architecture,\\_Feb\\_February\\_2013.jpg](http://en.wikipedia.org/wiki/File:Eucalyptus_Platform_Architecture,_Feb_February_2013.jpg).)



(a)



(b)

Рисунок 13.4 (а) Внутренняя архитектура OpenNebula (см. Архитектуру OpenNebula на <http://opennebula.org/documentation:archives:rel2.0:architecture>) и (б) интерфейсы (см. Масштабируемую архитектуру OpenNebula на <http://opennebula.org/documentation:rel3.8:introapis>).

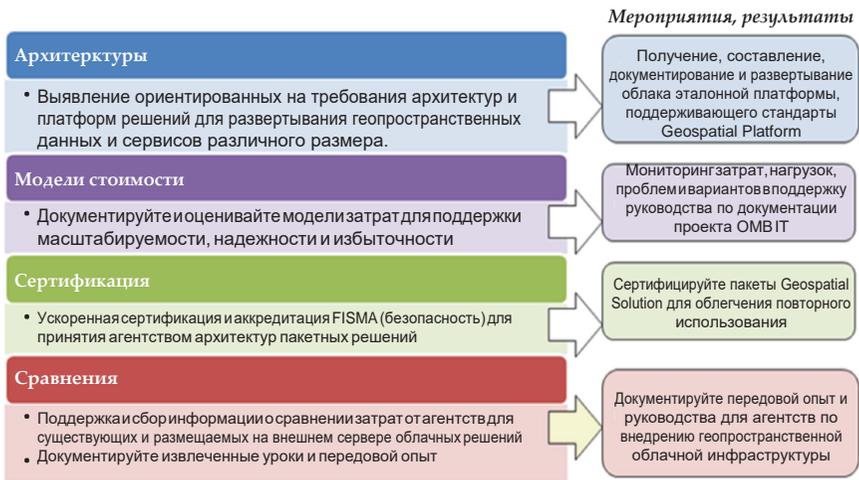
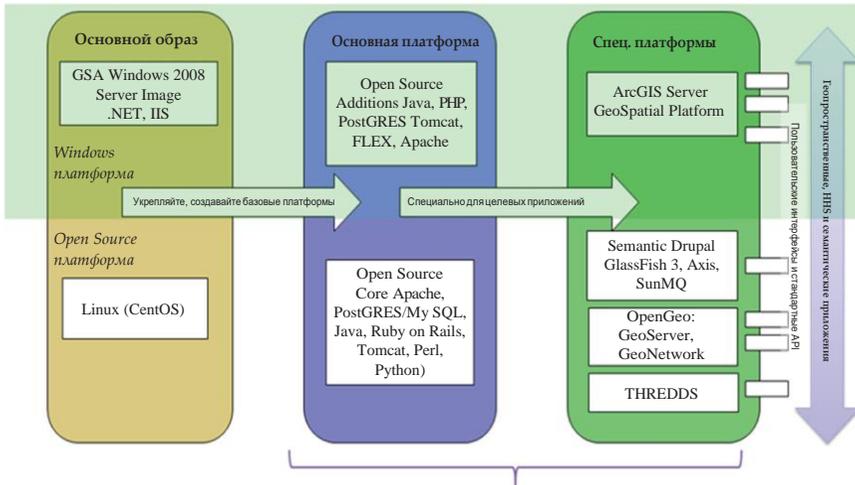


Рисунок 15.1 Цели, действия и результаты GeoCloud.



Рисунок 15.2 Архитектурный каркас GeoCloud.



Уровни (например, база данных, сервер приложений) могут быть разделены или объединены по мере необходимости.

Рисунок 15.4 Создание и настройка платформы GeoCloud для федеральных геопространственных приложений. (Из Неберта, 2010 г., [www.fgdc.gov/ngac/meetings/December-2010 / geocloud -riefing.pptx.](http://www.fgdc.gov/ngac/meetings/December-2010/geocloud-riefing.pptx))



Рисунок 16.5Г глобальная доставка контента для обработки глобального доступа пользователей.

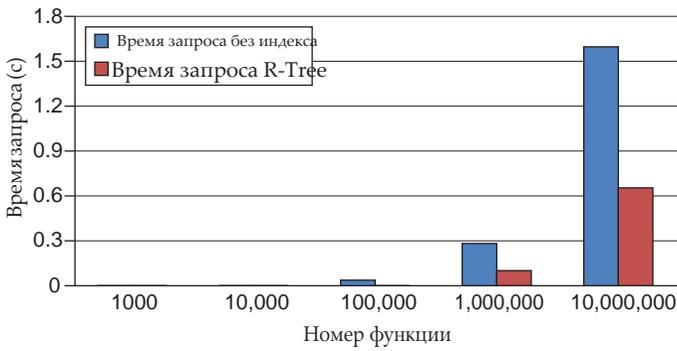


Рисунок 16.7 Пространственное индексирование для повышения скорости доступа.



## Глава 12

---

# Как проверить готовность облачных сервисов

*Чаовей Ян, Мин Сун, Цзичжэ Ся, Цзин  
Ли, Кай Лю, Цюньин Хуан и Чжипенг  
Гуй*

---

В настоящей главе используются три приложения для геофизических исследований, чтобы продемонстрировать, как проверить готовность облачных сервисов, включая NASA Nebula, Windows Azure и Amazon Elastic Cloud Computing (EC2). Результаты тестирования приложений, размещенных в облаке, сравниваются с результатами тестирования в традиционном кластере.

### 12.1 ВВЕДЕНИЕ

Расцвет облачных сервисов предоставляет различные варианты поддержки геонаучных приложений. Решение о переносе приложений в облачные сервисы частично основано на их готовности. В соответствии с вычислительными характеристиками приложений для геонаучных исследований (главы 1 и 5) готовность проверяется с помощью Информационного центра GEOSS (CLH), Climate@Home и прогнозирования пыльных бурь. В данной главе показано, как тестировать облачные сервисы на практике, оценивая производительность образцовых проектов на четырех различных вычислительных сервисах EC2, Azure, Nebula и традиционном кластере. Систематический тест и анализ готовности облачных сервисов к поддержке геонаук подробно описаны Yang et al. (2013).

### 12.2 СРЕДА ТЕСТИРОВАНИЯ

Тест проводится на разных облачных сервисах, расположенных в разных географических регионах (таблица 12.1) с различными сетевыми подключениями и конфигурациями вычислений.

#### 12.2.1 Сеть

Три облачных сервиса и кластер расположены в разных местах (Таблица 12.1) и связаны через National LambdaRail.<sup>1</sup>

---

<sup>1</sup>См. National LambdaRail на <http://www.nlr.net/>.

Таблица 12.1 Расположение вычислительных сервисов

Выч. сервисы	Геог. расположение	Облачные сервисы	Хост -Организация
EC2	Рестон, VA	IaaS	Amazon
Azure	Чикаго, IL	PaaS	Microsoft
Nebula	Эймс, CA	IaaS	NASA
Local Cluster	Фэйрфакс, VA	Традиционный кластер	George Mason University (GMU)

Таблица 12.2 Конфигурация оборудования вычислительного экземпляра

Название сервиса	Ядра ЦП	Частота ЦП (ГГц)	Память (ГБ)
EC2 1	1	2.3	0.5
EC2 2	2	2.3	7.5
EC2 3	4	2.3	7.5
EC2 4	8	2.8	7.5
Azure 1	1	1.6	1.75
Azure 2	2	1.6	3.5
Azure 3	4	1.6	7
Azure 4	8	1.6	14
Nebula 1	1	2.9	2
Nebula 2	2	2.9	4
Nebula 3	4	2.9	8
Nebula 4	8	2.9	16
Лок. сервер	8	2.4	23

Сквозное тестирование проводится для производительности каждой вычислительного сервиса. Интенсивность параллельного тока проверяется с помощью центра обмена информацией путем отправки различного количества одновременных запросов в центр обмена информацией, выполняемого для каждого вычислительного сервиса. Облачные сервисы также тестируются с использованием интенсивных вычислений Climate@Home и моделей прогнозирования пыльных бурь.

### 12.2.2 Конфигурация вычислительного сервиса

Экземпляры вычислительного сервиса настраиваются на основе доступной конфигурации оборудования (таблица 12.2).

## 12.3 ИНТЕНСИВНОСТЬ ИСПОЛЬЗОВАНИЯ GEOSS CLEARINGHOUSE В МНОГОПОТОЧНОМ РЕЖИМЕ (CLH)

### 12.3.1 Требования к информационному центру для компьютерных услуг

CLH - совместный проект FGDC, GEO и NASA, направленный на поддержку обмена данными Earth Observation (EO) в глобальном контексте. CLH обеспечивает функцию сбора метаданных данных EO из удаленных каталогов, которые были зарегистрированы в GEOSS, а также возможности поиска для обмена метаданными между глобальными пользователями.

Для поддержки механизма совместного использования сведений в CLH используются различные геопространственные стандарты, включая CSW, SRU и RSS для поиска и обнаружения, ISO 19139 для метаданных, а также стандарты Open Geospatial Consortium (OGC), связанные с WMS, WCS и WFS для доступа к данным и визуализации. Глобальный доступ пользователей к Информационному центру имеет разные пространственно-временные шаблоны для разных регионов и разных метаданных. Например, Центр обмена информацией получает больше обращений в дневное время по местному времени, чем в вечерние часы, и доступ может быть массовым, когда происходит значительное событие (например, землетрясение)

### 12.3.2 Проектирование теста

Чтобы сравнить возможности различных облачных сервисов по поддержке одновременных запросов, используются два теста с использованием CLH для сравнения среднего времени ответа для одновременного доступа:

1. Матрица проверки параллельных поисковых запросов. Когда CLH работает на четырех вычислительных сервисах, трех облачных сервисах (AWS, Azure и Nebula) и на локальном сервере (рис. 12.1), одновременные запросы отправляются от одной службы ко всем другим службам. Apache JMeter<sup>1</sup> используется для тестирования производительности одновременных запросов между службами. Результат теста представляет собой матрицу 4 \* 4, которая может представлять одновременную производительность каждой вычислительной службы по сравнению с другими вычислительными сервисами

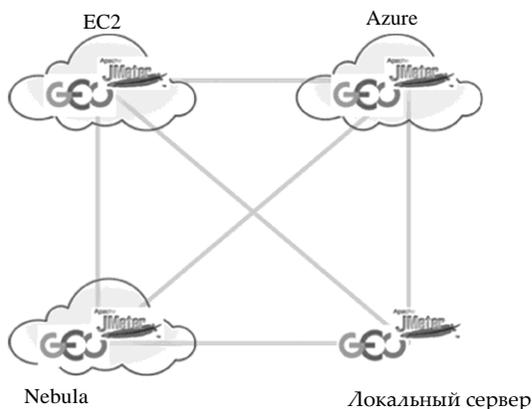


Рисунок 12.1 Дизайн матричного теста.

<sup>1</sup>См. The Apache Software Foundation на <http://jmeter.apache.org/>.

2. Тесты балансировки нагрузки и масштабировемости. Балансировка нагрузки и механизм автоматического масштабирования могут помочь поддерживать хорошую производительность за счет динамического использования нескольких экземпляров. В эксперименте с балансировкой нагрузки сравнивается одновременная производительность, когда за балансировщиком нагрузки используется разное количество экземпляров; эксперимент с автоматическим масштабированием отслеживает изменение производительности, когда облачные службы динамически масштабируют новые экземпляры с увеличением количества одновременных запросов

### 12.3.3 Последовательность выполняемых действий теста.

1. Матричный тест одновременных поисковых запросов. На рисунке 12.2 показан рабочий процесс матричного теста со следующими подробными шагами:
  - Шаг 1. Установка CLH в различных облачных сервисах. Есть два способа установить CLH: (а) Запустив экземпляр с использованием виртуального образа, образ AMI, используемый для Amazon AWS, можно получить из AWS Marketplace<sup>1</sup>, выполнив поиск «CLH». (б) Запуск экземпляра и установка CLH (см. главу 8).
  - Шаг 2. Настройка запроса CSW GetRecords - запрос CSW GetRecords может быть отправлен в CLH (см. Главу 8). Обычно запрос представляет собой файл XML и может храниться (например, post.xml) на машине, на которой размещен JMeter. В Приложении 12.1 показан пример GetRecords для поиска записей оводе.

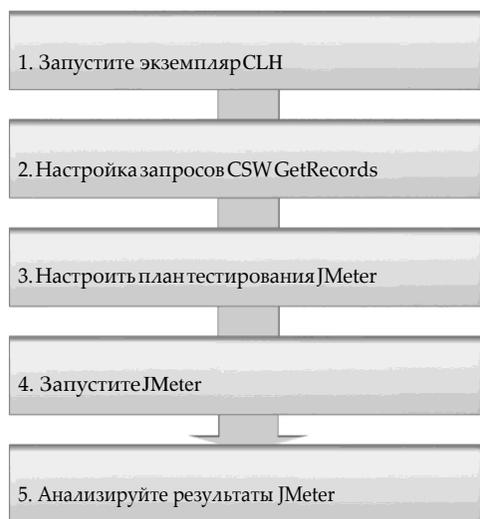


Рисунок 12.2 Последовательность выполняемых действий матрицы теста.

<sup>1</sup>См. AWS Marketplace на <https://aws.amazon.com/marketplace>.

- *Шаг 3. Настройка плана тестирования JMeter.* Тестировщики могут установить план тестирования для имитации одновременных запросов к CLN. Число одновременных пользователей может быть увеличено, например 1, 50, 100, 150, 200, 250, 300, 350, 400, 450, 500 и 600. Как правило, необходимо установить два типа конфигураций: группа потоков и контроллер. Группа потоков сохраняет основную информацию теста, такую как: количество потоков, используемое для имитации количества одновременных запросов; период нарастания, означающий общее время для отправки полного количества запросов (в тесте рекомендуется пять); и количество циклов, которое означает количество раз для выполнения теста (в тесте рекомендуется три). Помимо группы потоков, контроллер должен быть настроен для хранения информации о запросах CSW. Тестировщикам следует выбрать HTTP-запрос для настройки контроллера. В контроллере также должны быть установлены следующие параметры: IP-адрес сервера, порт, путь и тип запроса (POST), путь к XML-сообщению и путь для сохранения файла результатов. В Приложении 12.2 показан пример плана тестирования, который имитирует 100 запросов к CLN за 1 секунду.
- *Шаг 4. Запуск JMeter* - JMeter можно запустить двумя способами: графический интерфейс пользователя (GUI) и командная строка. Запуск графического интерфейса пользователя можно выполнить, нажав «Выполнить» -> «Пуск», если JMeter установлен с графическим интерфейсом. Между тем, мы также можем запустить JMeter через командную строку. Следующий сценарий используется для запуска разных планов тестирования с разным количеством потоков. Тестировщикам необходимо перейти в папку bin JMeter и выполнить следующую команду.

```
# Create the array of the threads number
threads=(1 50 100 150 200 250 300 350 400 450 500 600
700 800 900 1000)
# Run the test using jmeter from this machine to GMU
server
for (( i = 0; i <=11; i++))
do
    for (( j = 0; j <=2; j++))
do
    head=./jmx/GEOSSTestPlan_
    tail=.jmx
    planName=$head${threads[i]}$tail
    ./jmeter -n -t $planName
    sleep 20
done
done
```

- *Шаг 5. Анализировать результаты JMeter* - данный шаг описан в Разделе 12.3.4.

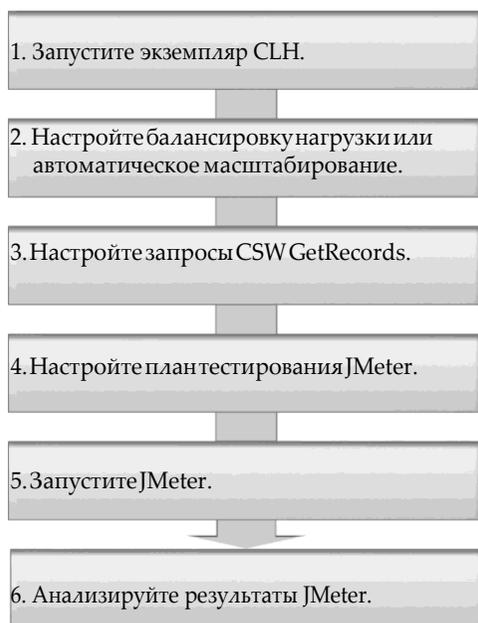


Рисунок 12.3 Рабочий процесс теста балансировки нагрузки и автомасштабирования.

2. Тесты балансировки нагрузки и автоматического масштабирования (рисунок 12.3). Данные два теста можно использовать для проверки улучшения производительности при использовании балансировки нагрузки или автоматического масштабирования для CLH. Тестировщики могут выполнять пошаговое тестирование, чтобы использовать от 10 до 400 одновременных запросов для выполнения тестов. Кроме того, тестировщики также могут настроить, сколько экземпляров будет задействовано в тесте, например 1, 2 и 5 экземпляров.

На рисунке 12.3 показан рабочий процесс теста балансировки нагрузки / автоматического масштабирования. Шаги 1, 3, 4, 5, 6 можно назвать матричным тестом. Шаг 2 настраивает балансировку нагрузки или автоматическое масштабирование, как описано в главе 8. Загрузка ЦП рекомендуется использовать в качестве триггера ресурсов. Например, тестировщики могут настроить файл масштабируемости для добавления нового экземпляра, когда средняя загрузка ЦП за 300-секундный период превышает 90%, и для удаления экземпляра, когда средняя загрузка ЦП за 300-секундный период меньше 75%. Подробную конфигурацию можно найти в онлайн-материалах к главе 12

### 12.3.4 Анализ результатов теста

JMeter записывает информацию об ответе в файл результатов, как это было установлено на шаге 3. Файл результатов содержит основную информацию о каждом потоке, такую как статус ответа и время ответа. Тестировщики могут сравнить время отклика, используя файл результатов.

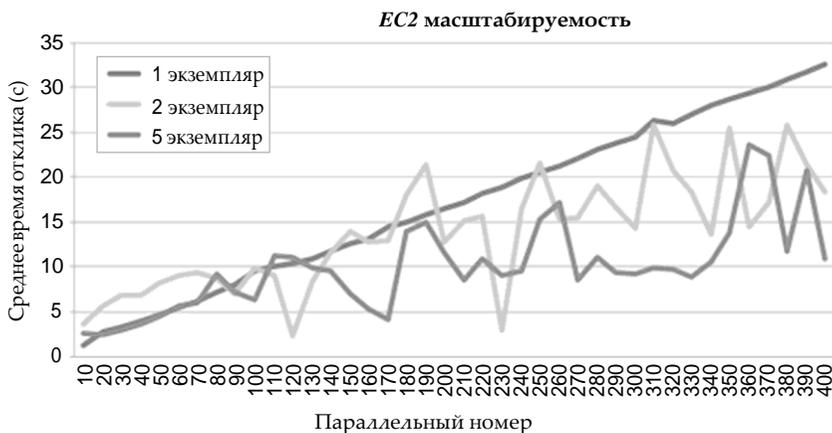


Рисунок 12.4 (См. цветную вставку) Масштабируемость EC2 до 1, 2 и 5 экземпляров.

Например, на рис. 12.4 показано среднее время ответа GEOSS, использующего Amazon EC2 для масштабирования различных уровней одновременных запросов. Проводятся три группы экспериментов: без увеличения и уменьшения с помощью всего одного экземпляра, автоматическое масштабирование до 2 экземпляров и автоматическое масштабирование до 5 экземпляров. Правило масштабирования экземпляра - это когда время ответа / задержки превышает 4 секунды. Согласно рисунку 12.4, использование 2 и 5 экземпляров для масштабирования одновременного доступа может уменьшить среднее время отклика. После 50–60 одновременных запросов, если среднее время ответа превышает 4 секунды, группа «2 экземпляра» и «5 экземпляров» запустила новый экземпляр. Поскольку для запуска CLN требуется время, среднее время начинает уменьшаться, когда количество одновременных операций равно 80. Файл журнала Amazon AWS показывает, что группа из «5 экземпляров» запускает третий экземпляр через 10 минут после запуска второго экземпляра, четвертый экземпляр находится на 8 минут позже после запуска третьего экземпляра, а пятый экземпляр - на 9 минут позже после запуска четвертого экземпляра. Следовательно, есть несколько сбросов, когда количество одновременных запросов составляет 170, 210 и 270 для группы «5 экземпляров».

## 12.4 ТЕСТ ДАННЫХ И ВЫЧИСЛИТЕЛЬНОЙ ИНТЕНСИВНОСТИ С ПОМОЩЬЮ CLIMATE @HOME

### 12.4.1 Требования к вычислениям Climate @ Home

Проект Climate@Home направлен на создание виртуального суперЭВМ для крупномасштабного моделирования климата и имитаций с использованием вычислительных ресурсов, предоставленных гражданами (глава 9). ModelE была выбрана в качестве модели климата. При включенном режиме последовательной обработки минимальные требования к оборудованию для выполнения модели довольно низкие (Глава 9). Вообще говоря, машина Linux, оснащенная одним ядром ЦП с частотой 1,5+ ГГц и 1 ГБ оперативной памяти, соответствует минимальным вычислительным требованиям.

Помимо требований к вычислениям, следует также учитывать начальную емкость хранилища данных, обеспечиваемую машиной. Хотя модель и связанные с ней входные данные занимают всего несколько мегабайт пространства, модель может давать огромные результаты (например, около одного ГБ для моделирования одного года). Следовательно, экземпляру облака следует выделить достаточно места на жестком диске для базовой установки, которая включает модель, входные данные и выходы модели.

## 12.4.2 Проектирование теста

Цель данного теста - продемонстрировать, как различные облачные сервисы поддерживают моделирование климата. Чтобы быть конкретным, тест разбит на следующие три задачи:

1. Определение рентабельной аппаратной конфигурации экземпляра путем сравнения временных затрат на выполнение модели запускается с виртуальными экземплярами различных конфигураций оборудования. Поскольку ModelE - это приложение для Linux, для сравнения используются только Amazon EC2 или Nebula (поддерживает экземпляры Linux). Набор образцов экземпляров с разным количеством ядер ЦП и сопоставимой оперативной памятью создается для изучения конфигурации облачных экземпляров, влияющих на время выполнения прогонов модели (Dai et al. 2009; Onisick 2011).
2. Оценка стабильности и надежности экземпляров виртуальных машин путем записи общего времени отказов и среднего времени, затраченного на выполнение нескольких запусков. Для каждого экземпляра выполняется несколько прогонов модели с разными периодами времени моделирования (например, один день, один месяц). Такой процесс завершается при выполнении общего сценария, который может контролировать общее количество запусков модели и изменять периоды времени моделирования для каждого запуска модели.
3. Сравнение производительности облачных виртуальных машин и физических машин. Экземпляры виртуальных машин имеют ту же конфигурацию оборудования, что и физические машины. Одни и те же прогоны модели выполняются как на виртуальных экземплярах, так и на физической машине.

## 12.4.3 Последовательность выполнения теста

Исходя из схемы тестов, рабочий процесс тестирования определяется следующим образом (рисунок 12.5):

- *Шаг 1.* Настройка вычислительной среды. Учитывая вычислительные требования для различных прогонов моделей и доступные ресурсы, мы составили список конфигураций оборудования и создали экземпляры на основе таких конфигураций.

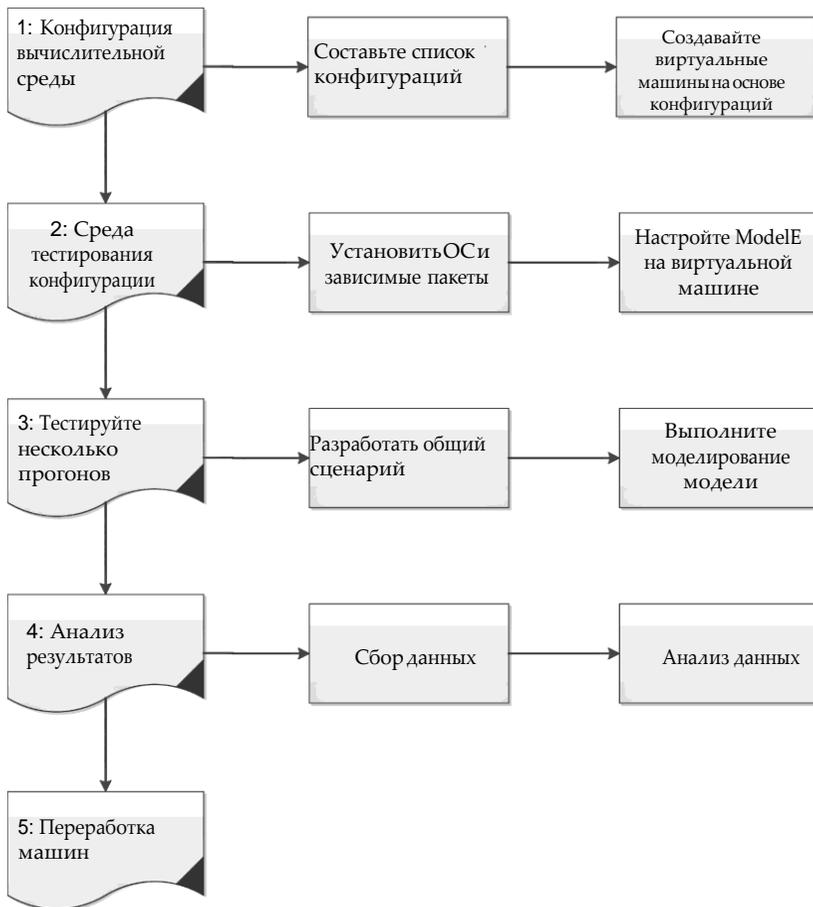


Рисунок 12.5 Рабочий процесс эксперимента по моделированию модели Climate@Home.

В этом случае физическая машина оснащена 8-ядерным процессором и 8 Гб оперативной памяти. Соответственно, хотя бы один экземпляр должен иметь одинаковую конфигурацию оборудования. Конфигурации ЦП других экземпляров могут быть 1-ядерными, 2-ядерными и 4-ядерными.

- *Шаг 2.* Настройка среды тестирования - для всех машин выбирается один и тот же тип операционной системы (ОС), что может минимизировать разницу между результатами тестирования, вызванную разными ОС. ОС также должна удовлетворять особым требованиям компиляции ModelE (например, версия gcc). Для каждой машины необходимо правильно установить и настроить ModelE. Конфигурация ModelE на экземплярах очень похожа на конфигурацию на физических машинах, за исключением того, что к экземпляру необходимо получить доступ через удаленное управление.

Подробную процедуру настройки модели на физической машине можно найти на официальном сайте ModelE.<sup>1</sup> В то время как один образ шаблона ОС с настроенной ModelE (доступный на веб-сайте книги) предпочтительнее для создания набора экземпляров с учетом количества экземпляры. Это не применимо к определенным облачным службам, где операционная система предоставляется как часть облачных служб. Настройка ModelE после запуска экземпляра - это общая стратегия для кроссплатформенных облачных сервисов. Обратите внимание, что ModelE настроен для последовательной обработки.

- *Шаг 3.* Тестирование нескольких запусков с помощью сценария оболочки Linux на всех машинах (экземпляре и компьютере). Общий сценарий содержит программу Linux, которая постоянно изменяет время моделирования в файле конфигурации ModelE, а затем запускает моделирование модели. Скрипт также управляет временем ожидания между симуляциями модели. Выполнение модели будет перезапущено после завершения одного эксперимента, чтобы избежать ошибок времени выполнения. Статус и время выполнения каждого прогноза модели записываются в файл журнала. Для каждого экземпляра сценарий следует скопировать в корневой каталог. После завершения моделирования на экземпляре файл журнала загружается на центральную управляющую машину для дальнейшего анализа. Пример сценария можно найти на сайте книги. Сценарий можно изменить для облегчения автоматического моделирования нескольких экспериментов по запросу.

```
#time:19491208
echo "Begin to run the model at " >logModel19491208.
txt
cd decks
make setup RUN=E4M20one
cd ..
sleep 30 &
wait
date
sed -i 's/YEARE=1961,MONTH=1,DATEE=1,HOURE=0/YEARE=1
949,MONTH=12,DATEE=8,HOURE=0/g' decks/E4M20one/I
##replace simulation time here
date >>logModel19491208.txt
echo "----real start----" >>logModel19491208.txt
date >>logModel19491208.txt
time1=$(date +%s)
./exec/runE_new E4M20one -np 1
date >>logModel19491208.txt
echo "----real end----" >>logModel19491208.txt
time2=$(date +%s)
echo "Model finished in: " $(($time2-$time1))
>>logModel19491208.txt
```

<sup>1</sup>См. NASA на [http://www.giss.nasa.gov/tools/modelE/HOWTO.html#part0\\_2](http://www.giss.nasa.gov/tools/modelE/HOWTO.html#part0_2).

```
Begin to run the model at
Fri Nov 18 15:58:41 PST 2011
----real start----
Fri Nov 18 15:58:41 PST 2011
Fri Nov 18 16:06:06 PST 2011
----real end-----
Model finished in: 445
```

Рисунок 12.6 Пример файла журнала («logModel19491208.txt», одноядерный экземпляр от Nebula).

- *Шаг 4.* Сбор и анализ файлов журнала. Поскольку файл журнала хранится отдельно для каждого экземпляра, файлы журнала собираются для каждого экземпляра. Это включает в себя передачу файлов с экземпляра на компьютер для интеграции. Методы анализа этих результатов обсуждаются в Разделе 12.4.4.
- *Шаг 5.* Очистка и освобождение экземпляра. После завершения тестов экземпляры следует закрыть и сохранить должным образом, чтобы избежать ненужного использования ресурсов.

#### 12.4.4 Результаты теста и анализы

Конфигурации виртуальных машин включают в себя 1-ядерные, 2-ядерные, 4-ядерные и 8-ядерные экземпляры с сопоставимой основной памятью (таблица 12.2). После выполнения общего сценария в каждом экземпляре временные затраты на каждый прогон модели записываются в файл журнала, помеченный временем окончания каждого периода моделирования. Например, файл logModel19491208.txt указывает, что файл журнала записывает временные затраты в секундах для периода моделирования с 1949/12/01 по 1949/12/08. Ниже приведен пример файла журнала для одноядерного экземпляра Nebula (рис. 12.6). В файле журнала записывается время начала выполнения, время окончания выполнения и временные затраты на выполнение. Есть два времени запуска: одно записывается после уведомления машины о подготовке к запуску модели, а другое - время при выдаче команды выполнения. При анализе файла журнала нам нужно четко обозначить связь файлов журнала с конфигурациями их компьютеров, которые не отражаются в имени файла.

## 12.5 ОБЛАЧНЫЙ ТЕСТ С ИСПОЛЬЗОВАНИЕМ ИНСТРУМЕНТА ПРОГНОЗИРОВАНИЯ ПЫЛЬНЫХ БУРЬ

### 12.5.1 Требования к вычислениям для прогнозирования пыльных бурь

Прогнозирование пыльных бурь включает в себя большие объемы данных геолого-геофизических исследований на входе и выходе модели, интенсивную обработку данных и вычисления (глава 10).

Периодические явления моделирования пыльных бурь требуют нескольких вычислительных ресурсов для выполнения моделирования параллельно с одним и тем же набором интенсивных вычислений много раз. Для решения вычислительных задач обычно используются современные вычислительные методологии и методы (Yang et al. 2011a). Предыдущее исследование моделирования пыльной бури (Хуанг и др., 2013) показало, что скорость ЦП и сети оказывает значительное влияние на производительность моделирования из-за массового обмена данными и синхронизации (как описано в главе 10). Следовательно, облачные виртуальные машины для прогнозирования пыльных бурь должны иметь высокую скорость ЦП и быть подключены к сети с высокой пропускной способностью.

## 12.5.2 Разработка теста

Модель пыльной бури настроена на локальном кластере высокопроизводительных вычислений (HPC) и двух различных облачных сервисах: Amazon EC2 и NASA's Nebula. В таблице 12.3 показана конфигурация трех вычислительных сервисов.

Чтобы изучить производительность локального кластера HPC и двух облачных сервисов, мы разработали следующие эксперименты:

1. Различное количество вычислительных узлов из разных сервисов для поддержки одного и того же моделирования пыльной бури. В этом эксперименте используются один, два, четыре и восемь вычислительных узлов кластеров Nebula, EC2 и GMU HPC для тестирования одной и той же задачи моделирования.
2. Одинаковое количество вычислительных ресурсов от разных сервисов для поддержки разных задач моделирования. Два вычислительных узла используются для выполнения семи задач моделирования на каждой платформе, чтобы проверить, согласована ли производительность различных платформ.
3. Влияние гиперпоточности. Технология Hyperthreading делает один физический процессор похожим на несколько логических процессоров и значительно повышает производительность вычислительных рабочих нагрузок (Koufaty and Marr 2003).

Таблица 12.3 Конфигурации трех вычислительных сервисов высокопроизводительных вычислений

Конфигурация	EC2 4	Nebula 4	GMU HPC
ЦП	2.93 ГГц	2.8 ГГц	2.33 ГГц
Кол-во процессоров	2	2	2
Количество ядер / узлов	8	8	8
Поддерживает гиперпоточность	Да (Итого 16 ядер процессора)	Нет	Нет
Общий объем памяти	23 Гб	16 Гб	16 Гб
Пропускная способность сети	10 Гбит / с	10 Гбит / с	1 Гбит / с
Виртуализация	Чистая виртуализация Группирование виртуальных машин на более близких физических машинах	Паравиртуализация C/O	C/O C/O
Стратегия размещения ресурсов	Физических машинах		

Основным преимуществом технологии гиперпоточности является ее способность улучшать использование ресурсов процессора (Vulpin and Pratt 2004). Чтобы проверить влияние гиперпоточности, эксперимент разработан для проверки производительности одного экземпляра EC2 до и после отключения возможности гиперпоточности. В этом эксперименте мы использовали один экземпляр EC2 с возможностью гиперпоточности и без нее, чтобы запустить одно и то же моделирование. Тип экземпляра кластера EC2 включает четырехъядерные процессоры с гиперпоточностью. В документации EC2 указано, что вычислительному экземпляру кластера назначены два четырехъядерных процессора (всего 8 ядер); Возможности гиперпоточности процессоров привели к тому, что эталонный тест показал всего 16 ядер (таблица 12.2).

### 12.5.3 Последовательность выполняемых действий теста

Первый тест в Разделе 12.5.2 исследует разные сервисы с использованием разного количества вычислительных узлов для запуска одного и того же моделирования модели с помощью следующих 10 основных шагов (рисунок 12.7):

- *Шаг 1.* Создание образов среды модели пыльной бури для облачных сервисов (EC2 и Nebula) и настройка среды пыльной бури в локальном кластере HPC - в разделе 10.3 представлены шаги по настройке среды HPC и построению образов для модели пыльной бури.

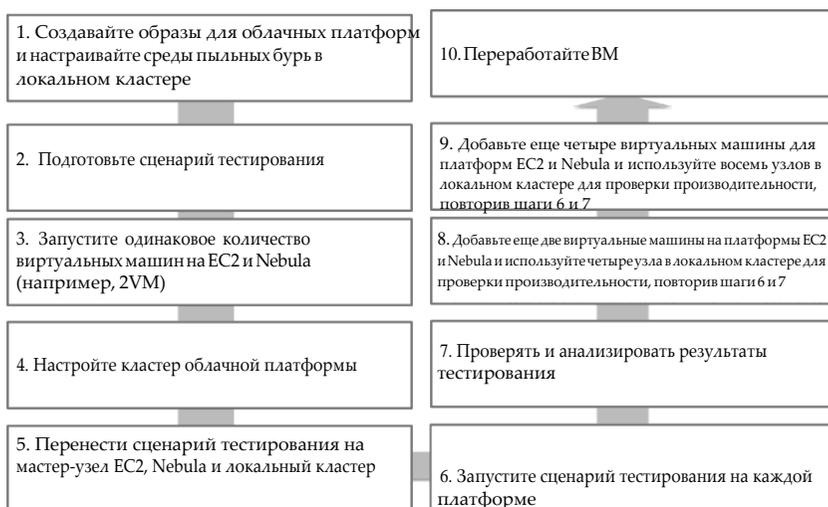


Рисунок 12.7 Рабочий процесс тестирования производительности трех разных платформ с 2, 4 и 8 виртуальными машинами с использованием модели пыльной бури.

При создании локальной кластерной среды НРС необходимо следовать тем же процедурам, за исключением создания образа, который не требуется для локального кластера НРС.

- *Шаг 2. Написание скрипта тестовой оболочки для запуска модели и записи производительности.*

```
#!/bin/sh
## usage: ./run_test.sh [test platform] [VM numbers]
platform=$1
vms=$2
## Run model and record time
cd /model_directory/model
startTime='date +%s%N'
./run_model.sh
endTime='date +%s%N'
echo 'expr \( $endTime - $startTime \) / 1000000'
## compress and copy model output data back to
repository
tar zcvf $platform.test.$ vms.tar.gz modeloutput
scp $platform.test.$ vms.tar.gz root@199.26.xxx.xxx:/
state/partition1/
```

- *Шаг 3.* Запуск одинакового количества экземпляров на платформе EC2 и Nebula - на основе образов, созданных на шаге 1, потребитель может запускать разное количество экземпляров. В нашем тесте мы сначала проверим производительность двух экземпляров.
- *Шаг 4.* Настройка облачной платформы. Данный шаг предназначен для обеспечения правильной работы MPICH2 на виртуальных машинах.
- *Шаг 5.* Передача скрипта тестирования на головной узел EC2, Nebula и локальный кластер НРС. Скрипты тестирования, написанные на шаге 2, следует скопировать в головной узел каждой платформы.
- *Шаг 6.* Запуск скрипта тестирования на каждой платформе. Если имя файла сценария - run\_test.sh, то мы можем использовать следующую команду для запуска сценария для платформы EC2 с двумя экземплярами

```
[root@headnode~] chmod a+x run_test.sh
[root@headnode~] ./run_test.sh EC2 2VMs
```

- *Шаг 7.* Проверка результатов каждой платформы и анализ результатов. Поскольку модель работает на нескольких ресурсах, сетевое взаимодействие между такими ресурсами иногда может давать сбой. Следовательно, потребители должны входить в головной узел и проверять ход выполнения и вывод модели каждые один или два часа, чтобы убедиться, что модель работает правильно.
- *Шаг 8.* Добавление еще двух экземпляров для платформы EC2 и Nebula и использование четырех узлов для проверки производительности путем повторения

*Шаги 6 и 7*—После того, как экспериментальная группа с двумя экземплярами будет завершена, мы можем запустить еще два экземпляра и сформировать кластер НРС с четырьмя узлами как для ЕС2, так и для Nebula, а также добавить два сервера в локальный кластер НРС. Чтобы убедиться, что ЦП и память в первых двух экземплярах освобождены от предыдущих задач, мы можем остановить сервисы MPICH2<sup>1</sup> и подождать несколько минут. Затем мы можем начать следующий этап моделирования с четырьмя экземплярами.

- *Шаг 9.* Добавив еще четыре экземпляра для платформы ЕС2 и Nebula и используя восемь узлов для проверки производительности, повторив шаги 6 и 7 - повторяя шаг 8, мы можем создать восемь экземпляров и снова запустить тест.
- *Шаг 10.* Повторное использование экземпляров. После завершения тестов экземпляры можно отключить и передать другим приложениям.

Второй экспериментальный тест (раздел 12.5.2) заключается в использовании одного и того же количества вычислительных ресурсов с разных платформ для поддержки различных задач имитации. Процедура тестирования аналогична рис. 12.7, где мы должны включить шаги с 1 по 7. В такой ситуации пользователи могут запустить любое количество экземпляров (например, один, два или четыре) для тестирования различных имитаций размера проблемы.

В третьем тесте технологии гиперпоточности (раздел 12.5.2) мы можем запустить виртуальную машину с гиперпоточностью и без нее, чтобы выполнить ту же задачу моделирования. По умолчанию для всех экземпляров ЕС2 включена гиперпоточность. Следующая команда может использоваться для отключения гиперпоточности для экземпляра ЕС2.

```
for cpunum in $(  
  
    cat/sys/devices/system/cpu/cpu*/topology/thread_  
        siblings_list |  
cut -s -d, -f2- | tr ',' '\n' | sort -un); do  
    echo 0 > /sys/devices/system/cpu/cpu$cpunum/online  
done
```

## 12.5.4 Анализ результатов теста

На рисунке 12.8 показан типичный результат теста с разными номерами процессов. Первая строка \*\*\*\*\* Тип теста ЕС2.2VMs \*\*\*\*\* указывает, что данный результат теста облачного сервиса ЕС2 с двумя экземплярами. Третья строка указывает рабочий каталог модели. Четвертая строка означает время начала запуска модели со 128 номерами процессов. В пятой строке указано время завершения модели со 128 процессами. Следующие строки указывают время начала и окончания с другими номерами процессов.

<sup>1</sup>См. MPICH2 на <http://phase.hpcc.jp/mirrors/mpi/mpich2/>.

```
***** Test type EC2.2VMs*****
*****Begin to run*****
/mnt/mirror/performance/test.1/nmmgmu3km.iop.0/model.parallel.8/exe
Fri Nov 11 20:58:57 UTC 2011
***** after 128 time is*****
Fri Nov 11 21:25:18 UTC 2011
***** after 120 time is*****
Fri Nov 11 21:50:14 UTC 2011
***** after 112 time is*****
Fri Nov 11 22:13:45 UTC 2011
***** after 104 time is*****
Fri Nov 11 22:37:07 UTC 2011
***** after 96 time is*****
Fri Nov 11 22:59:59 UTC 2011
***** after 80 time is*****
Fri Nov 11 23:22:04 UTC 2011
***** after 64 time is*****
Fri Nov 11 23:44:44 UTC 2011
***** after 48 time is*****
Sat Nov 12 00:11:35 UTC 2011
***** after 32 time is*****
Sat Nov 12 00:39:50 UTC 2011
***** after 16 time is*****
Sat Nov 12 01:14:56 UTC 2011
Finish tar begin copy data
Finish copy data
Wed May 23 19:48:13 EDT 2012
***** finish*****
```

Рисунок 12.8 Результат тестирования модели на EC2 с двумя экземплярами и разными номерами процессов.

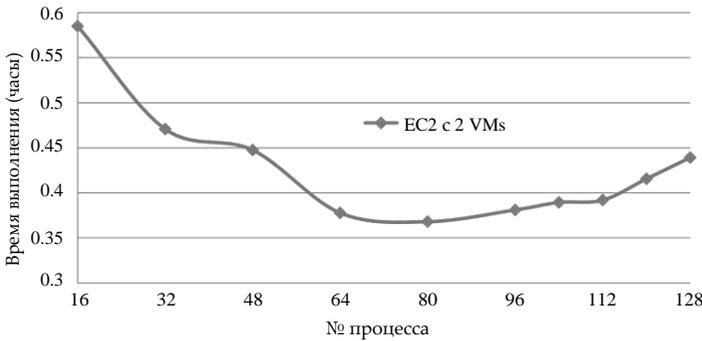
Последняя строка с Finish показывает, что тест успешно завершен.

Посредством обработки выходных данных тестового файла, показанного на рисунке 12.8, можно рассчитать и проанализировать время выполнения трехчасового прогнозирования модели с точностью  $2,3 * 3,5$  градуса облачным сервисом EC2 с использованием двух экземпляров и различных номеров процессов (рисунок 12.9a). Комбинируя различные выходные файлы тестов, можно выполнить более сложный анализ (рисунок 12.9b).

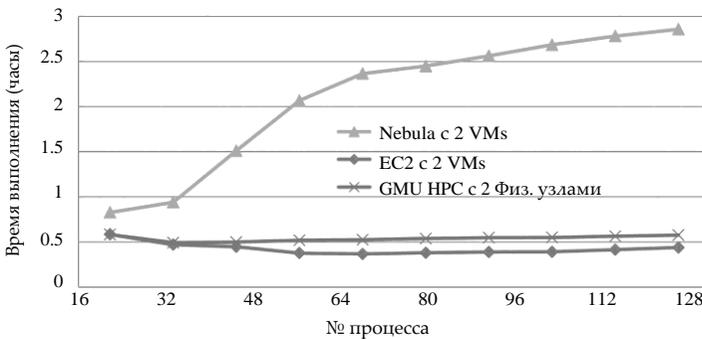
## 12.6 КРАТКОЕ ИЗЛОЖЕНИЕ

Геонаучные приложения могут являться данными, вычислениями, параллельными и пространственно-временными приложениями (Yang et al. 2011b). В данной главе описывается, как использовать три приложения для геонаучных исследований, CLN, Climate @ Home и прогнозирования пыльных бурь, для тестирования различных облачных сервисов. В разделах 12.1 и 12.2 представлены фоновая и тестовая среда. Раздел 12.3 вводит параллельное тестирование интенсивности. Раздел 12.4 вводит тест интенсивности данных и вычислений. Раздел 12.5 представляет комплексное тестирование с использованием проекта пыльной бури. Метод тестирования, опыт и сценарии, представленные на онлайн-сайте книги, могут быть использованы для тестирования других облачных сервисов. Больше результатов и систематических тестов сообщает Yang et al. (2013).

## Как проверить готовность облачных сервисов 217



(а) EC2 производительность с 2 VMs



(б) Различная производительность вычислительной платформы

Рисунок 12.9 Характеристики модели пылевой бури с разными платформами и номерами процессов.

### 12.7 ПРОБЛЕМЫ

1. Каковы вычислительные характеристики геонаучных приложений?
2. Используйте пример, чтобы проиллюстрировать, как проверить возможности облачного приложения в ответ на одновременную интенсивность.
3. Используйте пример, чтобы проиллюстрировать, как протестировать возможности облачного приложения в ответ на интенсивность данных.
4. Используйте пример, чтобы проиллюстрировать, как тестировать возможности облачного приложения в ответ на интенсивность вычислений.
5. Какие возможности размещенного в облаке приложения можно проверить с помощью приложения для моделирования пылевой бури?
6. Какие общие инструменты используются в данной главе для проверки готовности облачных сервисов?
7. Прочтите отчет о результатах (Yang et al. 2013) и опишите результаты в 500 словах. Обсудите динамику результатов, то есть то, как результаты могут измениться.

## ПРИЛОЖЕНИЕ 12.1: ПРИМЕР GETRECORDS ДЛЯ ПОИСКА МЕТАДААННЫХ

```
<?xml version="1.0"?>
<csw:GetRecords xmlns:csw="http://www.opengis.net/cat/
  csw/2.0.2" xmlns:gmd="http://www.isotc211.org/2005/gmd"
  service="CSW" version="2.0.2">
  <csw:Query typeNames="csw:Record">
    <csw:Constraint version="1.1.0">
      <Filter xmlns="http://www.opengis.net/
        ogc" xmlns:gml="http://www.opengis.
        net/gml">
        <PropertyIsLike wildCard="%"
          singleChar="_" escapeChar="\ ">
          <PropertyName>AnyText</
            PropertyName>
          <Literal>%water%</Literal>
        </PropertyIsLike>
      </Filter>
    </csw:Constraint>
  </csw:Query>
</csw:GetRecords>
```

## ПРИЛОЖЕНИЕ 12.2: ПРИМЕР ПЛАНА ТЕСТИРОВАНИЯ jmeter

```
<?xml version="1.0" encoding="UTF-8"?>
<jmeterTestPlan version="1.2" properties="2.1">
  <hashTree>
    <TestPlan guiclass="TestPlanGui" testclass="TestPlan"
      testname="GEOSSTestPlan_1" enabled="true">
      <stringProp name="TestPlan.comments">This the for
        the GEOSS Cloud Test</stringProp>
      <boolProp name="TestPlan.functional_mode">>false</
        boolProp>
      <boolProp name="TestPlan.serialize_
        threadgroups">>true</boolProp>
      <elementProp name="TestPlan.user_defined_
        variables" elementType="Arguments"
        guiclass="ArgumentsPanel" testclass="Arguments"
        testname="User Defined Variables"
        enabled="true">
        <collectionProp name="Arguments.arguments"/>
      </elementProp>
    <stringProp name="TestPlan.user_define_classpath"></
      stringProp>
    </TestPlan>
  </hashTree>
```

## Как проверить готовность облачных сервисов 219

---

```
<ThreadGroup guiclass="ThreadGroupGui"
testclass="ThreadGroup" testname="Thread_Group_1"
enabled="true">
<stringProp name="ThreadGroup.on_sample_
error">continue</stringProp>
<elementProp name="ThreadGroup.main_controller"
elementType="LoopController"
guiclass="LoopControlPanel"
testclass="LoopController" testname="Loop Controller"
enabled="true">
<boolProp name="LoopController.continue_
forever">>false</boolProp>
<stringProp name="LoopController.loops">1</stringProp>
</elementProp>
<stringProp name="ThreadGroup.num_threads">100</
stringProp>
<stringProp name="ThreadGroup.ramp_time">1</
stringProp>
<longProp name="ThreadGroup.start_
time">1319814086000</longProp>
<longProp name="ThreadGroup.end_time">1319814086000</
longProp>
<boolProp name="ThreadGroup.scheduler">>false</
boolProp>
<stringProp name="ThreadGroup.duration"></stringProp>
<stringProp name="ThreadGroup.delay"></stringProp>
</ThreadGroup>
<hashTree>
<HTTPSamplerProxy guiclass="HttpTestSampleGui"
testclass="HTTPSamplerProxy" testname="HTTP_Request_
GetRecords" enabled="true">
<elementProp name="HTTPSampler.Arguments"
elementType="Arguments" guiclass="HTTPEndpointsPanel"
testclass="Arguments" testname="User Defined
Variables" enabled="true">
<collectionProp name="Arguments.arguments"/>
</elementProp>
<stringProp name="HTTPSampler.domain">clearinghouse.
cisc.gmu.edu</stringProp>
<stringProp name="HTTPSampler.port">80</stringProp>
<stringProp name="HTTPSampler.connect_timeout"></
stringProp>
<stringProp name="HTTPSampler.response_timeout"></
stringProp>
<stringProp name="HTTPSampler.protocol"></
stringProp>
<stringProp name="HTTPSampler.contentEncoding"></
stringProp>
```

```
<stringProp name="HTTPSampler.path">geonetwork/srv/
en/csw</stringProp>
<stringProp name="HTTPSampler.method">POST</
stringProp>
<boolProp name="HTTPSampler.follow_redirects">true</
boolProp>
<boolProp name="HTTPSampler.auto_redirects">>false</
boolProp>
<boolProp name="HTTPSampler.use_keepalive">true</
boolProp>
<boolProp name="HTTPSampler.DO_MULTIPART_
POST">>false</boolProp>
<elementProp name="HTTPSampler.Files"
elementType="HTTPFileArgs">
<collectionProp name="HTTPFileArgs.files">
<elementProp name="/root/jakarta-jmeter-2.5.1/bin/
post.xml" elementType="HTTPFileArg">
<stringProp name="File.path">/root/jakarta-
jmeter-2.5.1/bin/post.xml</stringProp>
<stringProp name="File.paramname"></stringProp>
<stringProp name="File.mimetype">application/xml</
stringProp>
</elementProp>
</collectionProp>
</elementProp>
<boolProp name="HTTPSampler.monitor">>false</boolProp>
<stringProp name="HTTPSampler.embedded_url_re"></
stringProp>
</HTTPSamplerProxy>
<hashTree/>
<ResultCollector guiclass="TableVisualizer"
testclass="ResultCollector" testname="View Results in
Table" enabled="true">
<boolProp name="ResultCollector.error_
logging">>false</boolProp>
<objProp>
<name>saveConfig</name>
<value class="SampleSaveConfiguration">
<time>true</time>
<latency>true</latency>
<timestamp>true</timestamp>
<success>true</success>
<label>true</label>
<code>true</code>
<message>true</message>
<threadName>true</threadName>
<dataType>true</dataType>
<encoding>>false</encoding>
<assertions>true</assertions>
```

```
<subresults>true</subresults>
<responseData>>false</responseData>
<samplerData>>false</samplerData>
<xml>true</xml>
<fieldNames>>false</fieldNames>
<responseHeaders>>false</responseHeaders>
<requestHeaders>>false</requestHeaders>
<responseDataOnError>>false</responseDataOnError>
<saveAssertionResultsFailureMessage>>false</
saveAssertionResultsFailureMessage>
<assertionsResultsToSave>0</assertionsResultsToSave>
<bytes>true</bytes>
</value>
</objProp>
<stringProp name="filename">/home/result/result.xml</
stringProp>
</ResultCollector>
<hashTree/>
</hashTree>
</hashTree>
</hashTree>
</jmeterTestPlan>
```

## СПИСОК ЛИТЕРАТУРЫ

- Bulpin, J. R. and I. A. Pratt. 2004. Multiprogramming performance of the Pentium 4 with hyper-threading. In *Workshop on Duplicating, Deconstructing, and Debunking (WDDD04)*, June.
- Dai, Y., B. Yang, J. Dongarra, and G. Zhang. 2009. Cloud Service Reliability: Modeling and Analysis. From *The 15th IEEE Pacific Rim International Symposium on Dependable Computing. Hyperthreading*, 2011. <http://en.wikipedia.org/wiki/Hyper-threading> (accessed April 25, 2013).
- Huang Q., C. Yang, K. Benedict, S. Chen, A. Rezgui, and J. Xie. 2013. Utilize cloud computing to support dust storm forecasting. 6(4): 338–355.
- Koufaty, D. and D.T. Marr. 2003. Hyperthreading technology in the netburst micro-architecture. *Micro, IEEE* 23, no. 2: 56–65.
- Mell, P. and T. Grance. 2009. The NIST Definition of Cloud Computing Ver. 15. <http://src.nist.gov/groups/SNS/cloud-computing/> (accessed April 25, 2013).
- Onisick, J. 2011. Cloud availability 101. *Network Computing*, August.
- Yang, C., M. Goodchild, Q. Huang et al. 2011a. Spatial cloud computing—How can geospatial sciences use and help to shape cloud computing? *International Journal of Digital Earth* 4: 305–329.
- Yang, C., H. Wu, Q. Huang, Z. Li, and J. Li. 2011b. Using Spatial Principles to Optimize Distributed Computing for Enabling the Physical Science Discoveries. doi: 10.1073/pnas.0909315108.
- Yang, C., Q. Huang, J. Xia et al. 2013. Cloud service readiness for geospatial sciences. *International Journal of Geographic Information Science* (in press).



## Глава 13

---

# Решения для облачной обработки данных с открытым исходным кодом

*Чэнь Сюй, Чжипенг Гуй, Цзин Ли, Кай Лю,  
Цюньбин Хуан и Мира Бамбакус*

---

В дополнение к коммерческим облачным сервисам многие решения облачной обработки данных с открытым исходным кодом можно гибко адаптировать для создания частных облачных сервисов для конкретных требований пользователя. В данной главе представлены четыре основных решения для облачных вычислений с открытым исходным кодом, включая CloudStack, Eucalyptus, Nimbus и OpenNebula.

### 13.1 ВВЕДЕНИЕ В РЕШЕНИЯ ОБЛАЧНОЙ ОБРАБОТКИ ДАННЫХ С ОТКРЫТЫМ ИСХОДНЫМ КОДОМ

В то время как все больше коммерческих ИТ-предприятий предоставляют облачные сервисы через свои продукты (Armbrust et al. 2010; Huang et al. 2012), появляющиеся облачные решения с открытым исходным кодом способны преобразовать существующую инфраструктуру организации в частный или гибридный облачный сервис (Sotomayor et al. 2009).

Ключевой задачей ИТ-предприятий при создании облачного сервиса является комплексное управление физическими и виртуальными ресурсами (такими как серверы, хранилище и сети) (Rimal et al. 2011). Программный инструментарий, отвечающий за такую оркестровку, называется диспетчером виртуальной инфраструктуры (VIM) (рисунок 13.1). Как правило, VIM должны предоставлять набор функций, включая (1) управление и мониторинг жизненного цикла виртуальных машин (ВМ), например создание и выпуск ВМ, соединение виртуальных машин и настройка виртуальных дисков для ВМ, (2) размещение и динамическая замена виртуальных машин в пуле физической инфраструктуры, (3) планирование виртуальных машин на физических машинах и (4) обеспечение сетевых возможностей, таких как настройка общедоступных и частных IP-адресов и доменных имен, чтобы виртуальные машины были доступны через сеть.

CloudStack, Eucalyptus, Nimbus и OpenNebula - четыре самых популярных модуля VIM, доступных для создания множества облачных сервисов на основе частных и общественных организаций. Четыре VIM представлены в следующем разделе.

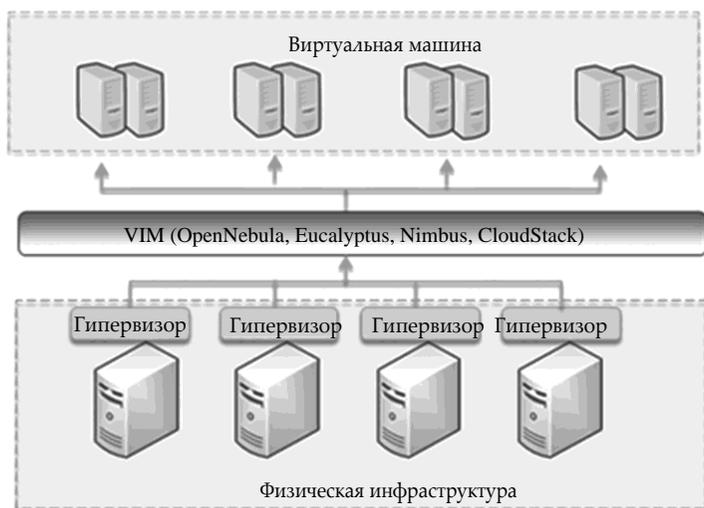


Рисунок 13.1 Архитектура инфраструктуры как услуги (IaaS) под управлением VIM.

### 13.1.1 Cloudstack

CloudStack - это VIM с открытым исходным кодом. CloudStack может использовать несколько гипервизоров, например три гипервизора, представленные в главе 3 (VMware ESXi, Xen и KVM). Помимо собственного API, он также реализует API Amazon EC2 и S3 для поддержки взаимодействия с облачными сервисами Amazon. CloudStack изначально был разработан cloud.com, стартапом, поддерживаемым венчурным капиталом. В конце 2011 года сайт cloud.com был приобретен Citrix, а CloudStack был передан в дар Apache Software Foundation в 2012 году. В настоящее время CloudStack находится под лицензией Apache License, версия 2.

### 13.1.2 eucalyptus

Eucalyptus - это аббревиатура от Elastic Utility Computing Architecture, связывающая ваши программы с полезными системами (Eucalyptus). Как программное обеспечение VIM с открытым исходным кодом, Eucalyptus может поддерживать корпоративные частную и гибридную облачную обработку данных. Eucalyptus поддерживает гибридные облачные сервисы за счет интеграции API Amazon Web Services (AWS). Таким образом, потребители могут создать гибридный облачный сервис, используя Eucalyptus для частного облака и Amazon для публичного облака, и перемещать виртуальные машины между двумя облаками. Эвкалипт был разработан исследователями из Калифорнийского университета в Санта-Барбаре. Первая версия программного обеспечения была выпущена в 2008 году. Последняя версия - Eucalyptus 3.2.1<sup>1</sup>, выпущенная в феврале 2013 года.

<sup>1</sup>См. Download Eucalyptus на <http://www.eucalyptus.com/download/eucalyptus>.

### 13.1.3 nimbus

Nimbus фокусируется на вычислительных потребностях научных пользователей. Nimbus имеет два варианта: Nimbus Infrastructure и Nimbus Platform. Инфраструктура Nimbus предоставляет решение для IaaS и была разработана для поддержки потребностей научно-исследовательских проектов с интенсивным использованием данных. Платформа Nimbus - это набор инструментов, которые помогают потребителям использовать облачные вычисления IaaS. Набор инструментов включает функции для установки, настройки, мониторинга и ремонта приложений. Платформа Nimbus позволяет потребителям создавать гибридные облака инфраструктуры Nimbus, Amazon AWS и других облаков. Nimbus поддерживает гипервизоры Xen и KVM. Nimbus был разработан Аргоннской национальной лабораторией Чикагского университета.

### 13.1.4 opennebula

OpenNebula может управлять частными, общедоступными или гибридными облаками IaaS. Пользователи OpenNebula могут гибко решать, какие облачные интерфейсы следует адаптировать из EC2 Query, Open Cloud Computing Interface (OC CI) OGF или vCloud. Гипервизоры Xen, KVM и VMware поддерживаются OpenNebula. OpenNebula была запущена как исследовательский проект в 2005 году. Первая версия была выпущена в 2008 году. К 2013 году было более 5000 загрузок в месяц<sup>1</sup>. Последняя версия OpenNebula - 3.8.3.<sup>2</sup>

В следующих разделах (разделы с 13.2 по 13.5) подробно описываются архитектура и характеристики четырех облачных решений с открытым исходным кодом.

## 13.2 CloudStack

CloudStack разработан для управления большими сетями виртуальных машин для обеспечения высокой доступности и масштабируемости платформы IaaS. Он включает в себя весь «стек» функций, которые требуются большинству организаций в облаке IaaS: <sup>3</sup> оркестровка вычислений, сеть как услуга, управление пользователями и учетными записями, полный и открытый собственный API, учет ресурсов и удобный пользовательский интерфейс (UI).

### 13.2.1 Архитектура

Архитектура CloudStack<sup>4</sup> (рис. 13.2) предусматривает платформу IaaS с вычислительными, сетевыми и хранилищами ресурсов, которыми нужно управлять, как описано: все ресурсы связаны вместе посредством общей архитектуры, включающей как минимум одно решение гипервизора. CloudStack предоставляет базовую модель сегментации, основанную на управлении учетными записями и распределении ресурсов.

<sup>1</sup> См. OpenNebula на <http://c12g.com/open-source-release-of-the-opennebulaapps-suite/>.

<sup>2</sup> См. OpenNebula Download на <http://opennebula.org/software:software>.

<sup>3</sup> См. Apache CloudStack на <http://incubator.apache.org/cloudstack/>.

<sup>4</sup> См. CloudStack Architecture на <http://www.slideshare.net/cloudstack/cloudstack-architecture>.

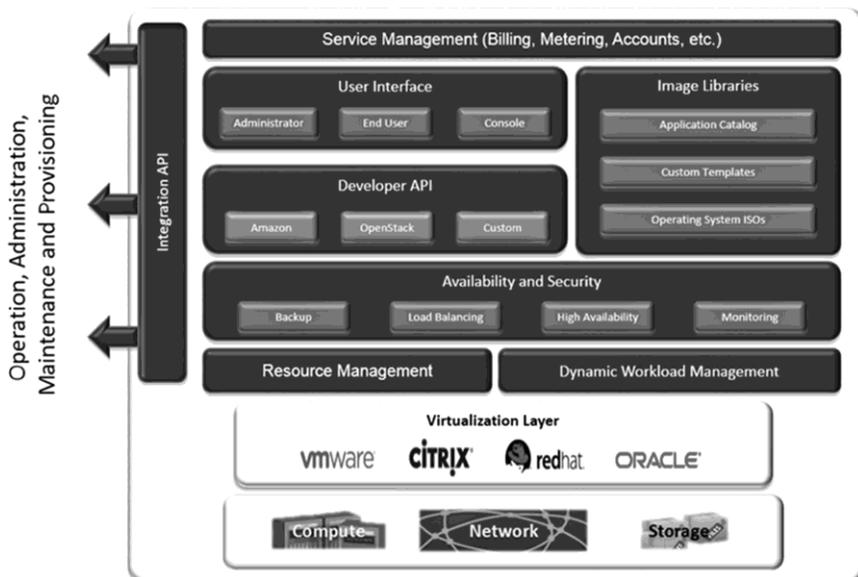


Рисунок 13.2 (См. цветную вставку) Архитектура CloudStack. (Взято из CloudStack Architecture по адресу <http://www.slideshare.net/cloudstack/cloudstack-architecture>.)

Данная модель обеспечивает мультитенантный режим для поддержки абстракции аренды, начиная от ведомственного режима и заканчивая общедоступным режимом посредника. Он включает в себя основные функции, такие как пользовательский интерфейс и управление изображениями, и позволяет поставщикам облачных услуг предоставлять расширенные услуги, такие как высокая доступность и балансировка нагрузки. Все сервисы связаны друг с другом через серию API-интерфейсов веб-сервисов, которые полностью автоматизируют CloudStack для поддержки уникальных потребностей потребителей.

### 13.2.2 Общие характеристики Cloudstack

- Масштабируемость - CloudStack может управлять большим количеством серверов в географически распределенных центрах обработки данных с помощью линейно масштабируемого централизованного сервера управления. Эта возможность устраняет необходимость в промежуточных серверах управления на уровне кластера.<sup>1</sup> Она поддерживает интеграцию как с программными, так и с аппаратными межсетевыми экранами и балансировщиками<sup>2</sup> нагрузки для обеспечения дополнительной безопасности и масштабируемости облачной среды пользователя, такой как балансировщик нагрузки F5<sup>2</sup> и Netscaler.<sup>3</sup>

<sup>1</sup> См. Understanding Apache CloudStack на <http://incubator.apache.org/cloudstack/software.html>.

<sup>2</sup> См. F5 Load Balancer на <http://www.f5.com/glossary/load-balancer/>.

<sup>3</sup> См. Citrix на <http://www.citrix.com/products/netscaler-application-delivery-controller/overview.html>.

## Решения для облачной обработки данных с открытым исходным кодом 227

---

- Облачная модель. CloudStack - это решение для облачных платформ уровня IaaS. Он объединяет, управляет и настраивает сеть, хранилище и вычислительные узлы для поддержки общедоступных, частных и гибридных облаков IaaS.
- Совместимость - помимо собственных API-интерфейсов CloudStack совместим с API-интерфейсами Amazon EC2 и S3<sup>1</sup>, а также с API-интерфейсами vCloud для приложений гибридного облака<sup>2</sup>.
- Развертывание и интерфейс. Пользователи могут управлять своими облаками с помощью удобного веб-интерфейса, инструментов командной строки или RESTful API. CloudStack предоставляет готовый к использованию многофункциональный пользовательский интерфейс, который реализует CloudStack API для управления инфраструктурой. Это решение на основе AJAX, совместимое с наиболее популярными веб-браузерами. Снимок состояния агрегированного хранилища, пулов IP, ЦП, памяти и других используемых ресурсов дает пользователям исчерпывающую информацию о состоянии облака.
- Гипервизоры - CloudStack совместим с множеством гипервизоров, включая VMware vSphere, KVM, Citrix XenServer и Xen Cloud Platform (XCP).
- Надежность. CloudStack - это высокодоступное и масштабируемое решение IaaS, в котором отказ одного компонента не может вызвать сбой в работе кластера или всего облака (Apache 2012). Это позволяет управлять обслуживанием серверов без простоев и снижает нагрузку на управление крупномасштабным облачным развертыванием.
- Поддержка ОС - CloudStack поддерживает Linux для сервера управления и базовых вычислительных узлов. В зависимости от используемого гипервизора CloudStack поддерживает широкий спектр гостевых операционных систем, включая Windows, Linux и различные версии Berkeley Software Distributions (BSD).
- Стоимость. Сам CloudStack является бесплатным программным обеспечением под лицензией Apache. Однако при использовании коммерческого гипервизора могут возникнуть затраты.

### 13.2.3 Основные пользователи и общие комментарии к решению CloudStack

- Список клиентов. CloudStack используется компаниями для предоставления широкого спектра облачных услуг, от общедоступных и частных облачных услуг до гибридных облачных сервисов. Многие ведущие телекоммуникационные компании выбирают Citrix Cloud Platform на базе Apache CloudStack в качестве основы для своих облачных сервисов следующего поколения.

---

<sup>1</sup> См. Apache CloudStack Features на <http://incubator.apache.org/cloudstack/software/features.html>.

<sup>2</sup> См. Apache CloudStack на <http://en.wikipedia.org/wiki/CloudStack>.

<sup>3</sup> См. Understanding Apache CloudStack на <http://incubator.apache.org/cloudstack/software.html>.

## 228 Чэнь Сюй, Чжипенг Гуй, Цзин Ли, Кай Лю, Цюньин Хуани и Мира Бамбакус

К этим компаниям относятся: British Telecom (BT),<sup>1</sup> China Telecom,<sup>2</sup> EVRY,<sup>3</sup> IDC Frontier,<sup>4</sup> KDDI,<sup>5</sup> Korea Telecom (KT),<sup>6</sup> Nippon Telegraph and Telephone (NTT),<sup>7</sup> и Slovak Telecom.<sup>8</sup> Другие компании включают в себя Tata communications,<sup>9</sup> Исследовательский центр Nokia,<sup>10</sup> и Logicworks.<sup>11</sup>

- Завершение - CloudStack был впервые выпущен в 2010 году, и с тех пор он был значительно улучшен. Впервые он был выпущен под Стандартной общественной лицензией GNU версии 3 (GPLv3). Когда CloudStack был подарен Apache Software Foundation (ASF) компанией Citrix в апреле 2012 года, лицензия была изменена на лицензию Apache, версия 2. Теперь CloudStack - это проект инкубатора Apache, версия 4.1.1.12
- Отзывы сообщества - CloudStack является одним из лидеров в области облачных решений с открытым исходным кодом и предоставляет широкие функциональные возможности и удобный веб-интерфейс. Превращение в подпроект Apache Software Foundation резко увеличило скорость разработки CloudStack и укрепило его лидерство в облачных решениях с открытым исходным кодом.
- Сложность установки - установка CloudStack проста. Apache предоставляет вики-сайт с подробными инструкциями, различными форумами и множеством документов для разных типов пользователей (например, обычных пользователей, администраторов и разработчиков).

### 13.3 euCalyPtus

Eucalyptus - это решение с открытым исходным кодом для облачной обработки данных IaaS. Облачные сервисы IaaS на основе Eucalyptus предоставляют потребителям возможность запускать и контролировать экземпляры виртуальных машин, развернутых на различных физических ресурсах (Nurmi et al. 2009).

#### 13.3.1 Архитектура

На рисунке 13.3 показана высокоуровневая архитектура Eucalyptus. Eucalyptus предоставляет решение IaaS для создания частных или гибридных облаков.

<sup>1</sup> См. bt на <http://www.bt.com/>.

<sup>2</sup> См. China Telecom на <http://en.chinatelecom.com.cn/>.

<sup>3</sup> См. Evry на <http://www.evry.com/>.

<sup>4</sup> См. IDC Frontier на <http://www.idcf.jp/english>.

<sup>5</sup> См. KDDI на <http://www.kddi.com/english/index.html>.

<sup>6</sup> См. KT на <http://www.kt.com/eng/main.jsp>.

<sup>7</sup> См. NTT Communications на <http://www.ntt.com>.

<sup>8</sup> См. Telekom на <http://www.telekom.sk/>.

<sup>9</sup> См. Tata на <http://instacompute.com/>.

<sup>10</sup> См. Nokia Research на <http://research.nokia.com/>.

<sup>11</sup> См. Logicworks на <http://www.logicworks.net/>.

<sup>12</sup> См. What is CloudStack на <http://incubator.apache.org/cloudstack/>.

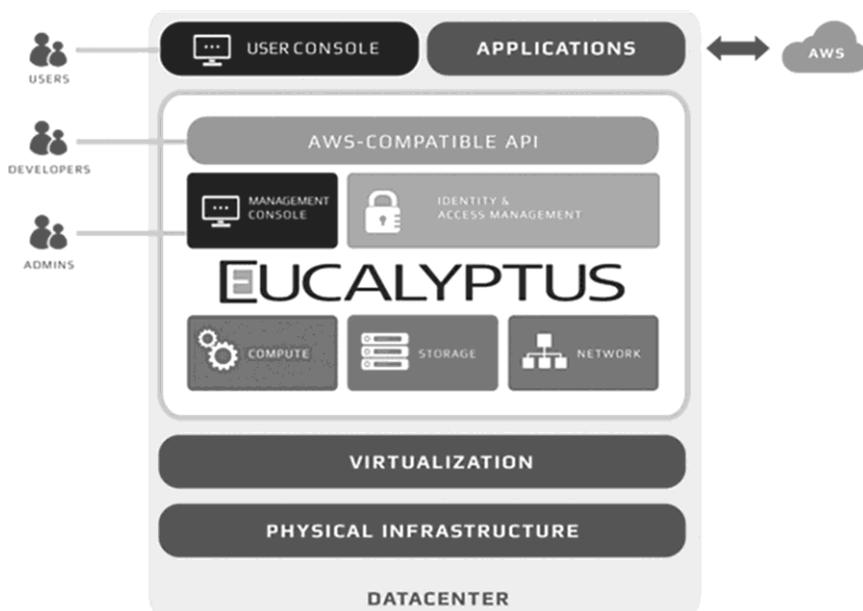


Рисунок 13.3 (См. цветную вставку) Архитектура эвкалипта. (См. Eucalyptus на [http://en.wikipedia.org/wiki/File:Eucalyptus\\_Platform\\_Architecture\\_Feb\\_February\\_2013.jpg](http://en.wikipedia.org/wiki/File:Eucalyptus_Platform_Architecture_Feb_February_2013.jpg).)

Путем виртуализации физических машин в центре обработки данных поставщики облачных услуг могут предоставлять потребителям облачных услуг совокупность виртуализированных аппаратных ресурсов компьютера, включая вычисления, сеть и хранилище. Потребители могут получить доступ к облаку с помощью инструментов командной строки (*euca2ools*)<sup>1</sup> или через веб-панель управления, такую как *Hybridfox*.<sup>2</sup> Eucalyptus также поддерживает AWS-совместимый API поверх Eucalyptus, чтобы потребители могли общаться с AWS.

### 13.3.2 Общие характеристики eucalyptus

- Масштабируемость - Eucalyptus поддерживает масштабируемость, начиная с Eucalyptus 2.0, на двух уровнях: масштабируемость клиентских транзакций и масштабируемость внутренних ресурсов (Eucalyptus 2.0 2013).
- Облачная модель. Eucalyptus предоставляет доступ к коллекциям виртуализированных компьютерных аппаратных ресурсов для вычислений, сети и хранения, чтобы предоставлять облачные сервисы IaaS. Пользователи могут собрать свой собственный виртуальный кластер, в котором они могут устанавливать, поддерживать и выполнять свой собственный стек приложений с помощью Eucalyptus.

<sup>1</sup>См. *Euca2ools* на <http://www.eucalyptus.com/download/euca2ools>.

<sup>2</sup>См. *Hybridfox* на <http://code.google.com/p/hybridfox/>.

## 230 Чэнь Сюй, Чжипенг Гуй, Цзин Ли, Кай Лю, Цюньин Хуан и Мира Бамбакус

- Совместимость. Eucalyptus совместим с облачными сервисами Amazon AWS, поэтому пользователи могут повторно использовать существующие инструменты и скрипты AWS. Eucalyptus обеспечивает совместимость с рядом функций AWS: Amazon EC2, Amazon EBS, AMI, Amazon S3 и Amazon IAM.
- Развертывание и интерфейс - Eucalyptus поддерживает API Amazon AWS для EC2 и S3. Eucalyptools - это поддерживаемые Eucalyptus инструменты командной строки для взаимодействия с веб-сервисами, совместимые с облачными сервисами Eucalyptus и Amazon.
- Гипервизоры - Eucalyptus совместим с гипервизорами Xen, KVM и VMware.
- Надежность - начиная с версии 3, Eucalyptus можно развернуть с высокой доступностью. Eucalyptus 3 повысил надежность облака IaaS с помощью механизмов автоматического переключения при отказе и восстановления после сбоя.
- Поддержка ОС - Eucalyptus 3.2 поддерживает Windows Server 2003 и 2008, Windows 7 и все современные дистрибутивы Linux, такие как Red Hat, CentOS, Ubuntu, Fedora и Debian.
- Стоимость - пользователи могут выбирать между бесплатным Eucalyptus Cloud с открытым исходным кодом и платным Eucalyptus Enterprise Cloud.

### 13.3.3 Основные пользователи и общие комментарии к Eucalyptus

- Список клиентов - ИТ-организации по всему миру используют облака Eucalyptus из-за их гибкости, эластичности и масштабируемости, необходимых для очень требовательных приложений (Наши клиенты, 2013). К этим организациям относятся: Puma,<sup>1</sup> NASA, Национальная служба охраны ресурсов (NRCS), Университет Джорджа Мейсона, Корнельский университет и другие.
- Завершение - с тех пор, как Eucalyptus выпустила свою первую версию в 2008 году, было выпущено шесть версий. Последняя версия 3.2 была выпущена 19 декабря 2012 года.
- Отзывы сообщества - Eucalyptus поддерживает сообщество с помощью вики-сайта и списков рассылки. Cloud Computing World Series поместила его в тройку лидеров в категории «Лучшие облачные сервисы» в 2012 году<sup>2</sup>.
- Сложность установки - Eucalyptus легко установить и развернуть. Он предоставляет исчерпывающую документацию,<sup>3</sup> отзывчивое сообщество<sup>4</sup> и списки рассылки<sup>5</sup> для потребителей и разработчиков.

<sup>1</sup> См. Puma на <http://www.puma.com/>.

<sup>2</sup> См. Cloud World Series на <http://www.cloudcomputinglive.com/awards/2012shortlist.html>.

<sup>3</sup> См. Eucalyptus на <http://www.eucalyptus.com/docs>.

<sup>4</sup> См. Eucalyptus Engage на <https://engage.eucalyptus.com/>.

<sup>5</sup> См. GitHub Eucalyptus Mailing Lists на <http://lists.eucalyptus.com/cgi-bin/mailman/listinfo>.

## 13.4 OpenNebula

OpenNebula разработана как решение для построения центров обработки данных корпоративного уровня и облаков IaaS. Его модульная архитектура позволяет разработчикам облаков настраивать и реализовывать широкий спектр облачных сервисов, сохраняя при этом высокий уровень стабильности и качества.

### 13.4.1 Архитектура

OpenNebula - одно из самых популярных решений для облачной обработки данных с открытым исходным кодом, которое полностью поддерживает частные, общедоступные и гибридные облачные платформы (Milošević et al. 2011). Начиная с версии 2.0, внутренние компоненты OpenNebula включают основной модуль, набор подключаемых драйверов и несколько инструментов (рисунок 13.4а). Базовый модуль управляет виртуальными ресурсами, такими как виртуальные машины, виртуальные сети, виртуальное хранилище и образы, и контролирует их. Он также обрабатывает клиентские запросы и вызывает соответствующие драйверы для выполнения операций с ресурсами. Драйверы служат адаптерами для взаимодействия с промежуточным программным обеспечением. Основные функции доступны конечным пользователям с помощью набора инструментов и API. В то время как более ранние версии OpenNebula действительно страдали недостатком масштабируемости, что делало его пригодным только для облаков малого и среднего размера (Sempolinski and Thain 2010), его последняя версия, версия 3.8, превратилась в архитектуру с высокой степенью масштабируемости (рисунок 13.4b). Интерфейсы были значительно расширены и включают REST API (например, EC2-Query API), OpenNebula Cloud API (OCA) и API для собственных драйверов. В частности, он имеет встроенную поддержку подключения к AWS.

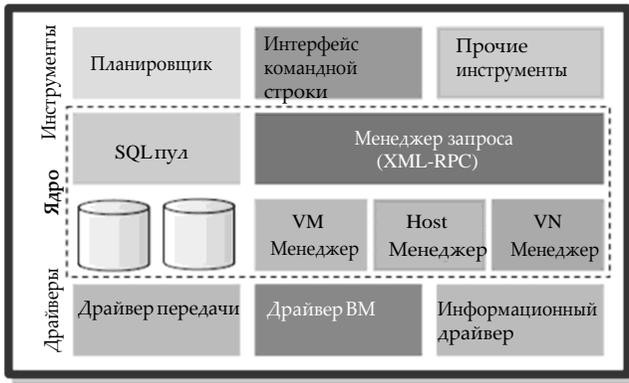
### 13.4.2 Общие характеристики opennebula

Согласно документации по OpenNebula <sup>1</sup>, основные особенности OpenNebula можно резюмировать следующим образом:

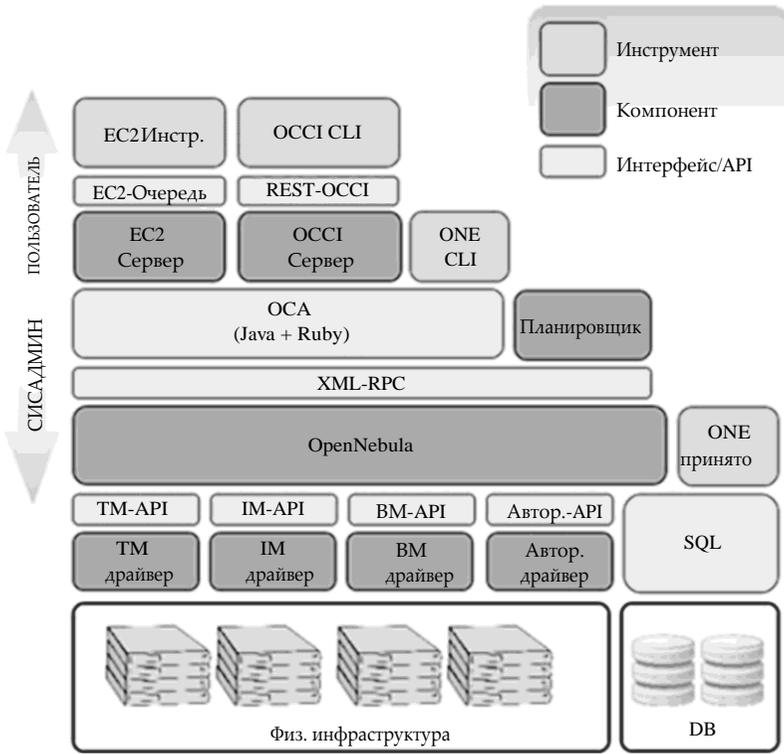
- Масштабируемость - OpenNebula использовалась для создания крупномасштабной инфраструктуры, а также для высокомасштабируемых баз данных. Драйверы виртуализации можно настроить для достижения максимальной масштабируемости.
- Облачная модель - OpenNebula, разработанная для поддержки облаков IaaS с использованием существующей инфраструктуры, обычно не предъявляет особых требований к инфраструктуре.
- Совместимость - OpenNebula можно развернуть в существующей инфраструктуре и интегрировать с различными облачными сервисами. Чтобы обновить систему, администраторы могут следовать руководству по совместимости в документации.<sup>2</sup>

<sup>1</sup> См. OpenNebula 3.8 Guides на <http://opennebula.org/documentation:rel3.8>.

<sup>2</sup> См. Apache CloudStack на <http://incubator.apache.org/cloudstack/docs/en-US/index.html>.



(a)



(b)

Рисунок 13.4 (См. цветную вставку) (a) Внутренняя архитектура OpenNebula (см. Архитектуру OpenNebula на <http://opennebula.org/documentation:archives:rel2.0:architecture>) и (b) интерфейсы (см. Масштабируемую архитектуру OpenNebula по адресу <http://opennebula.org/documentation:rel3.8:introapis>).

- Развертывание и интерфейсы - интерфейсы доступны для потребителей и поставщиков облака. Облачные провайдеры могут разрабатывать индивидуальные инструменты с облачными интерфейсами. Потребители могут использовать интерфейс командной строки (CLI) или веб-портал SunStone для выполнения большинства операций, особенно управления ресурсами. Кроме того, последняя версия предоставляет интерфейсы для облачных провайдеров, таких как Amazon.
- Гипервизоры - OpenNebula поддерживает три основных гипервизора: KVM, Xen и VMware. Поскольку драйвер гипервизора можно переключать между разными гипервизорами, он обеспечивает решение для платформы с несколькими гипервизорами.
- Надежность. Система OpenNebula разработала специальный модуль проверки качества OpenNebula QA, чтобы гарантировать качество каждого выпуска. Даже если решение для управления случайно выйдет из строя, виртуальные ресурсы, расположенные на отдельных машинах, можно восстановить.
- Поддержка ОС - поддерживаются основные версии Linux и Windows. Образы ОС, созданные гипервизорами с OpenNebula или без нее, можно использовать в качестве шаблонов ОС.
- Стоимость. Хотя OpenNebula является полностью бесплатным решением, его корпоративная версия OpenNebulaPro распространяется по годовой подписке. Коммерческая поддержка - это дополнительная услуга, предоставляемая C12G Labs<sup>1</sup> по индивидуальной цене.

### 13.4.3 Основные пользователи и общие комментарии к OpenNebula

- Заказчики - OpenNebula широко используется в различных сценариях приложений в качестве готового решения для предприятий. Можно выделить четыре основные категории пользователей: (1) частные корпоративные облака и виртуализация центров обработки данных (среди известных клиентов - IBM Global Business Services и Национальная центральная библиотека Флоренции); (2) провайдеры хостинга и облачных услуг, такие как China Mobile; (3) научное сообщество высокопроизводительных вычислений (HPC), такое как НАСА; (4) Облачные интеграторы и разработчики продуктов, такие как консалтинговая компания KPMG.
- Завершение. С момента своего создания в 2008 году проект OpenNebula выпустил более 20 версий, включая стабильные и бета-версии по состоянию на 2012 год. Последняя версия, 3.8, была выпущена 22 октября 2012 года. Тем временем разработчики тесно сотрудничали с сообществом пользователей OpenNebula для решения требований пользователей, а также технических проблем.
- Отзывы сообщества. Отзывы сообщества также отражают основные ценности OpenNebula: «Открытость», «Совершенство», «Сотрудничество» и «Инновации».

---

<sup>1</sup>См. C12G Labs на <http://c12g.com/>.

## 234 Чэнь Сюй, Чжипенг Гуй, Цзин Ли, Кай Лю, Цюньин Хуани и Мира Бамбакус

В частности, поставщики и потребители облачных сервисов подчеркивают открытость, которая позволяет в высокой степени настраивать приложения. Дополнительные отзывы можно найти на сайте OpenNebula Testimonials по адресу <http://opennebula.org/users:testimonials>.

- Сложности установки - OpenNebula полагается на другое программное обеспечение с открытым исходным кодом, такое как KVM, и не предлагает официально интегрированного установщика. Перед установкой программного обеспечения пользователи должны установить все зависимые пакеты и соответствующим образом настроить среду. Следовательно, установка относительно проста для опытных установщиков, знакомых с Linux и методами виртуализации. В отличие от опытных пользователей, начинающие пользователи могут столкнуться с крутой кривой обучения.

### 13.5 nImBus

Первоначальная разработка системы Nimbus была ориентирована на вычислительные потребности научных исследований. Nimbus включает в себя набор инструментов для создания облачного сервиса IaaS.

#### 13.5.1 Архитектура

Nimbus включает в себя платформу Nimbus и инфраструктуру Nimbus. Платформа Nimbus позволяет пользователям использовать облака IaaS, а инфраструктура Nimbus поддерживает создание облаков IaaS. На рисунке 13.5 показаны компоненты инфраструктуры Nimbus. Когда потребитель облака подписывается на сервис Nimbus, создается виртуальное рабочее пространство.

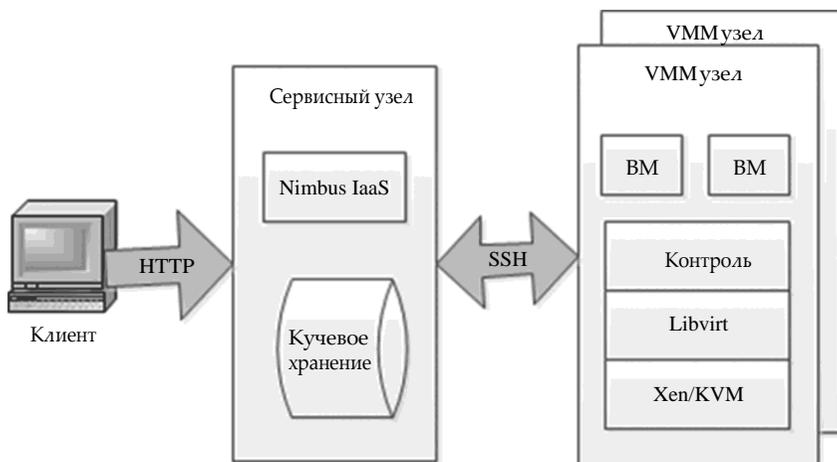


Рисунок 13.5 Структура Nimbus IaaS.

Рабочее пространство состоит из внешнего интерфейса, службы рабочего пространства, серверной части и рабочего пространства виртуальной машины. Рабочая область виртуальной машины развертывается на узле монитора виртуальных машин (VMM), который является физическим узлом. После завершения развертывания потребители могут получить доступ к узлу облачной службы через интерфейс HTTP. Cumulus - это важнейший компонент Nimbus, служащий интерфейсом для репозитория образов Nimbus VM. Любой образ виртуальной машины должен быть загружен в репозиторий Cumulus перед загрузкой.

### 13.5.2 Общие характеристики nimbus

- Масштабируемость - модуль Cumulus Redirection Nimbus управляет масштабируемостью. Этот модуль отслеживает загруженность службы. Поскольку Cumulus совместим с Amazon S3 REST API, его можно настроить для работы в виде набора реплицированных хостов для поддержки горизонтальной масштабируемости.
- Облачная модель. Nimbus - это решение VIM для IaaS.
- Совместимость. Хранилище Cumulus расширяет Amazon S3 REST API и совместимо с S3.
- Развертывание и интерфейс. Пользователи напрямую взаимодействуют с виртуальными машинами в пуле узлов почти так же, как с физической машиной. Nimbus публикует информацию о виртуальной машине, такую как IP-адрес каждой виртуальной машины, чтобы пользователи могли знать информацию о каждой виртуальной машине. Пользователи развертывают приложения в облаках Nimbus, используя конфигурацию облачного набора, которая включает хостинг службы менеджера и репозиторий образов.
- Гипервизоры - Nimbus поддерживает KVM и Xen.
- Надежность - для достижения того же уровня надежности, что и S3, конфигурация аппаратного обеспечения Cumulus должна быть на том же уровне, что и S3. Другими словами, надежность Nimbus частично зависит от аппаратной инфраструктуры, на которой построен Cumulus.
- Поддержка ОС - Nimbus поддерживает различные дистрибутивы Linux.
- Стоимость. Nimbus - это решение с открытым исходным кодом. Следовательно, программное обеспечение бесплатное.

### 13.5.3 Основные пользователи и общие комментарии о Nimbus

- Заказчики - Nimbus был разработан для научных задач. Он использовался для проекта STAR в Брукхейвенской национальной лаборатории, ING, JPMorgan и Toyota.
- Завершение - Nimbus стартовал как исследовательский проект в 2003 году. Последняя стабильная версия - 2.10.<sup>1</sup>

<sup>1</sup> См. Nimbus на <http://www.nimbusproject.org/downloads/>.

## 236 Чэнь Сюй, Чжипенг Гуй, Цзин Ли, Кай Лю, Цюньин Хуан и Мира Бамбакус

---

- Обратная связь от сообщества - Nimbus может удовлетворить потребности научных компьютерных сообществ (Mangtani and Bhingarkar 2012).
- Сложность установки. Установка Nimbus начинается с установки набора программного обеспечения на один сервисный узел под учетной записью без полномочий root. Облачный клиент Nimbus должен быть установлен на дополнительных VMM.

### 13.6 РАССМОТРЕНИЕ ЭТАЛОННЫХ РАСЧЕТОВ С ОТКРЫТЫМ ИСХОДНЫМ КОДОМ

В таблицах 13.1 и 13.2 приведено сравнение четырех решений IaaS, основанное на соображениях, которые следует учитывать при сравнении различных вариантов. В таблице 13.1 профили архитектурных характеристик, а в таблице 13.2 описаны настройки пользователя.

### 13.7 ЗАКЛЮЧЕНИЕ

Сообщество разработчиков ПО с открытым исходным кодом предоставило зрелые решения для создания облачных сервисов IaaS. CloudStack, Eucalyptus, Nimbus и OpenNebula - четыре решения, которые были успешно внедрены ИТ-предприятиями, а также академическими исследовательскими институтами. По сравнению с облачными сервисами коммерческих поставщиков решения с открытым исходным кодом предлагают дополнительную гибкость и лучший контроль над ресурсами. Для клиентов, которые заботятся о хранении и безопасности, решения для облачных вычислений с открытым исходным кодом предоставляют варианты для создания частных облаков. Каждое из четырех представленных выше решений IaaS с открытым исходным кодом может интегрировать различные типы оборудования и программного обеспечения. Из-за различий в управлении базовым оборудованием (например, ЦП, ОЗУ и т. д.) Эксплуатационная производительность каждого решения может быть разной. Следовательно, производительность платформы должна быть протестирована для поддержки различных вычислительных задач (от интенсивных данных, вычислений и обмена данными до одновременных интенсивных задач), и будет рассмотрено в Главе 14.

### 13.8 КРАТКОЕ ИЗЛОЖЕНИЕ

В данной главе представлены четыре решения облачной обработки данных с открытым исходным кодом IaaS в целом (раздел 13.1), а также подробные факты для: CloudStack (раздел 13.2), Eucalyptus (раздел 13.3), OpenNebula (раздел 13.4) и Nimbus (раздел 13.5). На основе деталей каждого решения было проведено сравнение четырех решений по их системным характеристикам (раздел 13.6). Содержание этой главы закладывает основу для следующей главы, в которой оцениваются четыре решения.

Таблица 13.1 Сравнение четырех решений IaaS

	<i>Nimbus</i>	<i>Eucalyptus</i>	<i>CloudStack</i>	<i>OpenNebula</i>
<b>Масштабируемость (Возможность управления)</b>	Масштабируемый	Масштабируемый	Масштабируемый	Масштабируемый
Совместимость (Совместимость с другими облачными сервисами, например, AWS)	EC2, S3 Совместимый	Поддержка EC2, S3 (через Amazon EC2)	CloudBridge)	Мультиплатформенный Динам. развертывание
Развертывание (возможность добавления и удаления физических узлов)	Динам. развертывание	Динам. развертывание	Динам. развертывание	
Интерфейс установки (Интерфейс для установки пакета)	CLI	CLI	CLI	CLI
Параметры образа диска (создание образа с нуля или повторное объединение)	Зависит от конфигураций	Параметры, устанавливаемые поставщиками услуг	Пользователи могут загружать и управлять шаблонами и изображениями.	В частном облаке большинство опций libvirt оставлены открытыми
Гипервизоры (поддерживаемое программное обеспечение для виртуализации)	KVM, Xen	VMWare (закрытый исходный код, Xen,	VMware, KVM, Oracle VM, Kronos, Citrix XenServer, и Citrix Xen Cloud Platform	Xen, VMWare, KVM
Веб-интерфейс (протоколы подключения)	RESTful HTTP API	KVM) Веб-сервис	Ajax- и jQuery-основанные Web GUI,	
Безопасный доступ к консоли VNC, поддержка API от Amazon и VMware	XML-RPC API, EC2 Query, OGF OCCL, и vCloud API			
Структура (способ организации компонентов системы, таких как клиент, агенты и менеджер ресурсов)	Легкие компоненты	Модульный	Модульный	Модульный

(Продолжение)

Таблица 13.1 (Продолжение) Сравнение четырех решений IaaS

	<i>Nimbus</i>	<i>Eucalyptus</i>	<i>CloudStack</i>	<i>OpenNebula</i>
Надежность (механизмы отказоустойчивости в облачной реализации)	Зависит от базового оборудования	Да, с использованием механизмов автоматического переключения при отказе и восстановления после отказа	Да, несколько серверов управления и серверов баз данных, а также функции живой миграции виртуальных машин	Откат хоста и ВМ
Встроенный мониторинг (ВМ, мониторинг облачной инфраструктуры)	Cloudinit.d	Состояние ВМ	Синхронизация ВМ и высокая доступность	Основные переменные
Управление брандмауэром (управление доступом и брандмауэром виртуальных машин)	Инструмент фильтрации iptables для межсетевого экрана	Группы безопасности	Группа пользователей для управления брандмауэром экземпляра	Используйте программный маршрутизатор для управления брандмауэром
Балансировщик нагрузки (предоставление услуг по балансировке нагрузки)	Нет, можно дополнить балансировщиками нагрузки, такими как Zeus, Zeus	Балансировщик нагрузки	Использование другой службы балансировки трафика для балансировки нагрузки	Да, используется встроенный плагин балансировки нагрузки

Таблица 13.2 Определенные действия пользователя и интерфейсы

	<i>Nimbus</i>	<i>Eucalyptus</i>	<i>CloudStack</i>	<i>OpenNebula</i>
<i>Разрешения пользователя</i>	Админ, пользователь	Админ, пользователь	Админ, пользователь	Админ, пользователь группы
<i>Подключение экземпляра</i>	SSH	SSH ключи, RDP для экземпляров	VNC, SSH	VNC, SSH
<i>Контроль облачного источника</i>	CLI and основанный на веб Eucal2ools, CMS	CLI, и EC2 API	Поддержка CLI и API из Amazon и VMware	CLI, Java API, основанный на веб CMS

### 13.9 ПРОБЛЕМЫ

1. Какие общие функции должен обеспечивать диспетчер виртуальной инфраструктуры (VIM)?
2. Каковы общие характеристики CloudStack?
3. Каковы общие характеристики эвкалипта?
4. Каковы общие характеристики OpenNebula?
5. Опишите основные различия между Nimbus и другими VIM, представленными в настоящей главе.
6. Используйте пример, чтобы проиллюстрировать, какой из трех VIM, CloudStack, Eucalyptus и OpenNebula, будет наиболее подходящим решением для примера.

### СПИСОК ЛИТЕРАТУРЫ

- Apache. 2012. CloudStack Installation Guide. [http://incubator.apache.org/cloudstack/docs/en-US/Apache\\_CloudStack/4.0.0-incubating/pdf/Installation\\_Guide/Apache\\_CloudStack-4.0.0-incubating-Installation\\_Guide-en-US.pdf](http://incubator.apache.org/cloudstack/docs/en-US/Apache_CloudStack/4.0.0-incubating/pdf/Installation_Guide/Apache_CloudStack-4.0.0-incubating-Installation_Guide-en-US.pdf) (accessed March 25, 2013).
- Armbrust, M., A. Fox, R. Griffith et al. 2010. A view of cloud computing. *Communications of the ACM* 53, no. 4: 50–58.
- Eucalyptus. 2013. Eucalyptus 2.0 Delivers High Scalability and Flexibility for Private Cloud Computing. <http://www.eucalyptus.com/news/08-25-2010> (accessed August 8, 2013).
- Huang, Q., J. Xia, C. Yang et al. 2012. An experimental study of open-source cloud platforms for dust storm forecasting. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, pp. 534–537. ACM.
- Mangtani, N. and S. Bhingarkar. 2012. The appraisal and judgment of Nimbus, Open Nebula and Eucalyptus. *International Journal of Computational Biology* 3, no. 1: 44–47.

240 Чэнь Сюй, Чжипенг Гуй, Цзин Ли, Кай Лю, Цюньин Хуани Мира Бамбакус

---

- Milojčić, D., I. M. Llorente, and R. S. Montero. 2011. OpenNebula: A cloud management tool. *Internet Computing IEEE* 15, no. 2: 11–14.
- Nurmi, D., R. Wolski, C. Grzegorzczak et al. 2009. The Eucalyptus open-source cloud-computing system. In *Cluster Computing and the Grid. CCGRID'09. 9th IEEE/ACM International Symposium*, pp. 124–131.
- Rimal, B. P., A. Jukan, D. Katsaros, and Y. Goeleven. 2011. Architectural requirements for cloud computing systems: An enterprise cloud approach. *Journal of Grid Computing* 9, no. 1: 3–26.
- Sempolinski, P. and D. Thain. 2010. A comparison and critique of Eucalyptus, OpenNebula and Nimbus. In *Cloud Computing Technology and Science (CloudCom), IEEE 2nd International Conference*, pp. 417–426.
- Sotomayor, B., R. S. Montero, I. M. Llorente, and I. Foster. 2009. Virtual infrastructure management in private and hybrid clouds. *Internet Computing, IEEE* 13, no. 5: 14–22.

## Глава 14

---

# Как проверить готовность решений для облачной обработки данных с открытым исходным кодом

*Цюньбин Хуан, Цзичжэ Ся, Мин Сун, Кай Лю, Цзин Ли, Чжипенг Гуй, Чэнь Сю и Чаовой Ян*

---

В данной главе рассказывается, как протестировать и сравнить готовность трех решений для облачной обработки данных с открытым исходным кодом, включая Eucalyptus, CloudStack и OpenNebula.

### 14.1 ВВЕДЕНИЕ

Цель этой главы - проверить готовность решений с открытым исходным кодом к поддержке характеристик облачных вычислений NIST и требований к вычислениям для наук о Земле. Матрица тестирования включает в себя производительность, удобство использования, стоимость и характеристики решений. Изначально для тестирования выбираются три облачных решения: Eucalyptus, Cloudstack и OpenNebula. Факторы, определяющие выбор решений с открытым исходным кодом, включают: (i) зрелость решений, (ii) доступность версии с открытым исходным кодом, (iii) стоимость корпоративной версии, (iv) размер сообщества пользователей и (v) любые общедоступные показатели качества (доля рынка, отзывы клиентов и др.). Таблицы 14.1 и 14.2 суммируют наши выводы, включая основных клиентов облачных решений, отзывы сообщества о зрелости решений, сложности установки, доступности версии с открытым исходным кодом и стоимости. Также перечислены поддерживаемые операционные системы (ОС) для физических и виртуальных машин (ВМ).

Основное внимание в этой главе уделяется практическим знаниям, а подробное представление и обсуждение результатов тестирования предоставлено Huang et al. (2013).

### 14.2 СРЕДА ТЕСТИРОВАНИЯ

Кластер, созданный и размещенный Центром интеллектуальных пространственных вычислений для науки о воде и энергии (CISC) (рис. 14.1), предназначен для создания среды тестирования. Каждое решение для облачных вычислений с открытым исходным кодом устанавливается и настраивается в одной среде аппаратной инфраструктуры.

Таблица 14.1 Заказчики и оценки программных решений

Решение	Список Заказчиков	Зрелость (Лет в бмзнесе)	Обратная связь	Сложности установки
Eucalyptus DoD,	Cloudera, NASA, HP, SONY, FDA, Puma, USDA	Первая версия вышла в мае 2008 г.	Хотя программное обеспечение легко установить, потребители облачных сред сообщают о проблемах конфигурации сети.	Установка относительно проста для начинающего установщика; Конфигурация сети сложна
OpenNebula	IBM, Hexagrid, CloudWeavers, CERN	Первый публичный релиз был в 2008 году.	В проект добавлена новая стабильная версия	Крутая кривая обучения для начинающего установщика
CloudStack	Zynga, Edmunds.com, Nokia Research Center	CloudStack был выпущен в мае 2010 г.	Хорошо задокументированы; Могут возникнуть проблемы с созданием экземпляров с Xen	Легко установить, потому что пакеты подготовлены для руководства установщиками в процессе

Таблица 14.2 Общая характеристика выбранных решений

Решение	Версия с откр. исх. кодом	Стоимость (Издание компании)	ОС, поддерживающая облако	Поддерживаемые образы ОС
Eucalyptus	Да	Лицензирование на основе количества физических хостов	Red Hat Enterprise; CentOS; openSUSE-11; Debian; Fedora; Ubuntu	Windows; Major Linux OS распределение
OpenNebula	Да	Бесплатно	Ubuntu и CentOS	Windows; Major Linux OS распределение
CloudStack Enterprise;	Да	Бесплатно	Red Hat CentOS 5; CentOS 6.0; CentOS 6.1; Ubuntu 10.04; Fedora 14	Windows; Major Linux OS распределение

## Как проверить готовность решений для облачной обработки данных с открытым исходным кодом 243

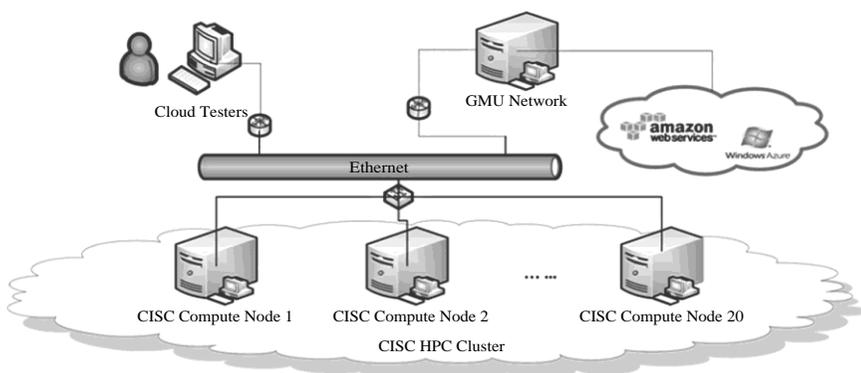


Рисунок 14.1 Среда тестирования решений с открытым исходным кодом.

Двадцать шесть вычислительных узлов используются из кластера CISC и подключены через локальные сети (LAN) (со скоростью 1 Гбит / с). Каждый узел имеет объем памяти 16 ГБ и двухъядерный центральный процессор (ЦП) с частотой 2,33 ГГц.

На рисунке 14.2 показана концептуальная архитектура решений с открытым исходным кодом (Eucalyptus, CloudStack и OpenNebula). Нижний уровень имеет одинаковую конфигурацию оборудования и настройку сетевого подключения для всех трех решений. Уровень операционной системы (ОС) включает CentOS 5.7, CentOS 5.8 и CentOS 6.0 в качестве ОС. Xen (Barham et al. 2003) или виртуальная машина на основе ядра (KVM) (2010), которые служат уровнем виртуализации, управляемым программным обеспечением с открытым исходным кодом. Каждое решение устанавливается на пяти вычислительных узлах, один из которых служит главным узлом, а другие четыре - вычислительными (или подчиненными) узлами. Прикладной уровень включает в себя два приложения для изучения геологии для оценки производительности приложений.

На каждом из трех решений тестируется пять конфигураций виртуальных машин (таблица 14.3). Каждая конфигурация виртуальной машины имеет разный размер оперативной памяти (ОЗУ) и количество ядер виртуального процессора..

### 14.3 ТЕСТЫ ОБЛАЧНОЙ РАБОТЫ

Проверены характеристики распределения и выпуска единичных вычислительных ресурсов. Тестируются запуск и выпуск вычислительных ресурсов, включая виртуальную машину и объем хранилища. Каждый тест повторяется пять раз для каждой конфигурации виртуальной машины. Регистрируются и сравниваются временные затраты на запуск, приостановку, перезапуск и удаление виртуальной машины с разными виртуальными вычислительными ресурсами.

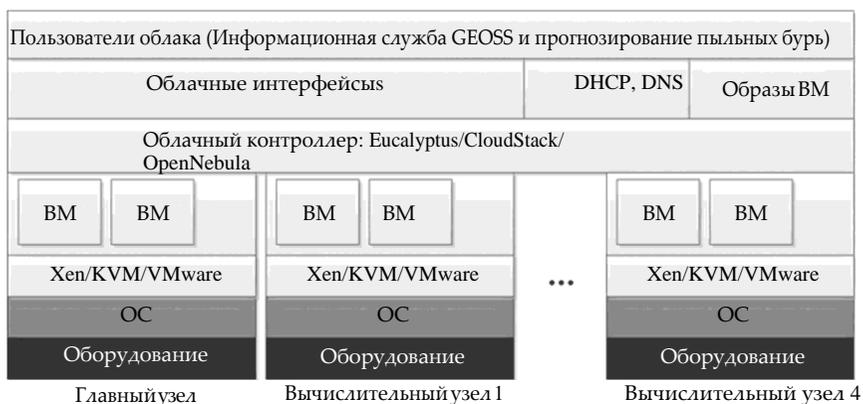


Рисунок 14.2 Концептуальная облачная архитектура.

Таблица 14.3 Конфигурации облачных виртуальных машин

Тип экземпляра	Архитектура	ОС	RAM (Гб)	Виртуальные ядра	Диск (Гб)
Маленький	64-бит	CentOS 6.0	1.7	1	10
Средний	64-бит	CentOS 6.0	1.7	2	10
Большой	64-бит	CentOS 6.0	7.5	2	10
Хбольшой	64-бит	CentOS 6.0	12	4	10
ХХбольшой	64-бит	CentOS 6.0	12	8	10

## 14.4 ИСПЫТАНИЯ ВИРТУАЛЬНЫХ ВЫЧИСЛИТЕЛЬНЫХ РЕСУРСОВ

### 14.4.1 Краткое введение

Тесты виртуальных вычислительных ресурсов включают в себя тестирование ЦП виртуальной машины, памяти, жесткого диска и производительности сети. В Таблице 14.4 представлены выбранные наборы инструментов для тестирования. UBench тестирует производительность процессора. Bonnie ++ - это инструмент тестирования производительности с открытым исходным кодом для оценки производительности жесткого диска и файловой системы. CacheBench, компонент пакета LLCbench (Mucci 2012), оценивает производительность иерархии памяти компьютерной системы. Iperf (Tirumala et al. 2012) измеряет пропускную способность протокола управления передачей (TCP) и протокола дейтаграмм пользователя (UDP).

В следующих разделах представлены дизайн теста виртуальных вычислительных ресурсов, рабочий процесс и интерпретация результатов.

Как проверить готовность решений для облачной обработки данных с открытым исходным кодом 245

Таблица 14.4 Общие бенчмаркеры для тестирования производительности виртуальных машин (ВМ)

	Бенчмаркер	Испытуемый объект
Виртуальные вычисления Ресурс	UBench <sup>a</sup>	ЦПУ
	Bonnie++ <sup>b</sup>	Жесткий диск
	CacheBench	Сеть производительности иерархии памяти
	h <sup>c</sup> Iperf <sup>d</sup>	MySQL
Общее приложение	sql-bench	Java
	SPECjvm2008 <sup>e</sup>	Информационная служба GEOSS
Геонаучное приложение Spatial Web portal	Spatial Web (database, tomcat) MPI application	Модель пыльной бури

<sup>a</sup> Из UBench, 2012,

<http://www.tucows.com/preview/69604/Ubench>. <sup>b</sup> Из Coker, 2008, <http://www.coker.com.au/bonnie++/>.

<sup>c</sup> См. CacheBench на <http://icl.cs.utk.edu/projects/lcbench/cachebench.html>.

<sup>d</sup> Из Tirumala et al., 2012,

<http://iperf.sourceforge.net/>. <sup>e</sup> См. Spec на <http://www.spec.org/download.html>.

## 14.4.2 Проектирование теста

- Производительность процессора. Инструмент тестирования UBench (UBench 2012) на основе C++ используется для измерения производительности процессора. Ubench выполняет серию вычислений с плавающей запятой и целых чисел в течение 3 минут на тест.
- Производительность ввода / вывода. Bonnie++ (Coker 2008) используется для эталонного тестирования производительности ввода-вывода путем тестирования операций файловой системы, чтобы получить результат объема работы, выполненной в секунду, и процентного соотношения процессорного времени, затраченного на операции. Для тестирования ввода-вывода используются два индикатора: (1) результаты производительности с более высокими числами лучше и (2) использование ЦП с более низким процентом лучше. Файловые операции, которые выполняет Bonnie++: (1) скорость чтения и записи данных, (2) количество операций поиска в секунду и (3) количество операций с метаданными файла в секунду.
- Производительность иерархии памяти. LLCbench (Low Level Characterization Benchmark Suite) (Mucci 2012) был создан путем объединения MPBench, 1 CacheBench1,2 и BLASBench3 в один тестовый пакет. CacheBench можно использовать для проверки производительности иерархии памяти. Каждый процесс тестирования включает в себя чтение кеша, операции изменения и записи, а также один процесс тестирования одновременно для каждой виртуальной машины. Пропускная способность передачи записывается.
- Производительность сети. Производительность TCP / UDP измеряется с помощью Iperf.4 Проверяется пропускная способность внутренних и межмашинных сетевых подключений виртуальной машины. Для внутренних сетевых подключений виртуальная машина для отправки пакетов данных и виртуальная машина для приема пакетов данных расположены на одной физической машине.

<sup>1</sup> См. MPBench на <http://icl.cs.utk.edu/projects/lcbench/mpbench.html>.

<sup>2</sup> См. CacheBench1 на <http://icl.cs.utk.edu/projects/lcbench/cachebench.html>.

<sup>3</sup> См. BLASBench на <http://icl.cs.utk.edu/projects/lcbench/blasbench.html>.

<sup>4</sup> См. Iperf на <https://code.google.com/p/iperf/>.

Для межмашинного сценария две виртуальные машины подготовлены на двух разных физических машинах.

### 14.4.3 Последовательность выполняемых действий теста

Общие этапы тестирования виртуальных вычислительных ресурсов (рис. 14.3) для каждой облачной платформы включают:

*Шаг 1. Создание образа с различными программными пакетами для тестирования виртуальных вычислительных ресурсов, установленными для каждого решения с открытым исходным кодом. Чтобы создать образ, необходимо запустить виртуальную машину с конфигурацией, указанной в Таблице 14.3, и ее можно дополнительно настроить. Программный пакет для тестирования виртуальных вычислительных ресурсов должен быть установлен на базовой виртуальной машине. Установка и настройка этих тестеров, включая UBench, Bonnie ++, CacheBench и Iperf, описаны ниже.*

- ЦПУ сравнительный анализ. Следующие команды показывают установку UBench на виртуальной машине Linux.

```
PROMPT>> wget ftp://ftp.uwsg.indiana.edu/pub/FreeBSD/  
ports/distfiles/ubench-0.32.tar.gz  
PROMPT>> tar xvfz ubench-0.32.tar.gz  
PROMPT>> cd ubench-0.32  
PROMPT>> ./configure  
PROMPT>> patch -p0 < ubench-patch.txt  
PROMPT>> make
```

Программный пакет UBench имеет небольшую ошибку и требует использования файла патча ubench-patch.txt, выполнив команду `patch -p0 <ubench-patch.txt`, чтобы исправить ошибку. В Приложении 14.1 показано содержимое файла патча..

- Тестирование ввода-вывода. Следующие команды используются для загрузки и установки Bonnie ++.

```
PROMPT>> wget http://pkgs.repoforge.org/bonnie++/  
bonnie++-1.03e-1.e15.rf.x86_64.rpm
```

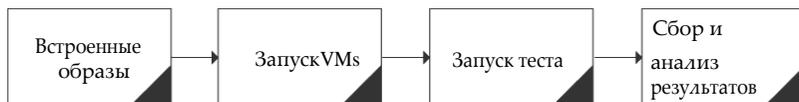


Рисунок 14.3 Рабочий процесс тестирования виртуальных вычислительных ресурсов.

## Как проверить готовность решений для облачной обработки данных с открытым исходным кодом 247

---

```
PROMPT>> yum install gcc-c++ ## If g++ is not install
PROMPT>> rpm -i bonnie+-1.03e-1.el5.rf.x86_64.rpm
PROMPT>> /usr/sbin/bonnie++ ## Run Bonnie++ without
options to check the usage
```

- Тестирование памяти. Программа CacheBench может определять параметры производительности подсистемы архитектуры памяти. Он интегрирован в LLCbench. Следующие команды показывают загрузку LLCbench и установку CacheBench на виртуальную машину Linux.

```
PROMPT>> wget http://icl.cs.utk.edu/projects/
llcbench/llcbench.tar.gz
PROMPT>> tar xvfz llcbench.tar.gz
PROMPT>> cd llcbench
PROMPT>> make linux-lam
PROMPT>> cd cachebench
PROMPT>> make
```

- Тестирование сети. Iperf можно легко установить в любой системе UNIX / Linux или Microsoft Windows. Следующие команды показывают, как установить Iperf в системе Linux CentOS.

```
PROMPT>> http://superb-dca2.dl.sourceforge.net/
project/iperf/iperf/2.0.4%20source/iperf-
2.0.4.tar.gz
PROMPT>> yum install gcc-c++
PROMPT>> tar xvfz iperf-2.0.4.tar.gz
PROMPT>> cd iperf-2.0.4
PROMPT>> ./configure
PROMPT>> make && make install
```

После установки этих инструментов тестирования можно создать образ виртуальной машины для каждого решения.

*Шаг 2.* Запуск различных типов виртуальных машин - пять типов виртуальных машин (таблица 14.3) могут быть запущены из образов, созданных на шаге 1 для каждого решения.

*Шаг 3.* Выполнение теста - на этом этапе необходимо создать сценарий для последовательного запуска каждого тестера. Как правило, каждый тестер повторяется трижды с ожиданием 60 секунд между каждым тестом. Подробная информация о скрипте для каждого тестера:

- Тестирование процессора. Скрипты будут повторять тест UBench три раза с ожиданием 60 секунд между каждым тестом.
- ```
testNum=3
i=0
while [ $i -lt $ testNum];
do
```

```
/Directory_for_ubench/ubench >> ubench.log ## run the
test
sleep 60 ## wait for 60 seconds
let i=i+1
done
```

- Тестирование ввода-вывода. Тестирование Bonnie ++ проводилось в два этапа: (1) определение наименьшего размера файла, которое делает недействительным кэш ввода-вывода на основе памяти, и (2) использование размера файла, превышающего пороговый размер, для проверки реального ввода-вывода. производительность (рекомендуется использовать размер файла, в два раза превышающий размер ОЗУ). Размеры файлов варьируются от 1024 КБ до 40 ГБ. ВданномпримереиспользуетсявиртуальнаямашинасразмеромОЗУ4ГБ, поэтому размер файла для записи на жесткий диск должен быть 8 ГБ. Чтобы запустить тестирование Bonnie ++, строка.

```
/Directory_for_ ubench/>> ubench.log should be
changed to the fol-lowing commands.
/usr/sbin/bonnie++ -d /mnt -s 8g -m centos_server
-f -b -u root >> bonnie.log
```

Фактический процесс тестирования для каждого времени занимает от минут до часов в зависимости от размера записываемого файла.

- Тестирование памяти. Чтобы запустить тестирование CacheBench, строка /Directory\_for\_ubench/ubench > > ubench. log должна быть заменена на следующие команды.

```
/Directory_for_ cachebench/cachebench -b -x1 -m9 -e1
> cachebench.log ## run the test
```

- Тестирование сети. Тестирование сети Iperf отличается от двух предыдущих тестов, поскольку для каждого теста требуется две виртуальные машины: одна виртуальная машина должна быть настроена как клиент, а другая - как сервер. Показан серверный сценарий для запуска процесса прослушвателя сервера Iperf для клиентского соединения.  
iperf -s -f m >& iperf.server.log.out &

Данная команда запускает прослушватель Iperf на порте по умолчанию 5001. Iperf также необходимо создать клиентский процесс на клиентской виртуальной машине для подключения к прослушивателю. В плане эксперимента тест Iperf будет повторяться три раза и ждать 60 секунд перед следующим циклом тестов для каждого типа VM. Таким образом, клиентские скрипты для проверки производительности сети между серверным приемником и клиентской виртуальной машиной выглядят как следующие (IP-адрес прослушвателя сервера 199.26.254.162).

## Как проверить готовность решений для облачной обработки данных с открытым исходным кодом 249

---

```
echo "Begin iperf test"
iperf -c 199.26.254.162 -P 4 -f m -w 256K -t 60 >>
  iperf.client.log.out
sleep 60 ## wait for 60 seconds
iperf -c 199.26.254.162 -P 4 -f m -w 256K -t 60 >>
  iperf.client.log.out
sleep 60
iperf -c 199.26.254.162 -P 4 -f m -w 256K -t 60 >>
  iperf.client.log.out
echo "Three iperf tests finished"
```

*Шаг 4:* Соберите и проанализируйте результаты - по завершении тестирования проверьте файлы журнала (например, ubench.log) на предмет результатов тестирования процессора.

### 14.4.4 Анализ результатов теста

(4) • Результаты тестирования процессора. Конечный результат - это оценка, отражающая производительность процессора. На рис. 14.4 показан раздел выходных данных UBench (файл `iperf.client.log.out`), включая результаты тестов ЦП и ОЗУ. Можно заметить, что виртуальная машина получает 2161708 баллов за тест процессора.

• Результаты тестирования ввода-вывода. По умолчанию выходные данные представлены в формате значений, разделенных запятыми (CSV), включая результаты скорости чтения и записи данных, количество поисков в секунду и количество операций с метаданными файла в секунду. Статистика использования ЦП также включается в отчет о сравнительном тестировании для следующих операций: (1) создание файлов в последовательном порядке, (2) указание файлов в последовательном порядке, (3) удаление файлов в последовательном порядке, (4) создание файлов в случайном порядке, (5) указание файлов в случайном порядке и (6) удаление файлов в случайном порядке. На рисунке 14.5 показан образец отчета о сравнительном анализе.

Такие результаты, как «Последовательный вывод» в методах «Per Chr», извлекаются из отчетов и вводятся в таблицу (Таблица 14.5).

```
Unix Benchmark Utility v.0.3
Copyright (C) July, 1999 PhysTech, Inc.
Author: Sergei Viznyuk <sv@phystech.com>
http://www.phystech.com/download/ubench.html
Linux 2.6.18-194.el5xen #1 SMP Fri Apr 2 15:34:40 EDT 2010 x86_64
Ubench CPU: 2161708
Ubench MEM: 340333
-----
Ubench AVG: 1251020
```

Рисунок 14.4 Вывод UBench.

```

Version 1.03      -----Sequential Output----- --Sequential Input-
--Random-
                -Per Chr- --Block-- -Rewrite- -Per Chr- --Block--
--Seeks--
Machine          Size K/sec %CP K/sec %CP K/sec %CP K/sec %CP K/sec %CP
/sec %CP
test             8G 15945 23 16881 3 7192 0 39739 34 42668 0
87.5 0
                -----Sequential Create----- -----Random
Create-----
                -Create-- --Read--- -Delete-- -Create-- --Read---
-Delete--
files /sec %CP /sec %CP /sec %CP /sec %CP /sec %CP
/sec %CP
                16 21034 97 +++++ +++ +++++ +++ 15791 73 +++++ +++
+++++ +++
test,8G,15945,23,16881,3,7192,0,39739,34,42668,0,87.5,0,16,21034,97,
+++++,+++,+++++,+++,15791,73,+++++,+++,+++++,+++
    
```

Рисунок 14.5 Образец отчета об эталонном тестировании Bonnie ++.

Таблица 14.5 Пример результатов теста Bonnie ++ на платформе Eucalyptus Cloud

|                         | Пор. выход (КБ / с) |       |            | Пор. вход (КБ / с) |       | Ранд. ввод<br>(искать / с) |
|-------------------------|---------------------|-------|------------|--------------------|-------|----------------------------|
|                         | Признак             | Блок  | Переписать | Признак            | Блок  |                            |
| Eucalyptus<br>Маленький | 365                 | 31561 | 16924      | 411                | 42135 | 91.9                       |
| Средний                 | 310                 | 30155 | 16484      | 415                | 41901 | 135.5                      |
| Большой                 | 312                 | 30663 | 18457      | 458                | 45461 | 44                         |
| Хбольшой                | 308                 | 27945 | 16440      | 477                | 40535 | 29.5                       |
| ХХбольшой               | 308                 | 29027 | 17025      | 378                | 41949 | 29.7                       |

double RMW Cache Test

| C Size | Nanosec | MB/sec   | % Chnge |
|--------|---------|----------|---------|
| 256    | 0.49    | 15473.64 | 1.00    |
| 384    | 0.46    | 16448.61 | 0.94    |
| 512    | 0.46    | 16559.79 | 0.99    |

Рисунок 14.6 Вывод CacheBench.

Результаты показывают производительность ввода-вывода тестируемого экземпляра. Одинаковый процесс тестирования выполняется на всех платформах с разными конфигурациями вычислений. Когда все результаты нанесены на график, они отражают различия в производительности между экземплярами.

- Результаты тестирования памяти. CacheBench измеряет время доступа к памяти в наносекундах и пропускную способность в МБ/с. На рисунке 14.6 показан результат одного теста в файле журнала (cachebench.log).
- Результаты сравнительного тестирования сети. Iperf записывает пропускную способность в мегабитах в секунду (Мбит/сек). На рисунке 14.7 показан результат теста производительности сети для виртуальной машины с IP-адресом 199.26.254.162 (IP-адрес клиента, отправляющего запрос, - 199.26.254.163).

## Как проверить готовность решений для облачной обработки данных с открытым исходным кодом 251

```
-----  
Client connecting to 199.26.254.162, TCP port 5001  
TCP window size: 0.25 MByte (WARNING: requested 0.25 MByte)  
-----  
[ 4] local 199.26.254.163 port 34738 connected with 199.26.254.162 port 5001  
[ 5] local 199.26.254.163 port 34739 connected with 199.26.254.162 port 5001  
[ 3] local 199.26.254.163 port 34737 connected with 199.26.254.162 port 5001  
[ 6] local 199.26.254.163 port 34740 connected with 199.26.254.162 port 5001  
[ ID] Interval      Transfer      Bandwidth  
[ 3] 0.0-30.0 sec   825 MBytes    230 Mbits/sec  
[ ID] Interval      Transfer      Bandwidth  
[ 4] 0.0-30.0 sec   850 MBytes    238 Mbits/sec  
[ ID] Interval      Transfer      Bandwidth  
[ 5] 0.0-30.0 sec   851 MBytes    238 Mbits/sec  
[ ID] Interval      Transfer      Bandwidth  
[ 6] 0.0-30.0 sec   845 MBytes    236 Mbits/sec  
[SUM] 0.0-30.0 sec  3370 MBytes   941 Mbits/sec  
-----
```

Рисунок 14.7 Вывод Iperf.

## 14.5 ИСПЫТАНИЯ ОБЩЕГО ПРИЛОЖЕНИЯ

### 14.5.1 Краткое введение в аспекты теста

Этот общий тест приложения включает в себя тест производительности работы Java и базы данных (таблица 14.4). SPECjvm2008 - это эталонный инструмент для измерения производительности среды выполнения Java (JRE). MySQL-Bench используется для проверки производительности работы базы данных на примере MySQL.

### 14.5.2 Проектирование теста

- Производительность JRE - SPECjvm2008 содержит несколько реальных приложений и тестов для оценки производительности функциональности Java. SPECjvm2008 фокусируется на тесте производительности JRE на одной машине. Другими словами, он отражает общую производительность процессора и памяти, в то время как операции ввода-вывода и сеть на разных машинах слабо зависят от результатов тестирования SPECjvm2008. Метрики, предоставляемые SPECjvm2008, - это количество операций в минуту (операций / м), которое включает в себя:

- SPECjvm2008 base ops / m - общая производительность при полностью соблюденном базовом прогоне.
- SPECjvm2008 peak ops / m - пиковая производительность при запуске приложения.

Этот эксперимент включает установку и запуск SPECjvm2008 на виртуальных машинах, запущенных разными облачными провайдерами, для тестирования производительности JRE в разных облаках.

- Производительность MySQL - MySQL-Bench используется для проверки производительности базы данных MySQL. Он может тестировать различные типы операций с базой данных, такие как создание таблиц, вставка и удаление записей, а также запросы к таблицам, что обеспечивает всестороннее тестирование возможностей базы данных MySQL, поддерживаемой различными решениями. Метрики, предоставляемые MySQL-Bench, регистрируют общие временные затраты на выполнение всех операций с базой данных, определенных инструментом тестирования в секундах.

### 14.5.3 Последовательность выполнения теста

Подробные шаги для создания окончательных отчетов по тестированию производительности JRE и MySQL-Bench аналогичны тесту виртуальных вычислительных ресурсов (рисунок 14.3):

*Шаг 1. Создавайте образы с помощью пакета программного обеспечения JRE для тестирования производительности или MySQL-Bench, установленного на каждой облачной платформе.*

- Установите и настройте JRE. Чтобы создать эталонный образ JRE для различных облачных платформ, необходимо запустить базовую виртуальную машину на каждой облачной платформе. JRE должен быть установлен на виртуальной машине. Тестер SPECjvm2008 также должен быть загружен и установлен на виртуальную машину с помощью следующей команды.

```
PROMPT>> wget http://spec.it.miami.edu/downloads/osg/  
java/SPECjvm2008_1_01_setup.jar  
PROMPT>> java -jar SPECjvm2008_1_01_setup.jar -i  
console
```

- Установите и настройте MySQL. Для создания образов с помощью средства тестирования MySQL необходимо запустить виртуальные машины на каждой платформе и проверить их на наличие версии MySQL. Если старая версия существует, удалите старую версию и установите последнюю версию с помощью следующих команд.

```
PROMPT>> which mysql  
PROMPT>> yum remove mysql-server  
PROMPT>> yum remove mysql  
PROMPT>> yum install mysql-server  
PROMPT>> yum install mysql  
PROMPT>> yum install mysql-devel
```

После правильной установки MySQL MySQL-Bench, инструмент для тестирования производительности, можно установить с помощью следующей команды.

```
PROMPT>> yum install mysql-bench  
PROMPT>> cd /usr/share/sql-bench
```

## Как проверить готовность решений для облачной обработки данных с открытым исходным кодом 253

*Шаг 2: Запускайте разные типы виртуальных машин в разных решениях - пять типов виртуальных машин (таблица 14.3) могут быть запущены из образов, созданных на шаге 1 для каждого решения.*

*Шаг 3: Проведите тест.*

- Проверьте производительность JRE. Чтобы запустить тест SPECjvm2008 три раза, командную строку /Directory\_for\_ubench/ubench>>ubench.log необходимо изменить на следующую команду.

```
/Directory_for_SPECjvm2008/run-specjvm.sh startup.  
helloworld -ikv >> SPECjvm2008.log
```

- Проверьте производительность MySQL. Чтобы повторить полный тест MySQL-Bench три раза, командную строку /Directory\_for\_ubench/ubench >>ubench.log следует изменить на следующую команду.

```
/usr/share/sql-bench/perl run-all-tests
```

```
--user=ACCOUNT -password=MYROOTPASSWORD >> sql-  
bench.log
```

ACCOUNT и MYROOTPASSWORD следует заменить на настоящую учетную запись пользователя и пароль для доступа к базе данных MySQL.

Шаг 4: Соберите и проанализируйте результаты - файл журнала (SPECjvm2008.log) должен содержать результаты трехкратного тестирования для теста JRE. Файл журнала sql-bench.log должен содержать результаты теста SQL-Bench.

### 14.5.4 Анализ результатов теста

- Производительность JRE. Сценарий тестирования JRE создает файл журнала (SPECjvm2008.log), который трижды записывает количество операций за одну секунду и подробную информацию о производительности. На рисунке 14.8 показан результат одного тестового прогона. Результат показывает, что виртуальная машина выполняет 141,51 операций в секунду.

#### SPECjvm2008 Base

n/a n/a

Oracle Corporation OpenJDK 64-Bit Server VM

Tested by: n/a

Test date: Wed Mar 20 21:34:55 EDT 2013

| Benchmark | ops/m  |
|-----------|--------|
| startup   | 141.51 |

Рисунок 14.8 Пример результата тестирования SPECjvm2008.

- MySQL. В файл журнала (sql-bench.log) записываются временные затраты на выполнение всех операций на тестовой виртуальной машине, на которой находится сценарий. На рис. 14.9 показаны результаты одного тестового прогона, записанные в файл журнала. Первое числовое значение указывает затраты времени в секундах. Следовательно, на этой машине тест стоит 823 секунды.

## 14.6 ТЕСТ НА ГОТОВНОСТЬ ОБЛАКА GEOSS CLEARINGHOUSE

### 14.6.1 Требования к вычислительным средствам Информационного центра

Информационная служба GEOSS (CLH) (см. Главу 8 и Главу 12), геопространственный портал с массовым одновременным доступом, используется для проверки одновременной производительности облачных платформ.

### 14.6.2 Проектирование тестов, рабочий процесс и анализ

Матричный тест, проведенный в главе 12 для коммерческого облачного тестирования, также может быть использован в тесте облачной платформы с открытым исходным кодом после пересмотра. На рис. 14.10 показан план тестирования трех решений: (1) CLH и JMeter установлены на трех виртуальных машинах с облаками с открытым исходным кодом и традиционным локальным сервером; (2) Четыре решения (три облачные платформы плюс одна традиционная вычислительная платформа) связаны через Интернет; (3) CLH можно протестировать, отправив одновременные запросы от одного решения (на стороне клиента) к трем другим решениям (на стороне сервера) с помощью JMeter.

```
Start sql-bench:
TOTALS                823.00   227.04   54.25  281.29 3425950
```

Рисунок 14.9 Пример результата теста.

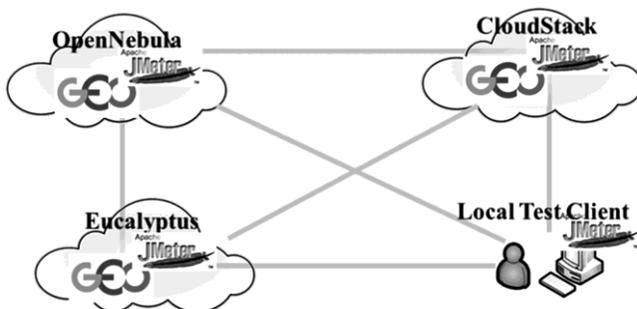


Рисунок 14.10 Матричный тест GEOSS Clearinghouse.

Как проверить готовность решений для облачной обработки данных с открытым исходным кодом 255

---

Виртуальные машины трех решений могут служить в качестве серверов, а виртуальные машины каждого решения будут последовательно выступать в качестве тестового клиента. Наконец, в ходе теста можно получить одновременные результаты производительности этих трех решений.

Глава 12, раздел 12.3 включает подробный дизайн, рабочий процесс и анализ тестов, касающихся матрицы CLN, баланса нагрузки в облаке и тестов масштабируемости.

## **14.7 ТЕСТ НА ГОТОВНОСТЬ ОБЛАКА К ПРОГНОЗИРОВАНИЮ ПЫЛЕВОЙ БУРЫ**

### **14.7.1 Требования к вычислениям для прогнозирования пыльных бурь**

Как было сказано в главах 10 и 12, прогнозирование пыльных бурь требует как вычислений, так и обмена данными. Таким образом, он используется для проверки готовности решений с открытым исходным кодом для поддержки приложений, требующих интенсивных вычислений.

### **14.7.2 Проектирование теста**

В главе 12, раздел 12.5, сравнивались и тестировались различия платформ EC2 и Nebula, а также оценивалась способность гиперпоточности. В этом разделе представлены планы нескольких групп экспериментов для тестирования облачных решений с открытым исходным кодом. Модель пыльной бури развернута в локальном кластере HPC и трех различных решениях с открытым исходным кодом.

Три группы экспериментов разработаны следующим образом:

1. HPC против облаков. Виртуальный кластер, созданный из виртуальных машин, сравнивается с традиционными кластерами для количественной оценки накладных расходов на преобразование физической инфраструктуры в облака. Результат этого эксперимента покажет, насколько хорошо решения поддерживают крупномасштабные научные вычисления. В рамках этого эксперимента сравнивается различное количество виртуализированных (от одной до четырех виртуальных машин) и не виртуализированных вычислительных ресурсов, чтобы исследовать влияние виртуализированной вычислительной мощности, хранилища и сети.
2. Сравнение решений с открытым исходным кодом. В этом эксперименте проверяется способность различных решений поддерживать приложения, интенсивно использующие вычисления и связь, с разным количеством виртуальных машин на физических машинах и тремя облачными решениями соответственно. Результаты показывают производительность этих облачных решений для поддержки научных вычислений.
3. Технология виртуализации. В этом эксперименте сравнивается производительность такого же количества вычислительных ресурсов, виртуализированных с помощью KVM и Xen.

### 14.7.3 Последовательность выполнения тестов

Рабочий процесс тестирования для каждой группы аналогичен главе 12, раздел 12.5.3 (рисунок 12.7): (1) создание образов для каждой облачной платформы; (2) написание сценария тестовой оболочки для запуска модели и записи производительности; (3) запуск одинакового количества виртуальных машин на каждой платформе с использованием образов; (4) настройка модельных сред; (5) запуск тестов и сбор результатов. Некоторые конкретные конфигурации виртуальных машин для трех групповых тестов, разработанных в Разделе 14.7.2, описаны ниже:

1. НРСпротивоблаков. Тестызапускаютодноитоже моделированиена одной, двух, трех и четырех виртуальных машинах XXLarge с использованием разныхномеров процессовдля трех решений.
2. Сравнение облачного промежуточного программного обеспечения. Тест добавляет еще одну группу экспериментов с двумя большими VM на разных решениях. Интегрируя результаты предыдущих групп производительности, можно сравнивать и тестировать возможности различных решений для управления виртуальными машинами.
3. Технология виртуализации. Этот тест запускает моделирование на одной, двух, трех и четырех виртуальных машинах, созданных в одном решении (например, OpenNebula или Eucalyptus), но с двумя разными популярными технологиями виртуализации с открытым исходным кодом KVM и Xen (глава 3, раздел 3.4). Цель - проверить различия технологий KVM и Xen.

### 14.7.4 Анализ результатов теста

В разделе 12.5.4 рассказывается, как интерпретировать результаты теста прогноза пыльной бури.

## 14.8 КРАТКОЕ ИЗЛОЖЕНИЕ

В этой главе рассказывается, как тестировать решения для облачных вычислений с открытым исходным кодом. В разделе 14.1 представлены общие сведения, сообщество и отзывы пользователей о трех решениях с открытым исходным кодом, включая Eucalyptus, CloudStack и OpenNebula. В разделе 14.2 представлены тестовые среды. В разделе 14.3 представлены процедуры тестирования производительности управления облачными ресурсами (например, виртуальной машиной) с помощью решений. В разделе 14.4 рассказывается, как тестировать виртуальные вычислительные ресурсы с помощью различных инструментов. В этой главе также представлены стратегии тестирования для общих приложений (раздел 14.5) и приложений для геолого-геофизических исследований, включая Информационный центр GEOSS (раздел 14.6) и моделирование пыльной бури (раздел 14.7). Метод тестирования, опыт и сценарии, предоставленные через веб-сайт в Интернете, также могут использоваться для тестирования других решений с открытым исходным кодом. Huang et al.(2013) предоставляет подробные результаты тестирования трех решений.

Как проверить готовность решений для облачной обработки данных с открытым исходным кодом 257

---

## 14.9 ПРОБЛЕМЫ

1. Какие аспекты следует учитывать при тестировании виртуальных вычислительных ресурсов?
2. Перечислите инструменты, которые можно использовать для тестирования виртуальных вычислительных ресурсов.
3. Каков общий рабочий процесс тестирования виртуальных облачных вычислительных ресурсов?
4. Как вы тестируете решения с открытым исходным кодом с общими приложениями?
5. Какие аспекты следует учитывать при тестировании решений с открытым исходным кодом?
6. Как вы проверяете способность решения с открытым исходным кодом поддерживать параллельную интенсивность?
7. Как вы проверяете способность решения с открытым исходным кодом поддерживать интенсивность вычислений?
8. Прочтите отчет о результатах (Huang et al. 2013) и опишите результаты в 500 словах. Обсудите динамику результатов, то есть то, как результаты могут измениться.

### ПРИЛОЖЕНИЕ 14.1 ФАЙЛ ПАТЧА UBENCH (UBENCH-PATCH.TXT)<sup>1</sup>

```
-- membench.c.old 2008-05-18 21:21:02.000000000 +0800
+++ membench.c 2008-05-18 21:38:20.000000000 +0800
@@ -23,6 +23,7 @@
#define MAX_CHILDEN 128
#define MUFEN 1024
+#include <bits/time.h>
#include <sys/types.h>
#include <sys/times.h>
#include <stdio.h>
-- cpubench.c.old 2008-05-18 21:21:06.000000000 +0800
+++ cpubench.c 2008-05-18 21:38:16.000000000 +0800
@@ -22,6 +22,7 @@
#define CPUREFSCORE 50190
#define MAX_CHILDEN 128

+#include <bits/time.h>
#include <sys/types.h>
#include <sys/times.h>
#include <stdio.h>
-- configure.old 2008-05-18 21:39:14.000000000 +0800
+++ configure 2008-05-18 21:39:42.000000000 +0800
```

<sup>1</sup>Примечание: Спасибо анонимному участнику за публикацию этого патча в Интернете.

```
@@ -24,7 +24,7 @@
    i486)
cat <<! >> Makefile
CC = gcc
-CFLAGS = -O2 -m486 -Wall -malign-loops = 2 -malign-jumps =
      2 -malign-functions = 2 -fomit-frame-pointer
+CFLAGS = -O2 -m486 -Wall -falign-loops = 2 -falign-jumps =
      2 -falign-functions = 2 -fomit-frame-pointer
LDFLAGS = -s -lm
INCLUDES = -I.
```

```
@@ -33,7 +33,7 @@
    i586)
cat <<! >> Makefile
CC = gcc
-CFLAGS = -O2 -Wall -malign-loops = 2 -malign-jumps = 2
      -malign-functions = 2 -fomit-frame-pointer
+CFLAGS = -O2 -Wall -falign-loops = 2 -falign-jumps = 2
      -falign-functions = 2 -fomit-frame-pointer
LDFLAGS = -s -lm
INCLUDES = -I.
```

```
@@ -42,7 +42,7 @@
    i686)
cat <<! >> Makefile
CC = gcc
-CFLAGS = -O2 -Wall -malign-loops = 2 -malign-jumps = 2
      -malign-functions = 2 -fomit-frame-pointer
+CFLAGS = -O2 -Wall -falign-loops = 2 -falign-jumps = 2
      -falign-functions = 2 -fomit-frame-pointer
LDFLAGS = -s -lm
INCLUDES = -I.
```

```
@@ -51,7 +51,7 @@
    *)
cat <<! >> Makefile
CC = gcc
-CFLAGS = -O2 -Wall -malign-loops = 2 -malign-jumps = 2
      -malign-functions = 2
+CFLAGS = -O2 -Wall -falign-loops = 2 -falign-jumps = 2
      -falign-functions = 2
LDFLAGS = -s -lm
INCLUDES = -I.
```

## СПИСОК ЛИТЕРАТУРЫ

Barham, P., B. Dragovic, K. Fraser et al. 2003. Xen and the art of virtualization.  
*ACM SIGOPS Operating Systems Review* 37, no. 5: 164–177.

Как проверить готовность решений для облачной обработки данных с открытым исходным кодом 259

---

Huang, Q., C. Yang, K. Liu et al. 2013. Evaluating open-source cloud computing solutions for geosciences. *Computers & Geosciences* 59, 41–52.

KVM. 2010. Kernel-Based Virtual Machine. <http://www.linuxkvm.org> (accessed April 25, 2013).

Mucci, P. 2012. LLCbench (Low-Level Characterization Benchmarks). <http://icl.cs.utk.edu/projects/llcbench/> (accessed April 28, 2013).

UBench. 2012. Ubench 0.32. <http://www.tucows.com/preview/69604/Ubench> (accessed April 25, 2013).



## Глава 15

---

# Инициатива GeoCloud

*Дуг Неберт и Цюньин Хуан*

---

Инициатива Geospatial Cloud Sandbox (GeoCloud) была разработана в рамках деятельности Geospatial Platform<sup>1</sup> для развертывания и документирования геопространственных облачных сервисов на основе общих образов платформы как услуги (PaaS), которые совместно разрабатываются и быстро настраиваются несколькими агентствами. В этой главе представлены методы и уроки, извлеченные из проекта.

### 15.1 ВВЕДЕНИЕ

Благодаря гибкости, рентабельности и рентабельности облачных вычислений многие федеральные агентства внедряют облачные технологии для сокращения затрат и повышения эффективности федеральных ИТ-операций. В декабре 2011 года главный информационный директор (CIO) США Вивек Кундра объявил о политике «Облако прежде всего» для федеральных агентств, требующей, чтобы все агентства переместили хотя бы одну систему в хостинговую среду<sup>2</sup>. Министерство финансов США переместило Treasury.gov., SIGTARP.gov, MyMoney.gov, TIGTA.gov и IRS Oversight Board.treasury.gov в среду Amazon Elastic Cloud Computing (EC2) в рамках перехода федерального правительства к облачным сервисам.<sup>1</sup>

GeoCloud - это ежегодная инициатива по созданию прототипов, которую координирует Федеральный комитет по географическим данным (FGDC) в сотрудничестве с Управлением общих служб США (GSA) и Министерством здравоохранения и социальных служб. Он задуман как инкубатор для проверки возможности создания и, в конечном итоге, для продвижения концепции платформы сообщества GeoCloud для приложений агентства в безопасной облачной среде. С помощью этих стандартизованных виртуальных геопространственных серверов можно быстро перенести различные правительственные геопространственные приложения в облако.

---

<sup>1</sup> См. Geospatial Platform на <http://www.geoplatform.gov/> (operated by the U.S. Federal Geographic Committee).

<sup>2</sup> См. eWeek на <http://www.eweek.com/c/a/Cloud-Computing/US-Treasury-Moves-Public-Web-Sites-to-Amazon-EC2-Cloud-201782/>.

Кроме того, федеральные приложения могут обеспечить экономию как инфраструктуры, так и платформы:

- Экономия на инфраструктуре. Стоимость и трудозатраты на приобретение оборудования, расходы на конфигурации и операции оборудования, а также реализацию масштабируемости приложений могут быть значительно сокращены. Кроме того, поскольку эти приложения размещаются в облаке, можно снизить затраты на обслуживание сети и инфраструктуры хостинга.
- Экономия на платформе - время и усилия на создание системы сокращаются за счет создания, поддержки и совместного использования общего облачного пакета сообщества. Таким образом, агентства могут добиться более быстрого развертывания и рентабельной разработки, а также совместно использовать профили безопасности системы и документацию.

GeoCloud был разработан для тестирования и мониторинга внешних облачных данных и сервисных решений для геопространственной области, а также для поддержки деятельности межведомственной геопространственной платформы. Для достижения этой цели несколько проектов ежегодно номинируются федеральными агентствами в качестве существующих общедоступных служб геопространственных данных. Из первоначальных проектов-кандидатов набор общих требований к операционной системе и программному обеспечению был определен как базовый для пакетов платформы PaaS. Затем проекты развертываются с использованием этих общих пакетов платформы. В процессе разработки лучших практик документируется информация о стоимости и производительности. Типовые проекты требуют утверждения системы безопасности агентства для продолжения спонсируемых агентством операций в облаке после завершения хостинга, спонсируемого FGDC. Для набора проектов были определены две среды (платформы) развертывания, включая стек сервисов с открытым исходным кодом в Linux 64 и коммерческий стек сервисов в Windows. На рисунке 15.1 представлена подробная информация о целях, действиях и результатах GeoCloud.

## 15.2 Архитектура GeoCloud

Архитектурная структура GeoCloud включает три уровня (рисунок 15.2): (1) уровень инфраструктуры, (2) уровень PaaS и (3) уровень приложений. Уровень инфраструктуры объединяет ресурсы облачных вычислений (например, виртуальную машину [VM], хранилище и сеть) и базовые комплекты операционных систем для управления виртуальными машинами. Уровень PaaS находится в центре внимания развертываний GeoCloud и включает в себя программное обеспечение общих геопространственных сервисов, позволяющее настраивать пользовательский интерфейс проектами (приложениями). В связи с этим предлагаемые решения PaaS включают инструменты для программирования и настройки компонентов пользовательского интерфейса. Это сильно отличается от типичной среды «Программное обеспечение как услуга» (SaaS), где интерфейсы имеют ограниченный потенциал настройки и расширения.

## Инициатива GeoCloud 263



Рисунок 15.1 (См. цветную вставку) Цели, действия и результаты GeoCloud.

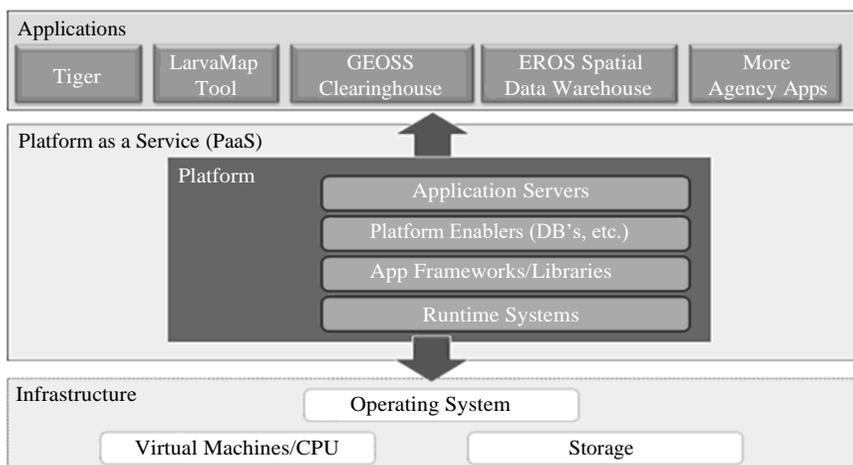


Рисунок 15.2 (См. цветную вставку) Архитектурный каркас GeoCloud.

Уровень PaaS состоит из следующих сервисов:

- Серверы приложений - серверы предоставляют среду развертывания для реальных бизнес-приложений с доступом к средствам реализации платформы, средам и средам выполнения, включая стандарты программирования и пользовательских интерфейсов, а также стандарты формата полезной нагрузки, одобренные FGDC.
- Активаторы платформы - механизмы реализации предоставляют возможности поддержки ядра для разработки, тестирования и развертывания кода, включая СУБД, каталог, очередь и служебную шину.

Реляционная база данных является наиболее распространенным примером активатора, который либо включен в PaaS, либо связан с экземпляром удаленного сервера базы данных.

- Платформы и библиотеки. Платформы предоставляют API-доступ к общим функциям и службам, на которые могут полагаться приложения. Библиотеки - это повторно используемые модули кода, которые можно вызывать непосредственно из приложения. Они могут сократить время и расходы, а также избавить разработчиков от необходимости создавать общие коды и модели поведения.
- Среда выполнения - обеспечивает поддержку выполнения для разработки и выполнения кода. Примеры включают Java, Python и Microsoft Common Language Runtime.

Проекты GeoCloud действуют как суррогат прикладного уровня, как показано на рисунке 15.2. В 2011 и 2012 годах следующие проекты завершили развертывание в облаке:

- Топологически интегрированное географическое кодирование и привязка (TIGER) / Line (Бюро переписи США 2010) - TIGER / Line - это пространственные данные, извлеченные из базы данных TIGER Бюро переписи населения, содержащие такие объекты, как дороги, железные дороги и реки, а также юридические и статистические данные. географические районы. Бюро переписи в настоящее время предлагает общественности наборы данных для загрузки с сервера данных переписи.
- Инструмент LarvaMap1 - это инструмент для моделирования траекторий движения личинок рыб, используемый Национальным управлением океанических и атмосферных исследований / Национальной службой морского рыболовства / Научным центром рыболовства Аляски. Инструмент предназначен для использования менеджерами ресурсов и учеными-рыболовами для моделирования распространения личинок рыб в различных условиях окружающей среды. Инструмент разработан на Java и размещен на Apache Tomcat, а ОС инструмента - Windows.
- Национальная инвентаризация водно-болотных угодий (Служба рыболовства и дикой природы США, 2012г.) - эта инвентаризация вместе с приложением «Картограф водно-болотных угодий» предоставляет данные и доступ к картам ко всем данным о водно-болотных угодьях в США.
- Информационная служба GEOSS (GEOSS 2011). Информационная служба GEOSS (глава 8) - это проект FGDC, GEO и NASA, который напрямую подключается к различным компонентам и службам GEOSS, а также собирает и выполняет поиск распределенных данных и служб с помощью взаимодействующих механизмов.

### 15.3 Деятельность GeoCloud

На рисунке 15.3 показан цикл основной деятельности GeoCloud, включающий: (1) создание прототипов платформ, (2) проверку с помощью приложений агентства и (3) документирование и публикацию.

<sup>1</sup> См. NOAA Fisheries на <http://www.afsc.noaa.gov/databases.htm>.

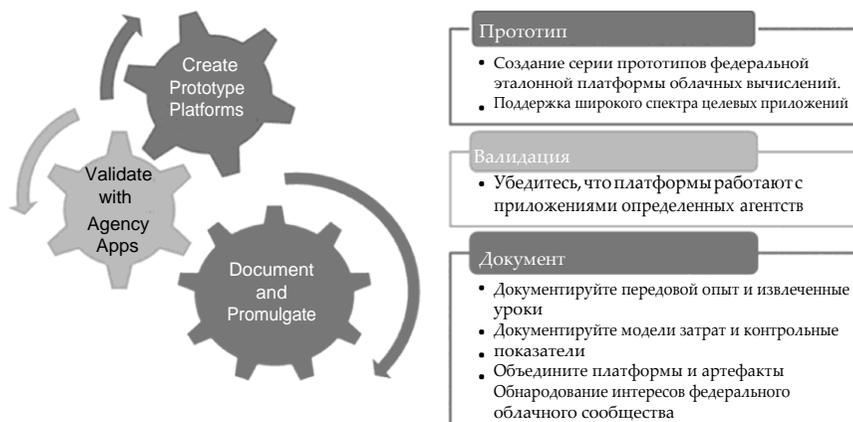


Рисунок 15.3 Цикл основной активности GeoCloud.

Таблица 15.1 Приложение и базовый программный пакет<sup>a</sup>

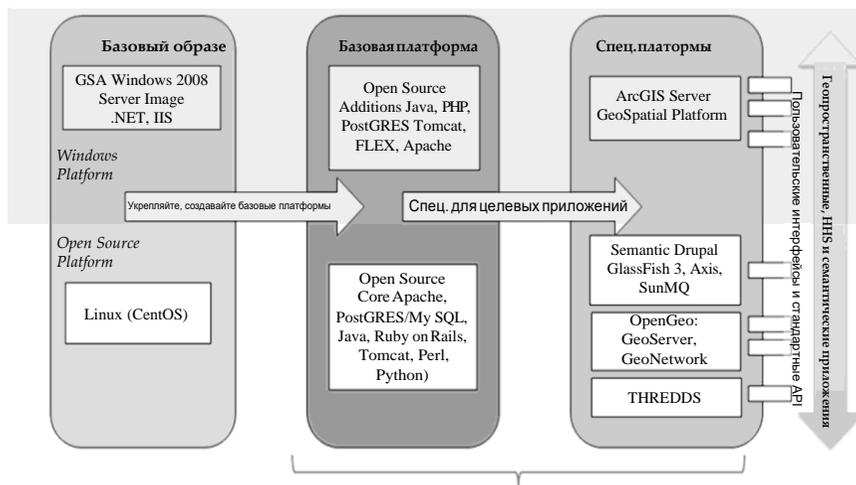
| Приложения                                       | Базовый программный пакет        |
|--------------------------------------------------|----------------------------------|
| Национальный реестр водно-болотных угодий (NWI)  | AWS, Windows 2008, ArcGIS сервер |
| Водно-болотные угодья                            |                                  |
| Картограф                                        | AWS, Linux 64 (CentOS)           |
| Каталог комплексных систем наблюдения за океаном | AWS, Linux 64 (CentOS),          |
| TIGER / Line Downloads U.S.Census                | GeoNetwork                       |
| и зритель (NOAA)                                 |                                  |
| NOAA ERDDAP                                      | AWS, Linux 64 (CentOS),          |
| EPA Озера и пруды                                | THREDDS                          |
| Вычислительный сервис отслеживания частиц в      | AWS, Windows 2008, ArcGIS        |
| облаке (NOAA) для дисперсии/ диффузии в воздухе  | сервер                           |
| или воде                                         | AWS, Linux 64 (CentOS)           |
| Информационная служба GEOCC                      |                                  |
| Приложение службы данных USDA FSA или NRCS       | AWS, Linux 64 (CentOS)           |
|                                                  | AWS, Linux 64 (CentOS)           |
|                                                  | AWS, Windows 2008, ArcGIS сервер |

<sup>a</sup> См. Federal Geographic Data Committee GeoCloud Platform на [http://semanticcommunity.info/@api/deki/files/10504/GeoCloud\\_Platform\\_Business\\_Use\\_Cases.pdf](http://semanticcommunity.info/@api/deki/files/10504/GeoCloud_Platform_Business_Use_Cases.pdf).

### 15.3.1 Создание платформы прототипа

Благодаря популярности и надежности Amazon Elastic Cloud Computing (EC2), она была выбрана в качестве основной общедоступной облачной вычислительной среды для различных размеров и количества виртуальных машин. Программные требования для платформ-кандидатов были продиктованы проектами, выдвинутыми федеральными агентствами. Стандарты продиктованы списком стандартов, одобренным FGDC, и при определении стека были выявлены общие черты. В таблице 15.1 показано базовое программное обеспечение, необходимое для каждого проекта. После анализа основных требований к программному обеспечению и зависимых пакетов для каждого проекта были разработаны и развернуты общие платформы GeoCloud для различных федеральных

геопространственных приложений, как показано на рисунке 15.4.



Уровни (например, база данных, сервер приложений) можно разделить или объединить по мере необходимости.

Рисунок 15.4 (См. цветную вставку) Создание и настройка платформы GeoCloud для федеральных геопространственных приложений. (Из Неберта, 2010 г., [www.fgdc.gov/ngac/meetings/de-December-2010/geocloud-briefing.pptx](http://www.fgdc.gov/ngac/meetings/de-December-2010/geocloud-briefing.pptx).)

Базовые образы относятся к двум типам ОС - Linux (CentOS 5.5) и серверу Windows 2008. Образы Windows поставляются непосредственно корпорацией Майкрософт с предоставлением лицензии в виде дополнительной почасовой оплаты. После создания базового образа с нуля наиболее распространенные программные пакеты устанавливаются на образы Linux и Windows в качестве базовых платформ, таких как Java, Apache Tomcat и MySQL.

На национальном уровне базовые платформы могут быть дополнительно настроены как специализированные платформы для различных доменных приложений. Эти специализированные платформы оснащены различными геопространственными пакетами с открытым исходным кодом, и они разделены на четыре категории в соответствии с требованиями этих идентифицированных приложений: (1) ArcGIS Server, (2) Semantic Drupal, (3) OpenGeo и (4) THREDDS.

В 2012 финансовом году были разработаны два стандартных образа PaaS для поддержки стандартизированного развертывания, включая ограниченный уровень приложений, поскольку серверы приложений включают поддержку настройки пользовательского интерфейса (UI) и интерфейса прикладного программирования (API). Выбор только двух наборов геопространственных сервисов, ArcGIS Server и OpenGeo Enterprise Suite, был основан на требованиях к консолидированным приложениям федеральных агентств. Поддержка только двух образов PaaS (платформенных решений) позволяет деятельности GeoCloud координировать и совместно использовать подходы к безопасности, пакетам программного обеспечения и обновлениям, а также разработывать хорошо документированную среду хостинга для нескольких агентств и доменов приложений. Это еще больше упрощает сертификацию и аккредитацию всех развернутых виртуальных «систем», в которых используются эти образы платформ, поскольку все они являются клонами хорошо известной среды.

### 15.3.2 Подтвердите с помощью приложений агентства

После создания базовой и специализированных платформ (рис. 15.4) выбранные приложения можно было развернуть и протестировать на этих платформах для проверки. Например, Национальная инвентаризация водно-болотных угодий (NWI) Wetlands Mapper была развернута на платформе ArcGIS Server, тогда как TIGER / Line и GEOSS Clearinghouse были непосредственно перенесены на базовые платформы. В ходе этого процесса базовая и специализированные платформы были скорректированы и настроены на основе результатов тестирования выбранных приложений.

### 15.3.3 Документировать и обнародовать

Чтобы предоставить рекомендации федеральным агентствам при поиске облачных решений, GeoCloud предоставляет ряд документов, включая:

- Лучшие практики и извлеченные уроки.
- Модели затрат и контрольные показатели.
- Объединенные платформы, обновления и артефакты.

Все эти документы доступны на веб-сайте FGDC<sup>1</sup> и на портале сообщества GeoCloud<sup>2</sup>. Проектные группы, участвующие в инициативе GeoCloud, работают над распространением среди федеральных облачных сообществ интересов с помощью различных мероприятий:

- Публикация результатов развертывания и тестирования на международной конференции (Huang et al. 2010).
- Представление деятельности GeoCloud на профессиональных и общественных конференциях, таких как Федерация информационных партнеров по наукам о Земле (ESIP), Ассоциация американских географов (AAG) и Международная конференция пользователей Esri 2012 г.

Одним из важных результатов этапа документирования и распространения GeoCloud является разработка общих образов PaaS и архитектурного шаблона - многообразного образа программного обеспечения, который монтирует всю конфигурацию и данные из подключенного постоянного тома EBS. Как упоминалось ранее, сосредоточение внимания только на двух решениях PaaS многократного использования значительно упрощает разработку, обслуживание и профилирование безопасности всех систем, которые используют их для развертывания в правительстве.

<sup>1</sup>См. FGDC на <http://www.fgdc.gov/initiatives/geoplatform/geocloud>.

<sup>2</sup>См. Cloud Community Platform Portal на <http://geocloud.eglobaltech.com/>.

Таким образом, изображения и скрипты, используемые для их предоставления, являются ключевыми полезными ресурсами инициативы GeoCloud, геопространственно позволяя программе общих служб администрации Обамы.

## 15.4 ЗАЩИТА GeoCloud

Существует два уровня безопасности проектов GeoCloud: безопасность облачной платформы Amazon EC2, используемой для размещения различных правительственных приложений, и операции безопасности GeoCloud для администрирования платформ и приложений в облаке.

### 15.4.1 Защита amazon Web services (aWs)

Безопасность и конфиденциальность данных являются основными проблемами владельцев систем и могут помешать внедрению облачных решений агентствами. После проведения углубленного исследования текущих популярных поставщиков облачных услуг с точки зрения возможностей и стратегий безопасности, Amazon EC2 был выбран в качестве платформы для GeoCloud. AWS имеет ряд сертификатов, включая ISO 27001 и сертификацию уровня 1 по финансовой безопасности и передаче личной идентификационной информации (PII) в период инициатив GeoCloud (AWS Security 2013). Компания получила одобрение на размещение 12 общедоступных федеральных систем, включая Recovery.gov и Treasury.gov, на основании Низкой аккредитации Федерального закона об управлении информационной безопасностью (FISMA), AWS внутренне завершила сертификацию FISMA - средний уровень, задокументированные процедуры и анализ пробелов. по состоянию на начало 2013 года.

Amazon EC2 IaaS предшествует процессу сертификации Федеральной программы управления рисками и авторизацией (FedRAMP), но Amazon предоставила документацию и средства контроля безопасности в рамках инициированных агентством разрешений FedRAMP. Ожидается, что AWS получит одобрение FedRAMP, так что оно может быть легко использовано любым федеральным агентством.

Хотя AWS предлагает свою инфраструктуру потребителям облака, она не предоставляет физическое частное облако. Однако доступны виртуальные частные облака, такие как GovCloud <sup>1</sup>, доступ к которым осуществляется через виртуальные частные сети (VPN).

В мае 2013 года AWS получила одобрение безопасности FedRAMP для своих предложений инфраструктуры, представленное Министерством здравоохранения и социальных служб (HHS) для использования в рамках своей деятельности по модернизации ИТ. Это право на эксплуатацию (АТО) может быть использовано другими федеральными агентствами для дальнейшего ускорения процесса утверждения (виртуальных) систем (см. Пресс-релиз по адресу <http://phx.corporate-ir.net/phoenix.zhtml?c=176060&p=irol-newsArticle&ID=1822454&highlight>).

---

<sup>1</sup> См. AWS на <http://aws.amazon.com/govcloud-us/>.

## 15.4.2 Операция безопасности GeoCloud

GeoCloud имеет отдельные учетные данные продукта для каждого проекта для входа в систему через веб-консоль управления AWS. С этими учетными данными потребитель облака (который является системным администратором каждого проекта) может получать доступ и управлять облачными сервисами, выделенными для проекта, только через веб-консоль управления AWS. Учетные данные были созданы и предоставлены координатором системы GeoCloud. Это может обеспечить конфиденциальность на уровне проекта, так что каждый системный администратор не сможет повлиять или разрушить данные и приложения других проектов.

Учетные данные на уровне продукта позволяют потребителям облака (системным администраторам проектов) выделять и выпускать виртуальные машины, а также настраивать сетевой доступ для целевых приложений. Некоторые высокоуровневые операции не разрешены системным администраторам уровня проекта, например создание нового общего образа ОС / программного обеспечения на основе запущенного экземпляра. Поскольку облачные образы для федеральных приложений должны соответствовать нескольким сертификатам, разрешениям и правилам, только избранные системные координаторы GeoCloud могут настраивать и создавать общие образы.

## 15.5 Операционные расходы в облаке

Оценка стоимости каждого из начальных проектов проводилась с использованием онлайн-калькулятора и основывалась на требованиях к передаче данных, хранению, ЦП и спросу. Большинство проектов можно разместить в AWS (~ 350–500 долларов в месяц). Ежемесячные расходы, потраченные на облако EC2 с июля по октябрь 2010 г. для TIGER / Line и GEOSS Clearinghouse, показаны на рисунке 15.5. Можно заметить, что ежемесячная стоимость проекта TIGER / Line составляет около 300 долларов. Средние ежемесячные расходы на Информационный центр GEOSS составляют около 270 долларов США.

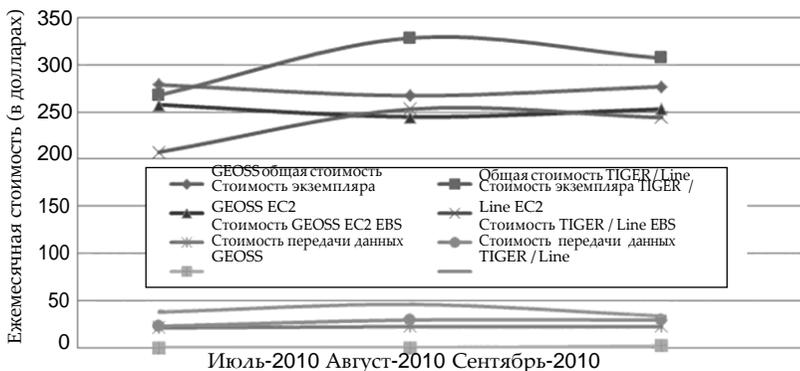


Рисунок 15.5 Ежемесячные затраты на проект TIGER / Line и Информационный центр GEOSS.

Однако некоторые проекты в облаке были дорогостоящими из-за больших затрат на хранение или передачу данных. Например, размещение всего хранилища пространственных данных USGS EROS в облачной среде может быть очень дорогостоящим. Для сервисов, определенных в рамках первого года работы, потребуются шесть больших экземпляров Windows, 5 ТБ хранилища EBS, 50 ГБ на входе в месяц и 500 ГБ на выходе без S3 или эластичных LB. В совокупности они будут стоить от 2500 до 4000 долларов в месяц (SDW 2010).

## 15.6 ОБСУЖДЕНИЕ

После одного года работы и мониторинга выбранных проектов в облаке в 2010–11 годах извлеченные уроки были записаны для использования федеральными агентствами в будущем облачных решений.

Процесс утверждения безопасности. Большой проблемой для GeoCloud была проверка утверждений системы безопасности агентства на основе документации от Amazon. FISMA требует от федеральных агентств разработки, документирования и внедрения системы защиты информации для своих данных и инфраструктуры. Чтобы перенести федеральные приложения GeoCloud на EC2, каждый проект должен был выполнить анализ оценки риска на том же уровне, что и FISMA. Например, проекту TIGER / Line удалось достичь уровня гарантии FISMA благодаря заключениям, сделанным в ходе оценки рисков, проведенной Управлением информационной безопасности (OIS). Оценка рисков OIS была сосредоточена на трех основных элементах безопасности: (1) конфиденциальность, (2) целостность и (3) доступность. Данные TIGER/Line имеют очень низкий уровень воздействия на конфиденциальность из-за их общедоступности. Более того, было определено, что внутреннее восстановление системы TIGER / Line было чрезвычайно быстрым и эффективным, что привело к эффективному компенсирующему контролю для предотвращения любых атак на доступность системы в AWS. На основании приведенного выше анализа в сочетании с низкой вероятностью атаки на этот тип общедоступной информации был сделан вывод, что общий риск, связанный с использованием этой системы в AWS, был очень низким (TIGER / Line 2010).

Развертывание программного обеспечения. В настоящее время все проекты GeoCloud используют один из двух общих образов платформы, включая основной сервер приложений, вспомогательные библиотеки и инструменты веб-развертывания, включая MySQL, PostgreSQL, gcc, Java и Tomcat. Однако хранилище и производительность системы могут пострадать, поскольку установлено и запущено больше программного обеспечения, чем требуется. С тех пор были разработаны сценарии конфигурации с использованием Amazon CloudFormation для выбора и активации определенного программного обеспечения.

Время развертывания. Основываясь на опыте системных администраторов тех федеральных приложений, которые были новичками в облаке, на ознакомление с сервисами AWS уходит примерно (1) от одной до двух недель, (2) два дня на настройку базовой системы, работающей из общедоступной среды. образ машины или усиленный образ, (3) два дня для написания сценария обработки настройки приложений и (4) одна неделя для изучения балансировщика нагрузки, возможностей автоматического масштабирования, эластичного IP-адреса, резервного копирования данных и кода и восстановления.

Поскольку для неспециалиста знакомство с AWS занимает много времени, возможно, было бы лучше провести вводный семинар для новичков в AWS, чтобы быстро начать работу. Эта книга может служить частью этого требования.

Отказоустойчивость, резервирование. AWS предоставляет возможность расширять системные ресурсы по мере необходимости. В соответствии с конфигурацией при необходимости могут быть доступны дополнительный ЦП и память (TIGER / Line 2010). Однако, когда количество одновременных пользователей превышает 600, могут быть некоторые ошибки ответа (GEOSS Clearinghouse 2011). Множественные копии могут быть распределены в разные облачные регионы для дополнительной защиты от сбоя системы, предлагая возможности быстрого аварийного переключения. Используя сценарии CloudFormation, можно создавать резервные копии томов данных с помощью процесса создания моментальных снимков с частотой, определяемой издателем.

## 15.7 КРАТКОЕ ИЗЛОЖЕНИЕ

На примере инициативы GeoCloud в этой главе показано, как переносить федеральные приложения в облако. Проект GeoCloud также показывает, как создать платформу сообщества геопространственного облака для различных приложений. В этой главе дается краткое описание истории GeoCloud (Раздел 15.1), архитектуры (Раздел 15.2), деятельности (Раздел 15.3), безопасности (Раздел 15.4), стоимости (Раздел 15.5) и извлеченных уроков (Раздел 15.6) нескольких федеральных приложений в нескольких агентствах для создание геопространственной платформы сообщества.

## 15.8 ПРОБЛЕМЫ

1. Что такое инициатива GeoCloud?
2. Каковы преимущества создания общей геоплатформы?
3. Что такое архитектура GeoCloud?
4. Перечислите действия GeoCloud.
5. Каковы соображения безопасности? Как достигается безопасность для федеральных приложений?
6. Каковы эксплуатационные расходы в облаке для типичного веб-приложения? Для приложений с интенсивным использованием данных?
7. Какие уроки были извлечены?

## СПИСОК ЛИТЕРАТУРЫ

- AWS Security. 2010. <http://aws.amazon.com/security/> (accessed March 13, 2013).
- GEOSS. 2011. GEOSS Clearinghouse Report. <http://www.fgdc.gov/initiatives/geoplatfrom/geocloud/reports/fgdc-geocloud-project-report-geonetwork.pdf> (accessed March 13, 2013).

- Huang, Q., C. Yang, D. Nebert, K. Liu, and H. Wu. 2010. Cloud Computing for Geosciences: Deployment of GEOSS Clearinghouse on Amazon's EC2. In *Proceedings of ACM SIGSPATIAL International Workshop on High Performance and Distributed Geographic Information Systems (HPDGIS)*, November 2, San Jose, CA.
- Nebert, D. 2010. Deploying Federal Geospatial Services in the Cloud. [www.fgdc.gov/ngac/meetings/december-2010/geocloud-briefing.pptx](http://www.fgdc.gov/ngac/meetings/december-2010/geocloud-briefing.pptx) (accessed March 13, 2013).
- SDW. 2010. EROS Spatial Data Warehouse Report. <http://www.fgdc.gov/initiatives/geoplatform/geocloud/reports/fgdc-geocloud-project-report-eros-sdw.pdf> (accessed March 13, 2013).
- U.S. Census Bureau. 2010. Topologically Integrated Geographic Encoding and Referencing (TIGER) TIGER/Line, 2010 Final Report. <ftp://ftp2.census.gov/geo/tiger/TIGER2010>.
- U.S. Fish and Wildlife Service. 2012. National Wetlands Inventory Database Metadata. [http://www.fws.gov/wetlands/data/metadata/FWS\\_Wetlands.xml](http://www.fws.gov/wetlands/data/metadata/FWS_Wetlands.xml) (accessed March 13, 2013).

## Часть V

---

# Будущие направления

---

В данной части рассматриваются, а затем обсуждаются направления будущих исследований и разработок в области пространственных облачных вычислений. Глава 16 знакомит с интенсивностью данных; вычислительные, параллельные и пространственно-временные; и обсуждает возможные решения и исследования, оставшиеся для решения проблем интенсивности с использованием облачных вычислений. В главе 17 представлены потребности в будущих исследованиях по трем аспектам геонаучного видения, технических достижений и синергетических достижений между облачными вычислениями и социальными науками.



## Глава 16

---

# Обработка больших объемов данных, вычислений, одновременного доступа и пространственно-временных шаблонов

*Цюньюн Хуан, Чжэньлун Ли, Кай Лю, Цзичжэ Ся, Юньфэн Цзян, Чэнь Сюй и Чаовой Ян*

---

Достижения наук о Земле включают работу с большими объемами данных, вычислениями, одновременным доступом и пространственно-временными моделями. В данной главе обсуждается каждый из уровней интенсивности и на примерах представлены потенциальные решения для решения проблем интенсивности с использованием облачных сервисов.

### 16.1 ВВЕДЕНИЕ

Земля и ее имущество эволюционируют в четырехмерном (4D) мире. Науки о Земле изучают закономерности и возможные тенденции этой эволюции конкретных или комплексных явлений, таких как пыльные бури и изменение климата. Грандиозные задачи, стоящие перед нами в 21 веке, неизбежно связаны с этой четырехмерной эволюцией и решением проблем, которые требуют от нас понимания принципов и взаимосвязи, лежащих в основе эволюции, и прогнозирования потенциальной будущей эволюции на основе различных факторов для разных сообществ пользователей. Например, нам необходимо интегрировать области суши, океана и атмосферы, чтобы лучше понять, как меняется климат. Эти человеческие знания в соответствии с процессом приложения включают в себя большие объемы данных (например, для изменения климата), значительные вычислительные требования для обработки растущих объемов данных и сложных алгоритмов (например, для обработки в социальных сетях) и большое количество одновременных подключений к знаниям и вычислениям (например, для аварийного реагирования на цунами и землетрясения). Три уровня интенсивности данных, вычислений и одновременного доступа - все основаны на пространственно-временной интенсивности, с которой нам приходится иметь дело в различных масштабах и разрешениях. Эта интенсивность естественно соответствует характеристикам облачных сервисов, таким как масштабируемые вычислительные ресурсы для одновременного доступа, практически неограниченные вычислительные мощности для вычислительной интенсивности и глобальные, доступные центры обработки данных поставщиками облачных услуг для распределенных местоположений данных (Yang et al. 2011a). В этой главе рассматриваются и обсуждаются направления будущих исследований использования облачных сервисов для решения таких проблем.

276 К. Хуан, З. Ли, К. Лю, Дж. Ся, Ю. Цзян, К. Сюй, К. Ян

---

## 16.2 БОЛЬШИЕ ДАННЫЕ

### 16.2.1 Введение

Большие данные обычно относятся к четырем V: объем, скорость, разнообразие и достоверность<sup>1</sup>. Объем относится к размеру данных; скорость указывает на то, что большие данные чувствительны ко времени; Разнообразие означает, что большие данные включают различные типы данных со сложными отношениями, а достоверность указывает на надежность данных. В геопространственной области большие данные в основном создаются (1) различными датчиками с разным пространственным, временным и спектральным разрешением, которые регистрируют физические явления, (2) различными научными моделями, моделирующими и предсказывающими физические явления, и (3) экспоненциальный рост взаимосвязи между большими данными. Как эффективно хранить и обрабатывать большие данные для эффективного доступа к данным и анализа представляет собой критически важную задачу:

- Задача управления большими данными. Эффективное управление большими данными для обнаружения, обработки и визуализации данных является фундаментальной, но все же важной предпосылкой для обработки больших объемов данных. Внутренняя неоднородность больших данных из-за различных методов сбора данных и сценариев использования создает серьезные проблемы для поиска, интеграции и управления данными (Ли и др., 2011). Большой объем данных еще больше усложняет управление большими данными. Для хранения больших данных требуется высокомасштабируемые устройства хранения, которые можно легко масштабировать для выполнения запроса и легко масштабировать для минимизации затрат.
- Проблема обработки больших данных. В большинстве случаев управление большими данными не является конечной целью. Конечная цель - преобразовать эти данные в информацию, знания и идеи путем обработки и анализа. Однако обработка больших данных в приемлемые сроки представляет собой серьезную проблему. Например, 10 терабайт - это нормальный объем данных, производимый климатической моделью в Climate @ Home (глава 9). Предположим, что скорость чтения/записи для устройства хранения составляет 50 мегабайт в секунду, для чтения 10 терабайт данных на одной машине требуется 2,5 дня, для чтения 100 терабайт данных требуется почти месяц.
- Задача визуализации больших данных - визуализация больших данных необходима для понимания географических явлений путем преобразования данных и информации в графические представления. Основываясь на измерении визуализации, визуализацию больших геопространственных данных можно разделить на две категории: двухмерная (2D) визуализация и многомерная (многомерная) визуализация, например, 3D и 4D (3D со временем). 2D-визуализация относительно проста, а методы более зрелые, такие как пиксельная визуализация для интуитивного просмотра и различные диаграммы для визуальной аналитики.

---

<sup>1</sup> См. IBM на <http://www-01.ibm.com/software/data/bigdata/>.

Однако многомерная визуализация по-прежнему создает проблемы из-за характеристик множества измерений и переменных, а также сложности и объема пространственно-временных данных (Yang et al. 2013).

## 16.2.2 Пример Climate@home

Climate @ Home использует вычислительные ресурсы, предоставленные гражданами, чтобы обеспечить запуск большого количества климатических моделей (ModelE) в распределенной среде. ModelE может запускаться много раз на тысячах добровольцев по всему миру. Каждая модель, запускаемая с 10-летним моделированием, будет генерировать 10 гигабайт данных как в 3D, так и в 4D, содержащих 566 климатических переменных. Модель 100 запускается с 200-летним моделированием, которое выдаст 20 терабайт данных (Ли и др., 2013). Для тестирования и оптимизации прогноза климата с помощью ModelE требуются тысячи прогонов модели, которые могут генерировать петабайты данных. Эти данные имеют большой объем, и скорость их накопления высокая.

## 16.2.3 Решения

Для решения проблем используются три решения соответственно.

- *Метаданные, пространственно-временной индекс и распределенная файловая система (DFS)*. Метаданные представляют собой структурированную описательную информацию о данных и широко используются в различных приложениях, интенсивно использующих данные (Singh et al. 2003; Yee et al. 2003). Метаданные важны в управлении большими данными для (1) поддержки организации массивных данных путем описания дизайна и спецификации сложных структур данных, (2) обеспечения эффективного обнаружения данных и доступа к ним путем описания содержимого данных и (3) использования в качестве фундаментальных компонент для интеграции разнородных данных. Хотя метаданные служат механизмом, помогающим описывать большие данные и управлять ими, производительность запросов и доступа к данным критически важна для геопространственных приложений по запросу. Механизм индексации может значительно улучшить обнаружение данных за счет предоставления эффективных алгоритмов поиска данных (Theodoridis et al. 1996). Пространственно-временной индекс объединяет пространственно-временные принципы (Ян и др., 2011b) и методы индексирования, чтобы обеспечить возможности высокопроизводительного поиска для геопространственных приложений с большим объемом данных. DFS (глава 3, раздел 3.4.1) - потенциальное решение проблемы хранения больших данных в облачной службе. Поскольку DFS обеспечивает возможность прозрачной репликации и отказоустойчивости, надежность повышается. Кроме того, DFS предназначена для использования стандартного оборудования (например, ПК) в слабосвязанной среде; следовательно, емкость хранилища можно легко увеличивать и уменьшать.

- Параллельные вычисления с облачными сервисами. Что касается решения проблемы обработки больших данных, используется интеграция параллельных вычислений и облачных вычислений (Fox et al. 2010). В частности, используется MapReduce (Lämmel 2008), платформа параллельной обработки данных, впервые разработанная Google, а Eucalyptus (глава 13) выбран в качестве платформы облачных вычислений. MapReduce предназначен для параллельной обработки больших данных с использованием обычных ПК в качестве вычислительных узлов. Обычно MapReduce основан на DFS. Интеграция DFS и MapReduce обеспечивает надежное, масштабируемое и эффективное решение для хранения и обработки больших данных. Реализации MapReduce и DFS с открытым исходным кодом в Hadoop MapReduce и Hadoop Distributed File System (HDFS) (глава 3, раздел 3.4.3) используются для реализации инфраструктуры хранения и обработки данных. Кроме того, Eucalyptus используется для создания частного облачного сервиса с гибкими вычислительными возможностями. Виртуальные машины, запущенные в этом частном облаке, могут использоваться в качестве узлов хранения и обработки для дальнейшего улучшения масштабируемости, а также снижения затрат.
- Распределенная визуализация на базе графического процессора (GPU) в облачных сервисах. Визуализация геопространственных данных в multi-D может создавать более интуитивно понятные и визуальные эффекты и, таким образом, помогать пользователям эффективно исследовать потенциальные закономерности в научных данных. Традиционные методы визуализации с помощью обычного канала визуализации не могут удовлетворить требования к производительности при визуализации многомерных данных большого объема, особенно с учетом измерения времени. Для решения этой проблемы используется визуализация на основе графического процессора за счет использования аппаратного ускорения графики, а также параллельных вычислений. Например, платформа CUDA<sup>1</sup>, опубликованная NVIDIA, обеспечивает мощные возможности параллельных вычислений на графическом процессоре, чтобы обеспечить более высокую скорость рендеринга. Кроме того, поскольку объем данных велик, применяется стратегия распределенной визуализации для дальнейшего повышения производительности визуализации путем разделения большого набора данных на небольшие объемы и их параллельного рендеринга. Облачные ресурсы с поддержкой графического процессора (например, экземпляры Amazon EC2 GPU) позволяют нам гибко использовать и реализовывать модель визуализации на основе графического процессора (см. Главу 17, раздел 17.2.5)..

#### **16.2.4 Сохраняющиеся проблемы и будущие исследования**

Потенциальные подходы обеспечивают комплексное решение для управления интенсивностью данных от управления большими данными; обработка больших данных для визуализации больших данных. Однако следующие три аспекта все еще требуют дальнейшего изучения.

---

<sup>1</sup> См. NVIDIA на <http://www.nvidia.com/object/nvidia-iray.html>.

- При работе с приложениями, интенсивно использующими данные, большое значение имеют *хранилища и вычислительные мощности*. Что касается емкости хранилища, можно использовать такие облачные сервисы хранения, как Amazon S3 и Elastic Block Store (EBS), однако стоимость хостинга и передачи данных в коммерческой облачной службе хранения данных может быть высокой, учитывая размер данных. То же самое относится и к вычислительной мощности. Мы можем создать частную облачную службу, чтобы снизить стоимость, но это не всегда так. Компромисс между использованием коммерческих сервисов и созданием частного облачного сервиса необходимо тщательно изучить с учетом существующей ИТ-инфраструктуры организации. Учитывая постоянно растущий объем данных и ограниченные (из соображений бюджета) возможности хранения и вычислений, необходимы дальнейшие исследования (1), как устранить избыточные и бесполезные данные и информации с помощью интеллектуальных методов предварительной обработки / очистки данных, (2) как уменьшить размер данных с помощью эффективных алгоритмов сжатия данных и (3) как точно рассчитать объем хранилища и вычислительные ресурсы при использовании коммерческих облачных сервисов.  
(1) Доступ к большим данным по запросу имеет решающее значение для геопространственных приложений реального времени, особенно для тех, которые работают в веб-среде. Для большинства приложений, когда размер данных невелик, доступ к данным по запросу не является проблемой; при работе с большими данными дела обстоят иначе. Метод параллельных вычислений имеет большие успехи в пакетной обработке больших объемов данных, что подходит для систем с допустимой задержкой. Однако запросы и доступ к терабайтам данных в реальном времени по-прежнему остаются сложной проблемой. Требуются дальнейшие исследования: (1) как разработать более эффективные методы пространственно-временной индексации для поиска данных и (2) как оптимизировать метод параллельных вычислений для подмножества данных и обработки.
- Разрыв между MapReduce и научными данными - MapReduce отлично справляется с обработкой неструктурированных данных, таких как тексты, документы и веб-страницы. Однако большинство приложений для геофизических исследований имеют дело со структурированными данными, такими как реляционные базы данных, и полуструктурированными данными, такими как научные данные на основе массивов. Как правило, для обработки наборов научных данных (например, NetCDF) в Hadoop MapReduce обычно используются три подхода: (1) преобразование двоичного кода в текст, который преобразует набор данных на основе двоичного кода в набор данных на основе текста (Zhao et al. 2010); (2) реорганизация данных, которая реорганизует исходный набор данных без преобразования его в другой формат данных и сохраняет его в файлах, поддерживаемых Hadoop, таких как Sequence Files<sup>1</sup> (Duffy et al. 2012); и (3) разработка промежуточного программного обеспечения поверх Hadoop для обработки научных данных (Buck et al. 2011), такого как SciHadoop (Buck et al. 2011).

<sup>1</sup> См. the Hadoop Wiki на <http://wiki.apache.org/hadoop/SequenceFile>.

280 К. Хуан, З. Ли, К. Лю, Дж. Ся, Ю. Цзян, К. Сюй, К. Ян

Такие подходы в некоторой степени заполняют пробел между данными Hadoop и геолого-геофизическими данными и обеспечивают общее руководство по использованию Hadoop для решения проблем с большими данными в геопространственной области. Тем не менее, необходимы дальнейшие исследования для улучшения полноты и эффективности этих подходов, а также для изучения других возможных решений.

## 16.3 ИНТЕНСИВНОСТЬ ВЫЧИСЛЕНИЙ

### 16.3.1 Введение

Интенсивность вычислений - еще одна проблема, которую необходимо решить в науках о Земле. В контексте геонаук, проблемы с интенсивными вычислениями обычно возникают при интеллектуальном анализе данных для информации / знаний, извлечения параметров и моделирования явлений (Yang et al. 2011a).

- Интеллектуальный анализ данных для информации / знаний. Многие технологии интеллектуального анализа данных были исследованы, чтобы лучше понять, относятся ли наблюдаемые временные ряды и пространственные закономерности к подсистемам Земли, включая биосферу, атмосферу, литосферу, а также социальные и экономические системы. Взаимодействия между этими подсистемами в пространственно-временных измерениях по своей сути сложны, а алгоритмы и процессы интеллектуального анализа данных для исследования таких взаимодействий требуют больших вычислительных ресурсов (Donner et al. 2009). Например, изучение взаимодействия между глобальным углеродным циклом и климатической системой требует больших вычислительных ресурсов (Kumar et al. 2006).
- Извлечение параметров - сложные геофизические алгоритмы используются для получения параметров явления (например, температуры) из массивных данных наблюдения Земли. Однако выполнение сложных алгоритмических процессов для извлечения параметров требует чрезвычайно больших вычислительных ресурсов. Например, требования к вычислениям и хранению данных для получения региональных и глобальных условий воды, энергии и углерода на основе мультисенсорных и многоступенчатых наборов данных намного превышают возможности одной рабочей станции (Kumar et al. 2006).
- Моделирование явлений. Моделирование геопространственных явлений особенно сложно при рассмотрении всей динамики явлений земной системы. Примером может служить моделирование и прогнозирование циклических процессов (Donner et al. 2009), включая океанские приливы (Crtwright, 2000), землетрясения (Shuster, 1897) и пыльные бури (глава 10).

### 16.3.2 Пример с интерполяцией цифровой оценочной модели

Цифровая модель оценки (DEM) относится к цифровому представлению топографии поверхности земли или ландшафта и также широко известна как цифровая модель местности (DTM). ЦМР может быть представлена как растр (сетка ячеек) или вектор (треугольная неправильная сеть) в Географической информационной системе (GIS) (Audenino et al. 2001).

Интерполяция ЦМР для больших географических областей может столкнуться с проблемами в практических приложениях, особенно для веб-приложений, таких как визуализация местности, которая требует быстрого реагирования. Кроме того, потребность в вычислениях превышает возможности традиционного единственного процессора, способного выполнять только последовательную обработку (Huang and Yang 2011) (см. Главу 5, раздел 5.3).

### **16.3.3 Решения**

Повышение вычислительной мощности, включающей ЦП, память и диск, является основным решением для решения вычислительных проблем с точки зрения оборудования. Многоядерные и многоядерные центральные процессоры (ЦП) или графические процессоры, память с большой емкостью и кэшем, а также диски с высокоскоростным вводом-выводом широко используются для развертывания вычислительных приложений. Однако общий подход к решению вычислительных проблем заключается в использовании параллельного программирования, при котором вычислительные задачи разделяются и выполняются на нескольких ядрах ЦП или графических процессоров на одной или нескольких машинах (Huang et al. 2012).

#### **16.3.3.1 Облачная обработка данных на ЦП**

Скорость вычислений традиционной модели последовательных вычислений на одной машине не может удовлетворить растущие потребности в вычислениях. Высокопроизводительные вычисления (HPC) или grid-вычисления (Armstrong et al. 2005; Huang and Yang 2011) использовались для решения все более крупных вычислительных задач в задачах географической науки, таких как моделирование пыльной бури. Такую большую проблему, требующую интенсивных вычислений, можно быстро решить, разделив ее на множество подзадач и используя несколько вычислительных ресурсов для решения этих подзадач параллельно. Взяв в качестве примера интерполяцию DEM, домен DEM можно разделить на несколько подобластей равного размера, и эти подобласти могут обрабатываться разными вычислительными ресурсами.

На рисунке 16.1 показана производительность интерполяции DEM с вычислением сетки, где для обработки вычислений используются традиционные распределенные ядра ЦП (Huang and Yang 2011). Однако не каждая организация или конечный пользователь может быть оснащена инфраструктурой HPC. Дефицит ресурсов препятствует продвижению HPC. К счастью, облачные вычисления предлагают мощную и доступную альтернативу для выполнения крупномасштабных вычислительных задач (Huang et al. 2013). Глава 5, раздел 5.3 представила подробные шаги по запуску интерполяции матрицы высот в EC2.

282 К. Хуан, З. Ли, К. Лю, Дж. Ся, Ю. Цзян, К. Суй, К. Ян



Рисунок 16.1 Интерполяция DEM с параллельной обработкой CPU.

### 16.3.3.2 Облачная обработка данных на GPU

Появление вычислений на графических процессорах в последние десятилетия предоставляет ученым и инженерам дополнительные вычислительные методы для решения сложных научных, инженерных и бизнес-задач. По сравнению с многоядерными вычислениями на ЦП, многоядерные вычисления на GPU оказались лучшим выбором во многих случаях, особенно в плане повышения эффективности рендеринга при геопространственной визуализации (Ли и др., 2012).

Рисунок 16.2 показывает производительность графического процессора по сравнению с процессором при поддержке интерполяции матрицы высот. Интерполяция выполняется отдельно на устройстве ЦП или ГП с тем же алгоритмом интерполяции, по которому каждое значение ячейки вычисляется на основе значения высоты его соседей. При вычислениях на ЦП для повышения производительности используются многопоточность. При вычислениях на GPU используются библиотеки CUDA, которые поддерживают параллельные вычисления. Каждая ячейка сетки в DEM вычисляется в одном потоке графического процессора. При использовании вычислений на GPU интерполяция может выполняться только в памяти GPU, а не в памяти CPU. Следовательно, необходимо переносить данные матрицы высот из памяти ЦП в память ГП. После завершения обработки полученный результат можно было перенести обратно в память процессора.

Графический процессор как новая вычислительная техника постепенно интегрируется в структуру облачных вычислений, которая называется Cloud GPU. В настоящее время потребители могут легко создавать виртуальные машины (VM) с графическим процессором для оптимизации своих приложений, используя архитектуру графического процессора, а также облачные возможности эффективности, масштабируемости и быстрой доступности. Например, облако Amazon EC2 позволяет использовать преимущества параллельной производительности графических процессоров NVidia Tesla с использованием моделей программирования CUDA (NVidia) и OpenGL для вычислений на графических процессорах.

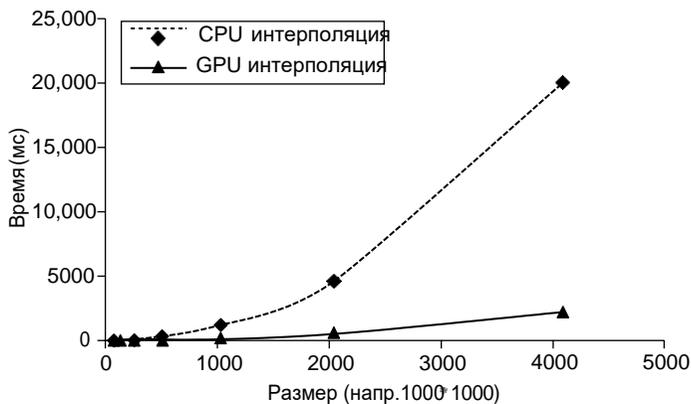


Рисунок 16.2 Интерполяция DEM с помощью GPU и CPU.

### 16.3.4 Сохраняющиеся проблемы и будущие исследования

Основываясь на обсуждениях решений интерполяции DEM, вычисления как на CPU, так и на графическом процессоре имеют потенциал для решения различных задач, связанных с интенсивными вычислениями в науках о Земле. Облачные вычисления могут предоставлять гибкие и мощные вычислительные ресурсы на базе CPU и GPU. Однако остается еще много проблем с облачными вычислениями для поддержки научных вычислений. Например, Хуанг и др. (2013) обнаружили, что облачные вычисления могут стать потенциальным решением для поддержки небольших по требованию приложений высокопроизводительных вычислений. Необходимо провести дополнительные исследования, чтобы изучить возможность использования облачных вычислений для лучшей поддержки крупномасштабных научных вычислений и узнать, как лучше всего адаптироваться к этой новой компьютерной парадигме.

- Интероперабельная и интеллектуальная структура. Вычисления на CPU и GPU могут поддерживать параллельную обработку и имеют свои преимущества и недостатки при выполнении задач геопространственных вычислений (Ли и др., 2013). Полноценное использование этих двух вычислительных парадигм для решения научных вычислительных задач ставит серьезные задачи. Например, сложно эффективно распределить подзадачи на соответствующие вычислительные устройства (например, CPU, ГП или гибридный GPU и CPU) с разными уровнями вычислительных возможностей. Срочно необходима интероперабельная и интеллектуальная структура, которая должна быть разработана и реализована для решения этой проблемы. В рамках этой структуры вычислительная задача может грамотно выбирать вычислительные устройства и выполняться эффективно.
- Сеть - Huang et al. (2013) утверждали, что облачные сервисы, как правило, не полностью готовы к тому, чтобы служить в качестве крупномасштабного виртуального кластера для поддержки вычислительных и коммуникационных приложений, а виртуализированная сеть является ключевым фактором, который не позволяет облачным сервисам достичь более высокой производительности.

Традиционные высокопроизводительные вычисления настраиваются для виртуализированных ресурсов в эпоху облачных вычислений. Например, с виртуализированной сетью в качестве узкого места приложения могут попытаться выполнить избыточные вычисления, уменьшая при этом накладные расходы на связь, чтобы полностью использовать масштабируемую вычислительную мощность облаков. В то же время облачные сервисы могут также устранять узкие места в сети путем (1) предоставления экземплярам облака высокоскоростного соединения (Ostermann et al. 2010) и (2) оптимизации алгоритмов планирования промежуточного программного обеспечения облачных сервисов при диспетчеризации Виртуальные машины на физических хостах для приложений НРС для уменьшения накладных расходов на связь.

- **Стоимость.** Стоимость является одной из важных проблем для потребителей при принятии решений облачных вычислений, особенно при использовании общедоступных и гибридных облаков (Zhang, Cheng, and Boutaba, 2010). В зависимости от требований приложения к вычислениям, хранению и обмену данными ресурсы общедоступного облака могут быть более или менее дорогостоящими, чем размещение приложения внутри компании в течение длительного времени (Armbrust et al. 2010). Однако многие приложения предъявляют высокие требования к вычислительным ресурсам, при которых при необходимости следует задействовать большое количество вычислительных ресурсов, например, поддержка прогнозирования пыльных бурь в реальном времени (Huang et al. 2013). В этой ситуации было бы более рентабельно использовать облачный сервис. Следует предложить разумную модель затрат, чтобы зафиксировать динамические потребности приложения в ресурсах и рекомендовать потребителям облака лучшие решения для запуска приложений (Глава 6).

## 16.4 ИНТЕНСИВНОСТЬ МНОГОПОТОЧНОЙ СРЕДЫ

### 16.4.1 Введение

Многие приложения для геолого-геофизических исследований сталкиваются с проблемами одновременного доступа, когда к приложениям одновременно обращаются многие пользователи. Избыточный параллельный доступ может повлиять как на оборудование, так и на программное обеспечение. В некоторых случаях одновременный доступ вызывает проблемы с оборудованием, например нехватку памяти или высокую загрузку ЦП. В других случаях одновременный доступ вызовет проблемы с программным обеспечением, например взаимоблокировку. Проблема одновременного доступа имеет решающее значение для приложений геолого-геофизических исследований с большими сообществами пользователей, таких как Информационная служба GEOSS (CLH) (Глава 8)<sup>1</sup> и Геопространственные платформы<sup>2</sup>, потому что либо (1) пользователи могут не получать правильные ответы, когда компьютер выполняет избыточные запросы; или (2) пользователи слишком долго ждут ответа от приложения.

<sup>1</sup> См. GEOSS Clearinghouse на <http://clearinghouse.cisc.gmu.edu/geonetwork>.

<sup>2</sup> См. Geospatial Platform на <http://www.geoplatform.gov/home/>.

## 16.4.2 Пример GEOSS Clearinghouse

CLH - это приложение для изучения земли, которое имеет большое количество пользователей, распределенных по всему миру. Он предоставляет функции как локального, так и удаленного поиска. Анализируя журналы доступа, большинство пользователей использует удаленный поиск с помощью службы каталогов Open Geospatial Consortium (OGC) для Интернета (CSW). На рисунке 16.3 показано пространственное распределение пользователей CLH. Существует сильная картина, когда регионы с высокой плотностью населения (например, США, Европа) имеют большое количество конечных пользователей и создают многоодновременного доступа. На рисунке 16.4 показана частота доступа пользователей в Австралии и Новой Зеландии с изменением времени с пиками в определенные часы, когда CLH приходится обрабатывать большое количество одновременных запросов конечных пользователей. Следовательно, одновременная интенсивность критична для CLH.

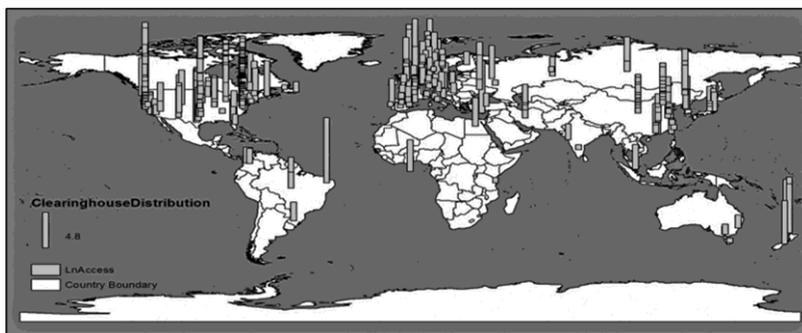


Рисунок 16.3 Распространение CLH пользователями.

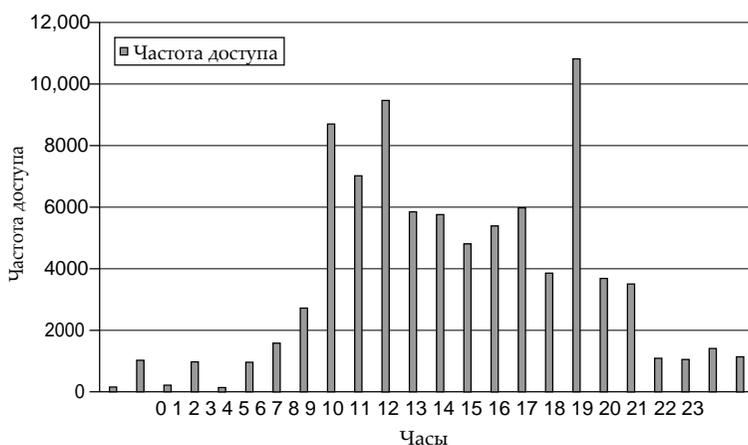


Рисунок 16.4 Частота доступа пользователей Австралии и Новой Зеландии к CLH.

286 К. Хуан, З. Ли, К. Лю, Дж. Ся, Ю. Цзян, К. Сюй, К. Ян

### 16.4.3 Решение

Есть несколько решений, которые можно использовать для обработки одновременной интенсивности в приложениях геолого-геофизических исследований.

#### 16.4.3.1 Доставка контента по всему миру

Используя облачные сервисы, потребители могут решить проблемы одновременного доступа двумя способами: балансировкой нагрузки и эластичностью. Балансировка нагрузки и масштабируемость облачных вычислений могут увеличить вычислительную мощность при низких затратах. CLN реплицируется для балансировки нагрузки от огромного числа конечных пользователей. Эти репликации распределены в нескольких местах и настроены для работы в качестве единой службы в глобальной геопространственной кибер-инфраструктуре (Yang et al. 2010). Облачные сервисы предоставляют идеальную платформу для реализации этого механизма балансировки нагрузки. Образ, содержащий настроенное приложение, может быть встроенным в облачные сервисы, а затем новое реплицированное приложение может быть запущено в доступных облачных зонах. Доступные облачные зоны подробно перечислены в главе 11. В настоящее время в CLN есть образы экземпляров в Windows Azure и Amazon EC2. Благодаря поддержке облачных сервисов мы можем легко создать механизм балансировки нагрузки для обработки массовых одновременных запросов для CLN и других пространственных сервисов.

В настоящее время у большинства облачных провайдеров имеется множество региональных центров обработки данных (например, Amazon EC2 имеет 11 центров обработки данных на начало 2013 года). В зависимости от распределения доступа пользователей мы можем развернуть пять серверов EC2 в разных местах, чтобы справиться с глобальной интенсивностью доступа пользователей. CLN можно развернуть в нескольких регионах с помощью облачных сервисов EC2 (рисунок 16.5). Например, когда пользователи получают доступ к услугам, пользователям западного побережья не нужно переходить на сервер восточного побережья.

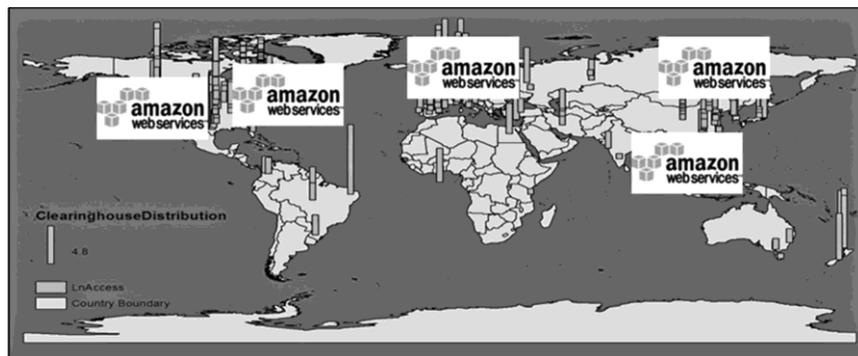


Рисунок 16.5 (См. цветную вставку) Глобальная доставка контента для управления глобальным доступом пользователей.

### 16.4.3.2 Гибкость

Гибкость облачных сервисов позволяет справляться с серьезными нагрузками от конечных пользователей. Многие облачные сервисы предоставляют механизмы эластичности. Например, Amazon EC2 позволяет потребителям облака определять определенные правила эластичности для автоматического увеличения числа вычислительных экземпляров, когда приложение имеет массовый одновременный доступ. Больше вычислительных экземпляров может сбалансировать нагрузку и поддерживать большее количество одновременных запросов. На самом деле эластичность облака широко применяется в электронной коммерции, мобильных и веб-приложениях.

На рисунке 16.6 CLN используется в качестве примера, чтобы проиллюстрировать, как эластичность облака реагирует на огромное количество одновременных запросов пользователей. На рисунке показаны новые экземпляры облака EC2, когда количество запросов пользователей превышает определенный предел, при этом значение оси X показывает номер доступа пользователя, а ось Y показывает время ответа системы. Пример иллюстрирует разное количество запросов к CLN. Новый экземпляр будет добавлен для запуска CLN, когда количество одновременных пользователей увеличится и загрузка ЦП превысит 90%. Замечено, что при использовании большего количества вычислительных экземпляров может быть получен более высокий прирост производительности, а время отклика может быть сохранено на уровне около 4 секунд с помощью эластичных вычислительных ресурсов (обозначенных пятью экземплярами масштабирования). Эластичное автоматическое предоставление и высвобождение вычислительных ресурсов в значительной степени подготовило нас к реагированию на всплески одновременного доступа, в то же время снижая затраты за счет совместного использования вычислительных ресурсов с другими приложениями, когда нет всплесков одновременного доступа.

### 16.4.3.3 Пространственно-временная индексация

Правильный механизм индексации значительно повысит скорость извлечения данных. CLN построил специальный механизм индексации, чтобы сократить время выполнения запроса, тем самым уменьшив время ответа на запросы пользователей.



Рисунок 16.6 Сравнение производительности отклика CLN для экземпляров с одной, двумя и пятью загрузками и пяти экземпляров с автоматическим масштабированием. (Yang et al., 2011a.)

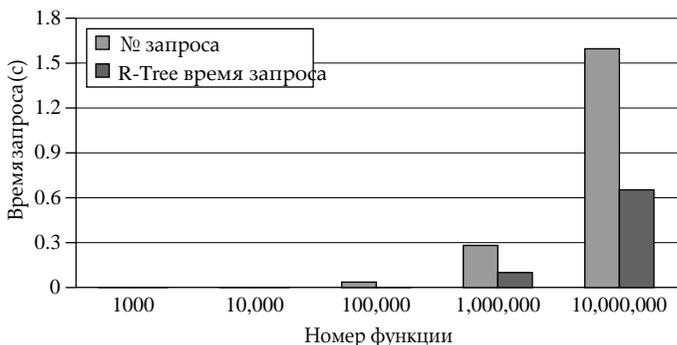


Рисунок 16.7 (См. цветную вставку) Пространственное индексирование для повышения скорости доступа.

Благодаря более короткому времени отклика CLH обрабатывает растущее число одновременных запросов конечных пользователей. На рисунке 16.7 показана производительность запросов с механизмом индексирования и без него. Механизм индексации значительно ускоряет выполнение запросов в Информационном центре GEOSS для обработки большого количества запросов конечных пользователей.

#### 16.4.4 Сохраняющиеся проблемы и будущие исследования

Синхронизация данных и производительность. Облачные сервисы могут предоставлять эффективные возможности, такие как услуги балансировки нагрузки и автоматического масштабирования, для повышения одновременной производительности приложений геолого-геофизических исследований. Однако многим приложениям необходимо часто динамически обновлять данные и контексты, что вызывает проблему несогласованности данных при настройке балансировки нагрузки и эластичности для запуска экземпляров в разное время. Кроме того, существует задержка при запуске нового экземпляра с использованием служб балансировки нагрузки и автоматического масштабирования, так как загрузка виртуальной машины может занять секунды или даже минуты. Следовательно, необходимы дальнейшие исследования этих проблем для улучшения производительности этих геопространственных приложений.

Использование пространственно-временных принципов - Пространственно-временные принципы управляют эволюцией физических и социальных явлений (Yang et al. 2011a). Распределение и запросы конечных пользователей зависят от пространственно-временных принципов, таким образом, имея определенные пространственно-временные шаблоны. Эти пространственно-временные шаблоны могут помочь нам лучше спроектировать механизм обновления данных, обслуживания системы и масштабирования вычислительных ресурсов, тем самым увеличивая способность справляться с одновременной интенсивностью. Кэширование памяти - это метод сохранения часто используемых данных в ОЗУ компьютера, позволяющий повысить производительность ввода-вывода. Однако ограниченное пространство ОЗУ является узким местом для механизма кэширования.

Суперкомпьютеры с высокой вычислительной мощностью и большим объемом оперативной памяти представляют собой потенциальное решение для устранения этого узкого места. Однако суперкомпьютеры обычно дороги, и для потребителей облака полезно разработать модель затрат, чтобы сбалансировать стоимость и требования к производительности системы. Кроме того, прогнозирование и предварительная выборка «наиболее часто используемых данных» могут улучшить скорость использования кэша, тем самым улучшая способность обрабатывать одновременную интенсивность.

## 16.5 ПРОСТРАНСТВЕННО-ВРЕМЕННАЯ ИНТЕНСИВНОСТЬ

Пространственно-временная интенсивность является фундаментальной для наук о Земле и вносит свой вклад в другие интенсивности (Yang et al. 2011a), что отражено в следующих примерах.

- **Пространственно-временные данные.** Большинство геолого-геофизических данных записываются как функция пространственно-временных измерений либо со статической пространственной информацией с определенной временной меткой, либо с изменяющимся временным и пространственным охватом (Terrenghi et al. 2010). Например, дневной диапазон температур для определенного места за последние 100 лет ограничен местоположением (местом) и временем (ежедневные данные за 100 лет).
- **Сбор данных.** Развитие сенсорных технологий расширило нашу способность проводить более точные измерения и своевременно получило лучшее пространственное покрытие (Goodchild 2007). Все наборы данных, записанные для наук о Земле, являются пространственно-временными либо явными (динамическими), либо неявными (статическими). Увеличение разрешения значительно увеличит объем данных и откроет гораздо больше геофизических принципов, которые предстоит открыть в результате научных исследований.
- **Природные явления и научные исследования.** Изучение геонаучных явлений описывается как пространство-время или геодинамика (Hornsby and Yuan 2008). В соответствующих геонаучных исследованиях, таких как науки об атмосфере и океане, пространство-время и геодинамика всегда были в центре областей исследований. И это ядро становится критически важным почти во всех областях последующего человеческого знания (Su et al. 2010).
- **Система реагирования на чрезвычайные ситуации - система реагирования в реальном времени** может включать в себя все виды интенсивности: (1) будут использоваться массивные пространственно-временные данные, которые исторически записываются или собираются в режиме реального времени, (2) данные будут обрабатываться и анализироваться с высокой вычислительной мощностью для поддержки принятия более эффективных решений и (3) возможность одновременного массового публичного доступа. Например, миллионы пользователей заходили на веб-сайт weather.com, чтобы быть в курсе развития урагана во время урагана Сэнди в 2012 году.

290 К. Хуан, З. Ли, К. Лю, Дж. Ся, Ю. Цзян, К. Сюй, К. Ян

**Таблица 16.1** Пространственно-временная интенсивность напрямую влияет на различные аспекты интенсивности обработки данных, вычислений и одновременного доступа

|                                       | <i>Инт. данных</i> | <i>Выч. инт.</i> | <i>Многопот. инт.</i> |
|---------------------------------------|--------------------|------------------|-----------------------|
| Пространственно-временные данные      | X                  | —                | —                     |
| Сбор информации                       | X                  | —                | —                     |
| Исследования явлений                  | X                  | X                | —                     |
| Реагирования на чрезвычайные ситуации | X                  | X                | X                     |

Примечание: «X» в ячейке означает, что задача имеет определенную интенсивность, например, пространственно-временные данные имеют характеристику интенсивности данных.

Четыре аспекта пространственно-временной интенсивности напрямую влияют на интенсивность данных, вычислений и одновременного доступа (таблица 16.1). Таким образом, вычислительные решения для трех других значений интенсивности также могут быть использованы для решения проблем, связанных с пространственно-временными интенсивностями.

## 16.6 КРАТКОЕ ИЗЛОЖЕНИЕ

В этой главе изложены грандиозные проблемы, с которыми геонауки сталкиваются в 21 веке, - интенсивность данных, вычисления, одновременный доступ и пространственно-временной анализ. Ян и др. (2011b) утверждают, что последние достижения облачных вычислений предоставляют потенциальное решение этих серьезных проблем. На примерах интенсивности данных, интенсивности вычислений, интенсивности одновременного доступа и пространственно-временной интенсивности мы проиллюстрировали, что облачные вычисления имеют решающее значение с точки зрения их возможностей: (a) обеспечивать обработку распределенных, неоднородных и больших геолого-геофизических данных (раздел 16.2); (b) поддержка обработки геолого-геофизических данных, требующих больших вычислительных ресурсов (Раздел 16.3); (c) позволяют своевременно реагировать на глобально распределенных и локально сгруппированных пользователей (Раздел 16.4), и (d) облегчать процесс пространственно-временной интенсивности и приложения (Раздел 16.5).

## 16.7 ПРОБЛЕМЫ

1. Каковы вычислительные проблемы для данных, вычислений, а также задач одновременной и пространственно-временной интенсивности?
2. Каковы общие решения проблем с интенсивностью данных?
3. Как облачные вычисления могут стать потенциальным решением проблем с интенсивностью данных? Какие еще проблемы?
4. Что такое параллельные вычисления? Объясните процесс интерполяции DEM с помощью параллельных вычислений.
5. Что представляет собой графический процессор (GPU)? Каковы общие шаги по использованию графического процессора для интерполяции DEM?

6. Обобщите, как облака облачных вычислений предоставляют потенциальное решение для решения проблем, связанных с интенсивностью вычислений. Какие еще проблемы?
7. Как эластичность используется для решения проблем одновременной интенсивности?
8. Как облачные вычисления могут предоставить потенциальное решение для решения одновременных проблем интенсивности? Какие еще проблемы?
9. Перечислите четыре примера, которые включают пространственно-временную интенсивность.

## СПИСОК ЛИТЕРАТУРЫ

- Armbrust, M., A. Fox, R. Griffith, A. D. Joseph et al. 2010. A view of cloud computing. *Communications of the ACM* 53, no. 4: 50–58.
- Armstrong, M. P., M. K. Cowles, and S. Wang. 2005. Using a computational grid for geographic information analysis: A reconnaissance. *The Professional Geographer* 57, no. 3: 365–375.
- Audenino, P., L. Rognant, J. M. Chassery, and J. G. Planes. 2001. Fusion strategies for high resolution urban DEM. In *Remote Sensing and Data Fusion over Urban Areas, IEEE/ISPRS Joint Workshop 2001*, pp. 90–94. IEEE.
- Buck, J. B., N. Watkins, J. LeFevre et al. 2011. SciHadoop: Array-based query processing in hadoop. In *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*, p. 66. ACM.
- Cartwright, D. E. 2000. *Tides: A Scientific History*. United Kingdom: Cambridge University Press.
- Donner, R., S. Barbosa, J. Kurths, and N. Marwan. 2009. Understanding the Earth as a complex system—Recent advances in data analysis and modelling in Earth sciences. *The European Physical Journal Special Topics* 174, no. 1: 1–9.
- Drebin, R. A., L. Carpenter, and P. Hanrahan. 1988. Volume rendering. In *ACM Siggraph Computer Graphics* 22, no. 4, pp. 65–74. ACM.
- Duffy, D. Q., J. L. Schnase, T. L., Clune, E. J. Kim, S. M. Freeman, J. H. Thompson, ... and M. E. Theriot. 2011. [http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20120009187\\_2012009164.pdf](http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20120009187_2012009164.pdf) (accessed August 9, 2013).
- Dumbill, E. 2012. What Is Big Data? An Introduction to the Big Data Landscape. <http://strata.oreilly.com/2012/01/what-is-big-data.html> (accessed January 25, 2013).
- Fox, G., S. H. Bae, J. Ekanayake, X. Qiu, and H. Yuan. 2010. Parallel data mining from multicore to cloudy grids. *High Speed and Large Scale Scientific Computing* 18: 311.
- Huang, Q. and C. Yang. 2011. Optimizing grid computing configuration and scheduling for geospatial analysis: An example with interpolating DEM. *Computers & Geosciences* 37, no. 2: 165–176.
- Huang, Q., C. Yang, K. Benedict et al. 2012. Using adaptively coupled models and high-performance computing for enabling the computability of dust storm forecasting. *International Journal of Geographic Information Science*. doi:10.1080/13658816.2012.715650.
- Huang, Q., C. Yang, K. Benedict et al. 2013. Enabling dust storm forecasting using cloud computing. *International Journal of Digital Earth*. doi:10.1080/17538947.2012.749949.

- Kumar, S. V., C. D. Peters-Lidard, Y. Tian et al. 2006. Land information system: An interoperable framework for high resolution land surface modeling. *Environmental Modelling & Software* 21, no. 10, 1402–1415.
- Lämmel, R. 2008. Google's MapReduce programming model—Revisited. *Science of Computer Programming* 70, no. 1, 1–30.
- Li, J., Y. Jiang, C. Yang, and Q. Huang. 2013. 59, 9: 78–89. Utilizing GPU and CPU for accelerating visualization pipeline. *Computers & Geosciences* 59, 9: 78–89.
- Li, Z., C. Yang, M. Sun et al. 2013. A high performance Web-based system for analyzing and visualizing spatiotemporal data for climate studies. In *Web and Wireless Geographical Information Systems*, pp. 190–198. Heidelberg: Springer Berlin.
- Li, Z., C. Yang, H. Wu, W. Li, and L. Miao. 2011. An optimized framework for seamlessly integrating OGC web services to support geospatial sciences. *International Journal of Geographical Information Science* 25, no. 4: 595–613.
- Ostermann, S., A. Iosup, M. N. Yigitbasi et al. 2010. A Performance Analysis of EC2 Cloud Computing Services for Scientific Computing. Lecture Notes of the Institute for Computer Sciences: Social Informatics and Telecommunications Engineering 34, no. 4: 115–131. doi: 10.1007/978-3-642-12636-9\_9.
- Schuster, A. 1897. On lunar and solar periodicities of earthquakes. *Proceedings of the Royal Society of London* 61, no. 369–370: 455–465.
- Singh, G., S. Bharathi, A. Chervenak et al. 2003. A metadata catalog service for data intensive applications. In *Supercomputing, 2003 ACM/IEEE Conference*, pp. 33–33. IEEE.
- Theoderidis, Y., M. Vazirgiannis, and T. Sellis. 1996. Spatio-temporal indexing for large multimedia applications. In *Multimedia Computing and Systems. Proceedings of the 3rd IEEE International Conference*, pp. 441–448.
- Wang, J. and Z. Liu. 2008. Parallel data mining optimal algorithm of virtual cluster. In *Fuzzy Systems and Knowledge Discovery. FSKD'08. 5th International Conference*, 5, pp. 358–362. IEEE.
- Yang, C., M. Goodchild, Q. Huang et al. 2011a. Spatial cloud computing—How can geospatial sciences use and help to shape cloud computing. *International Journal of Digital Earth* 4, no. 4: 305–329.
- Yang, C., R. Raskin, M. Goodchild, and M. Gahegan. 2010. Geospatial cyberinfrastructure: Past, present and future. *Computers, Environment and Urban Systems* 34, no. 4: 264–277.
- Yang, C., M. Sun, K. Liu et al. 2013. Contemporary computing technologies for processing big spatiotemporal data. In *Space-Time Integration in Geography and GIScience: Research Frontiers in the U.S. and China*. Mei-Po Kwan, Douglas Richardson, Donggen Wang and Chenghu Zhou (eds), Dordrecht: Springer (in press).
- Yang, C., H. Wu, Q. Huang, Z. Li, and J. Li. 2011b. Using spatial principles to optimize distributed computing for enabling the physical science discoveries. *Proceedings of the National Academy of Sciences* 108, no. 14: 5498–5503.
- Yee, K. P., K. Swearingen, K. Li, and M. Hearst. 2003. Faceted metadata for image search and browsing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 401–408.

- Zhang, Q., L. Cheng, and R. Boutaba. 2010. Cloud computing: State-of-the-art and research challenges. *Journal of Internet Services and Applications* 1, no. 1: 7–18. Zhao, H., S. Ai, Z. Lv, and B. Li. 2010. Parallel accessing massive NetCDF data based on MapReduce. In *Web Information Systems and Mining*, pp.425–431. Heidelberg: Springer Berlin.



## Глава 17

---

# Исследования облачной обработки данных для геонаук и приложений

*Чаовой Ян, Цюньбин Хуан, Чжипенг Гуй, Чжэньлун Ли, Чэнь Сюй, Юньфэн Цзян и Цзин Ли*

---

Облачные вычисления предоставляют возможности и решения, выходящие за рамки организаций, юрисдикций и континентов, для решения многих нерешенных глобальных проблем в 21 веке (NRC 2011b). Однако конечный успех облачных вычислений для геопространственных наук будет измеряться так, как если бы облачные вычисления могли ускорить геолого-геофизические исследования и улучшить работу геопространственных приложений с точки зрения управления, например, за счет экономии затрат (Luftman and Zadeh 2011). По-прежнему требуется много исследований и разработок, чтобы обеспечить решения, по крайней мере, с точки зрения видения, технологий и социальных аспектов (Yang, Xu, and Nebert 2013).

### 17.1 РАЗВИТИЕ ВИДЕНИЯ 21-ГО ВЕКА ДЛЯ ПРИЛОЖЕНИЙ ГЕОНАУКИ

Достижения в области компьютерных наук были вызваны различными требованиями к вычислениям в некомпьютерных областях. Развитие облачных вычислений - не исключение. Многие задачи в области наук о Земле 21 века предъявляют новые требования к облачным вычислениям. Следовательно, вполне вероятно, что геонауки станут основной движущей силой эволюции облачных вычислений. В этом разделе обсуждаются потребности в трех областях: фундаментальные исследования в области геолого-геофизических исследований, междисциплинарные геологические науки и приложения для геонаук.

#### 17.1.1 Запросы фундаментальной геопространственной науки

Фундаментальные исследования в области геолого-геофизических исследований находятся на переднем крае человеческого знаний и являются ключом к решению многих научных и прикладных задач (NRC 2012b). Например, изучение геологической истории Земли может помочь ученым найти решения, которые помогут прогнозировать многие стихийные бедствия, такие как землетрясения и цунами (NRC 2011a, 2011b). Прорыв в фундаментальных исследованиях требует от ученых всего мира обмена знаниями о собранных ими локализованных геопространственных данных. Облачные вычисления могут облегчить обработку данных, моделирование, обмен и совместную работу.

Необходимо изучить глобальную интеграцию различных ресурсов для облачных сервисов для поддержки глобальных научных открытий.

### **17.1.2 Интеграция геонаук с другими областями науки для новых открытий**

McGrath (2011) утверждал, что наиболее важные научные открытия происходят сейчас, но внутри, и особенно в, казалось бы, не связанных между собой областях науки. Интеграция научных областей потребует беспрецедентных инструментов и методологий для облегчения интегрированных научных открытий. Облачные вычисления могут способствовать интеграции научной области с вычислительным сервисом, выходящим за пределы каждой области. Определение потребностей вычислительной инфраструктуры для каждого домена и кросс-доменов поможет в планировании оптимизированных облачных служб для поддержки интеграции для новых открытий.

### **17.1.3 Видение приложения**

Видение приложений также является движущей силой технологических достижений. Например, концепция Gore Digital Earth способствовала развитию оборудования, программного обеспечения, управления информацией и визуализации и, в конечном итоге, привела к очень успешным Google Earth и Virtual Earth / Bing Maps. Продолжая концепции цифрового видения Земли, интеллектуальная Земля требует объединения данных сенсорных наблюдений и исторических данных в реальном времени для обеспечения интеллектуальной поддержки принятия решений при реагировании на чрезвычайные ситуации (Лю, Се и Пэн, 2009; Лю и др., 2010). Такое массовое слияние сенсорной информации и исторических данных в реальном времени создает серьезные проблемы для облачных вычислений.

Для моделирования виртуальной среды для геопространственных наук была предложена виртуальная географическая среда (VGE) (Lin, Chen, and Lu 2011) для решения этих проблем. В свою очередь, проект поставил множество задач в области визуализации и взаимодействия с пользователем для облачных вычислений для работы с большими данными и массовым одновременным доступом пользователей. Симуляторы глобального военного конфликта могут предоставить другое видение, которое приводит к одному из самых дорогостоящих антропогенных событий - боевым действиям для разрешения споров. Симулятор можно использовать для обучения солдат, помогающих принимать разумные решения. Симулятор может также предоставить механизм моделирования, чтобы помочь уменьшить потери и потери имущества в реальных боевых условиях (Tanase and Urzica 2009). Управление пространственно-временной информацией создает серьезные проблемы для надежности и своевременности облачных вычислений. Быстро разработанные карты реагирования на чрезвычайные ситуации, как описано в главе 1, помогут спасти сотни человеческих жизней и миллиарды долларов потерь активов. Поэтому важно, чтобы новый процесс картирования мог своевременно предоставлять данные, анализ и картирование. Существенные проблемы, существующие в рамках нового процесса картирования, должны быть решены с помощью новых картографических технологий (Meng 2011).

Облачные вычисления могут стать технологическим решением многих задач картографии.

Постоянное видение геопространственных областей поможет установить цель и стимулировать технологический прогресс.

## **17.2 ТЕХНОЛОГИЧЕСКИЕ ПЕРЕДАЧИ**

### **17.2.1 Оценка и выбор облака**

Оценка и выбор облачных сервисов - сложная задача для облачных клиентов. Чтобы решить эту проблему, необходимо рассмотреть следующие аспекты исследования:

- Критерии измерения облачных сервисов являются основой для измерения, оценки и выбора облачных сервисов. Хотя были предложены различные критерии измерения (Kulkarni 2012; Repschläger et al. 2011), эти критерии четко не определены. Например, хотя Национальный институт стандартов и технологий (NIST) изучил открытые проблемы и рекомендации (Badger et al. 2012), учредил Подгруппу RATAx Cloud Metrics<sup>1</sup> и стандартизировал критерии и связанные модели, необходимы дальнейшие усилия для определять согласованные, многократно используемые и операционные модели для поддержки всесторонних измерений облачных сервисов (NIST 2012).
- Рентабельность инвестиций (ROI) является важным определяющим фактором при выборе облака с точки зрения потребителей облака. Для расчета совокупной стоимости владения (TCO) и стоимости использования облака были разработаны модели расчета затрат<sup>2</sup> (Ли и др., 2009). Чтобы помочь клиентам понять их рентабельность инвестиций и сделать разумный выбор, следует разработать преднамеренные модели затрат, чтобы сделать точную оценку и анализ потенциального состава затрат и использования факторов дисбаланса.
- Сторонний аудит важен для обеспечения достоверного понимания производительности, надежности и согласованности облачных сервисов. CloudSleuth разработала платформу мониторинга под названием Global Provider View<sup>3</sup> для сбора и визуального анализа производительности и доступности многих популярных облачных сервисов. Он непрерывно отслеживает лучшие облачные сервисы повсюду миру, развертывая и запуская образец веб-приложения в каждой из облачных сервисов. Для широкого мониторинга с целью дальнейшей оценки архитектура сервисов мониторинга должна быть хорошо спроектирована, и для разных сценариев следует рассматривать различные примеры приложений.

---

<sup>1</sup> См. NIST на [http://collaborate.nist.gov/twiki-cloud-computing/bin/view/CloudComputing/RATax\\_CloudMetrics](http://collaborate.nist.gov/twiki-cloud-computing/bin/view/CloudComputing/RATax_CloudMetrics).

<sup>2</sup> См. Microsoft Windows Azure Platform TCO Calculator на <http://www.microsoft.com/brasil/windowsazure/tco/>.

<sup>3</sup> См. Global Provider View на <https://cloudsleuth.net/global-provider-view>.

- Чтобы помочь клиентам облачных вычислений понять преимущества и недостатки облачных услуг, а также сделать правильный выбор, необходимо изучить передовые принципы и модели выбора. На основе этих методов выбора следует разработать сложные и удобные для пользователя инструменты для консультирования по облачным технологиям (Goscinski et al. 2010; Martens et al. 2011) и веб-сайты<sup>1</sup>. Помимо сбора, сравнения и визуального анализа функций облачных сервисов (например, емкости, стоимости и отзывов пользователей), эти инструменты и веб-сайты должны иметь возможность давать рекомендации на основе требований и предпочтений потребителей. Например, Andrzejak (2010) предложил вероятностную модель для оптимизации денежных затрат, производительности и надежности с учетом требований пользователей и приложений и динамических условий. Эта модель может помочь потребителям оптимизировать ставки на спотовых экземплярах Amazon для достижения желаемых целей. Кроме того, для оптимизации выбора облачной службы на основе Соглашения об уровне обслуживания (SLA), стоимости, функций приложения и пространственно-временных характеристик пользователей приложений / облачных служб / данных и других свойств, также следует разработать алгоритмы оптимизации.

## 17.2.2 Управление ресурсами облачных сервисов

Облачные ресурсы (например, центры обработки данных и вычислительные центры) представляют собой значительные вложения в капитальные и текущие расходы. Оптимизированное управление облачными ресурсами и их использование не только улучшает использование ресурсов и производительность, но также сокращает бюджет, энергию и рабочую силу как для поставщиков облачных услуг, так и для потребителей облачных услуг.

Связывание ресурсов и фрагментация серьезно препятствуют использованию вычислительных ресурсов, а также увеличивают затраты на управление. Повышение гибкости сети и обеспечение соответствующих стимулов могут влиять на потребление ресурсов и улучшать использование вычислительных ресурсов (Greenberg et al. 2009). Для снижения затрат, связанных с сетями географически разнесенных центров обработки данных, следует также предложить совместную оптимизацию ресурсов сети и центра обработки данных и новые механизмы для состояния геораспределения.

Экономия электроэнергии, используемой в вычислительных ресурсах, важна для снижения глобального потребления энергии. Белоглазов и др. (2010) предложили энергоэффективную систему управления ресурсами для виртуализированных облачных центров обработки данных для снижения эксплуатационных расходов и обеспечения требуемого качества обслуживания (QoS). Экономия энергии может быть достигнута путем непрерывной консолидации виртуальных машин (VM) в соответствии с текущим использованием ресурсов, топологией виртуальной сети, установленной между VM, и тепловым состоянием вычислительных узлов.

<sup>1</sup> См. Cloud Computing Providers на <http://cloud-computing.findthebest.com/>.

Автоматизация управления виртуальными машинами может снизить трудозатраты и время для поставщиков облачных услуг. Для управления виртуализированной средой интеллектуальный автономный диспетчер ресурсов может отделить ресурсы от динамического размещения виртуальных машин. Используя подход программирования ограничений, менеджер может оптимизировать глобальную полезность, учитывая как степень выполнения SLA, так и эксплуатационные расходы..

### 17.2.3 Резервное копирование и синхронизация данных

Сервисы облачного хранения меняют способ доступа людей к данным и их хранения. Существует множество сервисов облачного хранения, таких как Amazon S3, Cloud Drive, Dropbox, Box, SugarSync, SkyDrive, CloudApp, Google Drive и Apple iCloud, как описано в главе 3, раздел 3.3. Услуги облачного хранилища предоставляют мощные возможности для резервного копирования и синхронизации между различными терминалами наших ежедневных данных и информации, таких как видео, музыка, электронная почта и документы. Однако резервное копирование и синхронизация данных становятся узким местом масштабируемости для научных приложений, позволяющих наилучшим образом использовать вычислительную мощность по требованию. В облаке можно легко развернуть несколько серверов для веб-приложений, чтобы сбалансировать запросы доступа. Однако есть трудности с одновременным доступом к данным. Следовательно, несколько серверов должны последовательно обращаться к хранилищу данных (Das, Agrawal, and Abadi 2009).

Резервное копирование и синхронизация данных - одна из важнейших исследовательских задач облачных вычислений. Дас, Агравал и Аббади (2009) предложили ElasTraS с целью решить проблему масштабируемости и эластичности транзакций базы данных в облаке. Vtable, Savage и Voelker (2009) представили систему Cumulus для эффективного выполнения резервного копирования файловой системы через Интернет, специально разработанную для работы с тонким облаком. Тока, Делл'Амико и Мичиарди (2010) изучали преимущества однорангового подхода для онлайн-приложений резервного копирования. Однако методологии резервного копирования и синхронизации данных с массивными геопространственными данными и частыми операциями требуют дальнейшего изучения, и поставщики облачных услуг должны интегрировать эти стратегии в свои облачные сервисы, которые могут быть легко настроены потребителями облака.

### 17.2.4 Совместимость

Функциональная совместимость должна опираться на стандарты, разработанные различными организациями, такими как OGC, OGF, NIST, ISO и IEEE, посредством проектирования архитектуры системы для поддержки совместного использования распределенных вычислительных ресурсов на разных уровнях (NIST 2012). Этим должны руководствоваться большие группы пользователей, домены приложений, поставщики и правительства для достижения необходимого уровня взаимодействия (Lee 2010).

При наличии множества моделей облачных сервисов (например, IaaS, PaaS и SaaS) и поставщиков (например, Amazon и Windows) возможность взаимодействия между различными облачными сервисами представляет большой интерес для потребителей, позволяющих в полной мере использовать преимущества инновационных технологий.

Однако отсутствие стандартов для взаимодействия облачных вычислений стало серьезным препятствием, сдерживающим эволюцию облачных вычислений. В настоящее время облачные провайдеры используют разные методы реализации и интерфейсы. Например, Amazon использует Xen в качестве гипервизора для своей платформы EC2, а виртуализация Windows Azure основана на Hyper-V 1, который ранее был известен как виртуализация Windows Server. Эти различия приводят ко многим проблемам взаимодействия, таким как привязка к поставщику и переносимость приложений.

К счастью, несколько организаций начинают решать эти вопросы с разных сторон и направлений. NIST тесно сотрудничает с сообществами по стандартизации, частным сектором и другими заинтересованными сторонами, включая федеральные агентства, по работе над дорожной картой и эталонной архитектурой стандартов облачных вычислений (Hogan et al. 2011). Постоянно обновляемый «Перечень стандартов, относящихся к облачным вычислениям», составляется рабочей группой NIST по стандартам облачных вычислений, в которую входят добровольцы из промышленности, правительства и научных кругов. В этом перечне стандартов собраны протоколы, определения и стандарты самого высокого уровня, которые широко применяются в сценариях использования облачных вычислений. разработка протоколов управления облачными ресурсами. Другие организации включают рабочую группу Open Group по облачным вычислениям, 4 форума Open Grid, 5 Open Cloud Consortium, 6 и Cloud Security Alliance.<sup>7</sup> Несмотря на некоторый прогресс, достигнутый в вопросах взаимодействия с облаками, больше совместных усилий со стороны правительства, научных кругов и большого облака продавцы требуются.

В связи с насущной потребностью в интероперабельной среде облачных вычислений и под руководством NIST и других стандартных организаций, будущие исследования, направленные на решение проблемы совместимости с облаками, включают, но не ограничиваются:

- Разработка и совершенствование стандартов для поддержки взаимодействия всех облачных моделей, от стека нижнего уровня IaaS до стека высокого уровня SaaS. Для конкретных моделей облачных сервисов требуются определенные стандарты.
- Разработка стандартных API-интерфейсов-посредников, позволяющих потребителям облака использовать, управлять, сравнивать и интегрировать облачные сервисы от разных поставщиков облачных услуг.

<sup>1</sup> См. Hyper-V на <http://en.wikipedia.org/wiki/Hyper-V>.

<sup>2</sup> См. NIST на <http://collaborate.nist.gov/twiki-cloud-computing/bin/view/CloudComputing/StandardsInventory>.

<sup>3</sup> См. DMTF на <http://www.dmtf.org/standards>.

<sup>4</sup> См. The Open Group на <http://www.opengroup.org/getinvolved/workgroups/cloudcomputing>.

<sup>5</sup> См. Open Grid Forum на <http://www.ogf.org/>.

<sup>6</sup> См. Open Cloud Consortium на [www.opencloudconsortium.org](http://www.opencloudconsortium.org).

<sup>7</sup> См. Cloud Security Alliance на <https://cloudsecurityalliance.org/>.

Например, приложения, созданные на основе различных облачных сервисов, могут использоваться совместно с помощью стандартного API-интерфейса посредника.

- Разработка протоколов управления облачными ресурсами и механизмов безопасности для облегчения взаимодействия с облаком. Например, уровни оркестрации могут использоваться для построения бизнес-процессов и рабочих процессов с использованием облачных сервисов, предоставляемых различными поставщиками облачных услуг (Parameswaran and Chaddha 2009).

### **17.2.5 Новая визуализация и интерактивные системы**

Визуализация и интерактивные системы внесли большой вклад в поддержку научных исследований крупномасштабных многомерных данных. Сегодняшняя инфраструктура облачных вычислений и хранилище данных значительно расширили наши возможности по выполнению ресурсоемких моделирования и симуляций, а также по управлению объемными распределенными данными. Кроме того, они предложили новые возможности и задачи для разработки визуализационных и интерактивных систем. Хотя для облегчения распределенной визуализации массивных данных были спроектированы и разработаны различные методы, техники и инструменты, эти подходы могут быть неприменимы в среде облачных вычислений из-за сложности облачной инфраструктуры и вспомогательной инфраструктуры. Дальнейшие исследования должны быть сосредоточены на:

- Настройка существующих алгоритмов и подходов визуализации с помощью облачных сервисов для поддержки сверхмасштабной интерактивной визуализации, когда интенсивность вычислений всегда является узким местом. Например, Amazon EC2 предоставляет экземпляры кластера с графическими процессорами (GPU), которые можно адаптировать для визуализации и интерактивных систем.
- Изучение пространственно-временных принципов управления реализациями визуализации и интерактивных систем с облачными сервисами. Для обеспечения руководства по проектированию системного уровня следует проводить всесторонние оценки.
- Разработка адаптивного рабочего процесса, который может наилучшим образом использовать облачные ресурсы, от предварительной обработки данных до окончательного отображения, в облачной среде. Например, вопрос о том, как распределять задачи интерактивного рендеринга на основе ресурсов облачных вычислений, требует дальнейшего изучения.
- Создание службы удаленной облачной визуализации, позволяющей пользователям из любого места интерактивно получать доступ к данным, изображениям, видео и приложениям как службам. Сервис должен поддерживать визуализацию на лету частичных или промежуточных результатов анализа данных. Цель состоит в том, чтобы потребителям облака не нужно было беспокоиться о проблемах и ограничениях, вызванных различными операционными системами, физическим расположением машин и их возможностями (Tanahashi et al. 2010).

## 17.2.6 Надежность и доступность

Надежность и доступность означают возможность эффективного доступа к среде из разных регионов. И то, и другое сложно достичь в среде облачных вычислений, где большинство компонентов распределены и управляются независимо. Даже крупные облачные провайдеры не смогли преодолеть эти проблемы. В 2012 году у двух первых облачных провайдеров, Amazon и Google, было несколько отключений облачных сервисов, что привело к перебоям в обслуживании из-за сетевых проблем. Например, в канун Рождества проблемы с AWS Amazon в Северной Вирджинии обрушились на несколько популярных офлайн-сервисов, включая Netflix<sup>1</sup>.

Существует две возможные причины недоступности и ненадежности службы, в том числе (1) серверы, состоящие из нескольких жестких дисков, модулей памяти, сетевых карт и процессоров, могут выйти из строя, даже если они тщательно спроектированы (Venkatesh et al. 2010), и (2) услуги предоставляются через Интернет и поэтому уязвимы для сбоев сети. Недостаточная надежность может также привести к большим финансовым потерям для бизнеса для некоторых пользователей, особенно для крупных корпоративных пользователей.

Многие исследовательские усилия все чаще направляются на сокращение количества отказов оборудования и сети для повышения надежности облачных вычислений. Например, Venkatesh et al.(2010) представили модель упреждающего прогнозирования отказов оборудования для физической машины, что может привести к своевременному перемещению рабочей нагрузки и данных с такого сервера во избежание любых возможных сбоев в обслуживании. Чтобы завоевать уверенность крупных предприятий в переходе стилей вычислений на облачные вычисления, необходимо приложить большие усилия для изучения стратегий прогнозирования сбоев облачных систем, реагирования и восстановления с точки зрения как поставщиков облачных услуг, так и потребителей облачных услуг.

## 17.2.7 Моделирование в реальном времени и доступ

Моделирование и доступ в реальном времени необходимы для различных типов поддержки принятия решений, от реагирования на чрезвычайные ситуации до повседневной жизни человека. Кроме того, новые устройства, такие как планшеты и смартфоны, все чаще используются населением. Это делает систему реагирования в реальном времени более доступной, чем когда-либо прежде. С другой стороны, это также создает огромную рабочую нагрузку для облачных сервисов, особенно во время зарождающегося периода времени, когда возможны массовые запросы (Huang et al. 2013). Облачные вычисления предоставляют большие возможности для поддержки таких систем реального времени с требованием быстрого реагирования. Например, Charalampos et al. (2010) внедрили систему управления медицинской информацией с использованием облачных вычислений, которая позволяет хранить, обновлять и извлекать электронные медицинские данные.

<sup>1</sup> См. Forbes на <http://www.forbes.com/sites/kellyclay/2012/12/24/amazon-aws-takes-down-netflix-on-christmas-eve/>.

Необходимо провести дополнительные исследования, включая моделирование на основе теории и многомасштабное, многокомпонентное моделирование, а также возможности интерактивной визуализации с большим объемом данных как для платформ облачных вычислений, так и для приложений (NRC 2011a, b).

## **17.3 СИНЕРГИСТИЧЕСКОЕ РАЗВИТИЕ СОЦИАЛЬНЫХ НАУК И ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ**

### **17.3.1 Управление облаком**

NIST (2011) определил пять характеристик облачных вычислений, чтобы отразить особенности облачных вычислений с разных точек зрения (Yang et al. 2011). Обычным требованием для реализации этих характеристик является автоматизация процесса управления в инфраструктуре облачных вычислений. На операционном уровне вмешательство пользователя должно быть сведено к минимуму, чтобы обеспечить автономное саморегулирование облачных вычислений. Саморегулирование требует снятия существующих организационных ограничений для поддержки автоматической обработки (Choi and Lee 2010). На технологическом уровне стандарты должны создаваться и поддерживаться различными службами. Больше трудностей связано с человеческим фактором; например, могут ли потребители облака доверять облачной службе для размещения своих данных и приложений?

### **17.3.2 Облачный охват**

Облачные вычисления становятся потенциальным вычислительным решением для локальных, региональных и глобальных проблем. Решение будет прогрессивным процессом и потребует целенаправленного взаимодействия с общественностью. Например, облачные вычисления сначала рекламировались как первопроходцы в решении всех вычислительных задач, но Amazon потребовалось четыре года (с 2006 по 2010), чтобы разработать свои предложения для высокопроизводительных вычислений. Понятие коммунальных вычислений заставляет общественность или даже некоторых лиц, принимающих решения в области информационных технологий, полагать, что нам нужно всего от 100 до 200 вычислительных центров в Соединенных Штатах, и все вычислительные потребности могут быть удовлетворены из этих центров. Но факт в том, что облачные вычисления будут ограничены пропускной способностью сети, а пространственно-временные принципы вынуждают нас объединять наши данные с вычислительной инфраструктурой, поэтому необходимо учитывать близость между облачными вычислениями и данными, пользователями, проблемами и приложениями. Еще одно недооцененное влияние - безопасность. Хотя облачная система может быть более защищенной от атак, общая безопасность данных, конфиденциальность и удовлетворение государственных потребностей иногда не позволяют правительствам внедрять общедоступное облако. Следовательно, важно донести до общественности правильное сообщение об облачных вычислениях, чтобы они не были чрезмерно загружены или недооценены.

### 17.3.3 Безопасность и правила

Безопасность и правила касаются использования облачных вычислений для определенной информации и приложений; например, приложения по труду и социальному обеспечению (Lu 2010). В настоящее время безопасность и правила являются одними из самых больших проблем, когда частные лица или предприятия рассматривают возможность внедрения облачных сервисов. В традиционной ИТ-среде потребители имеют полный контроль над своими ИТ-системами, включая (1) расположение центра обработки данных, (2) разрешение на доступ и использование этой инфраструктуры, данных и услуг и (3) стратегии безопасности. С облачными услугами предприятия теряют физический контроль над ИТ-системами. Например, предприятию может быть запрещено размещать свои данные и информацию в другой стране. Таким образом, при выборе облачной службы компании должны обращать внимание на доступные регионы, предоставляемые поставщиками облачных услуг, и проверять, нет ли конфликта между использованием этих облачных служб и правилами безопасности. Например, все федеральные ИТ-приложения должны соответствовать определенным требованиям безопасности

Федеральной программы управления рисками и авторизацией (FedRAMP).<sup>1</sup> FISMA требует от федеральных агентств разработки, документирования и внедрения системы защиты информации для своих данных и инфраструктуры. В дополнение к национальным нормам некоторые компании также должны соблюдать отраслевые правила и стандарты. Например, все организации, большие или малые, в США, должны соблюдать Закон Сарбейнса-Оксли (SOX).<sup>2</sup> Поскольку клиент может войти в систему из любого места для доступа к данным и приложениям, возможна конфиденциальность клиента. Могут быть скомпрометированы в среде облачных вычислений. Могут применяться несколько механизмов, включая, например, использование методов аутентификации, таких как имена пользователей и пароли, и каждый пользователь может получить доступ только к данным и приложениям, относящимся к его или ее работе.

Для данных и сервисов необходимы улучшения уровня безопасности и конфиденциальности облачных сервисов, включая безопасную веб-среду для разработки приложений (O'Leary and Kaufman 2011), создание международных лицензионных соглашений или исключений для обеспечения хранения контролируемых экспортом технических данных облако является безопасным и защищенным (Schoorl 2011), а также защищает конфиденциальные данные, конфиденциальность и системы, сохраняя при этом дух совместного использования облачных вычислений. В таких обстоятельствах очень важно, чтобы поставщики облачных услуг позволяли потребителям или третьим сторонам проверять политику безопасности и конфиденциальности и проверять ее полноту и действительность. Для обеспечения конфиденциальности данных и информации необходимо провести дополнительные исследования, такие как решения для шифрования и дешифрования данных (Patil and Behal 2012).

<sup>1</sup> См. GSA на <http://www.gsa.gov/portal/category/102371>.

<sup>2</sup> См. the Sarbanes-Oxley Act на <http://www.soxlaw.com>.

### 17.3.4 Глобальное сотрудничество

Глобализация, частично вызванная цифровыми технологиями, изменила ландшафт исследовательского сотрудничества. Сотрудничество в глобальном масштабе имеет жизненно важное значение для решения многих проблем, таких как изменение климата, бедность и устойчивость. Хотя эти проблемы проявляются локально, они имеют глобальные последствия. Например, компании, стремящиеся максимизировать свою прибыль, могут перенести свои производственные линии в места с более низкими затратами на рабочую силу и создать там развитие, а также загрязнение окружающей среды. Некоторое загрязнение может повлиять на регионы мира. Следовательно, глобальное сотрудничество необходимо для обеспечения эффективности решения и может быть единственным способом решения. Облачные вычисления зародились как многообещающая парадигма совместной сети исследований для поддержки глобального сотрудничества по нескольким направлениям: (1) интеграция нескольких научных областей через общую информационную инфраструктуру (Yang et al. 2010). Интегрированная среда упрощает обмен данными, что имеет решающее значение для научных исследований на глобальном уровне. (2) Облачные вычисления могут принести новые возможности, начиная с определенного направления Интернета вещей (IoT), чтобы всегда знать, где все находится, до продвинутых ГИС-наук. Интернет вещей стал новым способом сбора данных и породил новые типы данных для исследований в области мониторинга окружающей среды и городского планирования. Облачные вычисления создают новый механизм для подключения граждан через повсеместные вычислительные устройства. Новый механизм продемонстрировал свою эффективность в таких областях, как сбор средств и участие общественности в ликвидации последствий стихийных бедствий. Повсеместный Интернет вещей и совместные сети граждан могут помочь обнаруживать и отслеживать природные и социальные события. В эпоху больших данных визуальная аналитика стала эффективным методом обнаружения закономерностей. Этот процесс требует больших объемов данных и вычислений. Облачные вычисления уже продемонстрировали способность соответствовать этим требованиям (Yang et al. 2011b). В конце концов, нам нужно научить наше следующее поколение решать проблемы устойчивого развития. Облачные вычисления предлагают новое направление в образовании, при котором ресурсы могут совместно использоваться глобально, знания могут распространяться глобально, а человеческий интеллект может накапливаться для решения как локальных, так и глобальных проблем (Goodchild 2010). Craglia et al. (2012) также подчеркнули важность глобального сотрудничества в свете многих достижений в области информационных технологий, инфраструктур данных и наблюдения Земли. Очень важно развивать серию сотрудничества на глобальном уровне, чтобы воплотить видение в реальность.

### 17.4 КРАТКОЕ ИЗЛОЖЕНИЕ

В этой главе рассматривается потенциал облачных вычислений в контексте решения задач 21 века и потребности в исследованиях для развития облачных вычислений в следующем десятилетии. Раздел 17.1 знакомит с эволюцией миссий, необходимых для приложений геолого-геофизических исследований.

В разделе 17.2 представлены технологические достижения, необходимые для облачных вычислений для поддержки геолого-геофизических исследований. А в Разделе 17.3 представлены необходимые достижения в области социальных наук.

## 17.5 ПРОБЛЕМЫ

1. Приведите пример проблем геонауки и соответствующих потребностей в открытиях.
2. Приведите пример, чтобы проиллюстрировать, как геонаука взаимодействует с другими областями науки.
3. Каковы видения науки Земли и их приложений? Какую роль видения играют в облачных вычислениях?
4. Перечислите четыре аспекта облачных вычислений, которые нуждаются в улучшении, и обсудите потенциальные направления их достижения.
5. Как облачные вычисления могут способствовать развитию социальных наук?
6. Как достижения социальных наук могут помочь эволюции облачной обработки данных?

## СПИСОК ЛИТЕРАТУРЫ

- Andrzejak, A., D. Kondo, and D. P. Anderson. 2010. Exploiting non-dedicated resources for cloud computing. In *Network Operations and Management Symposium (NOMS)*, pp. 341–348. IEEE.
- Badger, L., T. Grance, R. Patt-Corner, and J. Voas. 2012. Cloud computing synopsis and recommendations. *NIST Special Publication* 800, 146.
- Beloglazov A. and R. Buyya. 2010. Adaptive Threshold-Based Approach for Energy-Efficient Consolidation of Virtual Machines in Cloud Data Centers. *Proceedings of the 8th International Workshop on Middleware for Grids, Clouds and e-Science (MGC 2010)*, November 29–December 3. Bangalore, India. ACM.
- Choi, S. and G. Lee. 2010. 3D viewer platform of cloud clustering management system: Google Map 3D. *Communication and Networking*: 218–222.
- Craglia, M., K. de Bie, D. Jackson et al. 2012. Digital Earth 2020: Towards the vision for the next decade. *International Journal of Digital Earth* 5, no. 1: 4–21.
- Das, S., D. Agrawal, and A. E. Abbadi. 2009. Elastras: An elastic transactional data store in the cloud. *USENIX HotCloud* 2.
- Goodchild, M. F. 2012. Twenty years of progress: GIScience in 2010. *Journal of Spatial Information Science* 1: 3–20.
- Goscinski, A. and M. Brock. 2010. Toward dynamic and attribute based publication, discovery and selection for cloud computing. *Future Generation Computer Systems* 26, no. 7: 947–970.
- Greenberg, A., J. Hamilton, D. A. Maltz, and P. Patel. 2008. The cost of a cloud: Research problems in data center networks. *ACM SIGCOMM Computer Communication Review* 39, no. 1: 68–73.
- Hogan, M., F. Liu, A. Sokol, and J. Tong J. 2011. NIST cloud computing standards roadmap. *NIST Special Publication* 35.

- Huang, Q., C. Yang, K. Benedict, S. Chen, A. Rezgui, and J. Xie. 2013. Utilize cloud computing to support dust storm forecasting. *International Journal of Digital Earth*. doi:/10.1080/17538947.2012.749949.
- Huang, Q., C. Yang, D. Nebert, K. Liu, and H. Wu. 2010. Cloud computing for geosciences: Deployment of GEOSS Clearinghouse on Amazon's EC2. In *Proceedings of the ACM SIGSPATIAL International Workshop on High Performance and Distributed Geographic Information Systems*, pp. 35–38. ACM.
- IEEE. 2011. IEEE technology time machine symposium on technologies beyond 2020, *TTM 2011*, p. 46.
- Kim, I. H., M. Tsou et al. 2013. Enabling digital earth simulation models with cloud computing or grid computing? Two approaches to support web GIS simulation frameworks. *International Journal of Digital Earth* 6, no. 4: 383–403.
- Kulkarni, G. 2012. Cloud computing—Software as a service. *International Journal of Cloud Computing and Services Science (IJ-CLOSER)* 1, no. 1: 11–16.
- Lee, C. A. 2010. A perspective on scientific cloud computing. In *Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing*, pp. 451–459. ACM.
- Li, B., J. Li, J. Huai, T. Wo, Q. Li, and L. Zhong. 2009 (September). EnaCloud: An energy-saving application live placement approach for cloud computing environments. In *Cloud Computing. CLOUD'09, IEEE International Conference*, pp. 17–24. IEEE.
- Lin, H., M. Chen, and G. Lu. 2011. Virtual geographic environment: A workspace for computer-aided geographic experiments. *Annals of AAG*. doi:10.1080/00045608.2012.689234.
- Lin, L., L. De-Ren, Y. Bo-Xiong, and L. Wan-Wu. 2010. Smart planet based on geomatics. In *Multimedia Technology (ICMT), International Conference*, pp. 1–4.
- Liu, F., J. Tong, J. Mao, R. Bohn, J. Messina, L. Badger and D. Leaf. 2010. NIST cloud computing reference architecture. *NIST Special Publication*, 500, 292.
- Liu, C., Z. Xie, and P. Peng. 2009. A discussion on the framework of a smarter campus. In *Intelligent Information Technology Application. IITA. 3rd International Symposium*, 2, pp. 479–482.
- Lu, X. 2010. Service and cloud computing oriented Web GIS for labor and social security applications. In *Information Science and Engineering (ICISE), 2010 2nd International Conference*, pp. 4014–4017.
- Luftman, J. and H. S. Zadeh. 2011. Key information technology and management issues 2010–11: An international study. *Journal of Information Technology* 26, no. 3: 193–204.
- Martens, B. and F. Teuteberg. 2011. Risk and Compliance Management for Cloud Computing Services: Designing a Reference Model. In *AMCIS*.
- McGrath, E. 2011. *The Most Important Scientific Discovery of All Time: McGrath's Hypothesis: And the Scientific Revolution Going on Right Now*. Strategic Book Publishing, ISBN-13: 978-1612044897, p. 146.
- Meng, L. 2011. Cartography for everyone and everyone for cartography—Why and how? *Kartographische Nachrichten* 61, no. 5: 246–253.
- NIST. 2011. The NIST Definition of Cloud Computing. <http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf> (accessed December 29, 2012).

- NIST. 2012. NIST Cloud Computing Reference Architecture Cloud Service Metrics Description (Draft). [http://collaborate.nist.gov/twiki-cloud-computing/pub/CloudComputing/RATax\\_CloudMetrics/RATAX-CloudServiceMetricsDescription-DRAFT-v1.1.pdf](http://collaborate.nist.gov/twiki-cloud-computing/pub/CloudComputing/RATax_CloudMetrics/RATAX-CloudServiceMetricsDescription-DRAFT-v1.1.pdf) (accessed March 12, 2013).
- NRC. 2011. *Grand Challenges in Earthquake Engineering Research: A Community Workshop Report*. Washington DC: National Academies Press, p. 23.
- NRC. 2011b. *Assessment of Impediments to Interagency Collaboration on Space and Earth Science Missions*. Washington DC: National Academies Press, p. 67.
- NRC. 2012. *International Science in the National Interest at the U.S. Geological Survey*. Washington DC: National Academies Press, pp. 161.
- O'Leary, M. A. and S. Kaufman. 2011. MorphoBank: Phylophenomics in the "cloud." *Cladistics* 27, no. 5: 529–537.
- Papamanthou, C., R. Tamassia, and N. Triandopoulos. 2010. Optimal authenticated data structures with multilinear forms. In *Pairing-Based Cryptography-Pairing*. Berlin, Heidelberg: Springer, pp. 246–264.
- Parameswaran, A. V. and A. Chaddha. 2009. Cloud interoperability and standardization. *SETlabs Briefings* 7, no. 7: 19–26.
- Patil, A. and S. Behal. 2012. RSA algorithm achievement with Federal Information Processing Signature for Data Protection in cloud computing. *International Journal of Computers & Technology* 3, no. 1: 34–38.
- Repschläger, J., S. Wind, R. Zarnekow, and K. Turowski. 2011 (September). Developing a Cloud Provider Selection Model. In *EMISA*, pp. 163–176.
- Schoorl, J. A. 2011. Clicking the export button: Cloud data storage and U.S. dual-use export controls. *Geo. Wash. L. Rev.* 80: 632.
- Tanase, C. and A. Urzica. 2009. Global military conflict simulator. *Studies in Computational Intelligence* 237: 313–318.
- Tanahashi, Y., C. K. Chen, S. Marchesin, and K. L. Ma. 2010. An interface design for future cloud-based visualization services. In *Cloud Computing Technology and Science (CloudCom), 2010 IEEE 2nd International Conference*, pp. 609–613. IEEE.
- Toka, L., M. Dell'Amico, and P. Michiardi. 2010. Online data backup: A peer-assisted approach. In *Peer-to-Peer Computing (P2P), IEEE 10th International Conference*, pp. 1–10.
- Van, N. H., D. F. Tran, and J. M. Menaud. 2009. Autonomic virtual resource management for service hosting platforms. In *Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing, IEEE Computer Society*, pp. 1–8.
- Vrable, M., S. Savage, and G. M. Voelker. 2009. Cumulus: File system backup to the cloud. *ACM Transactions on Storage (TOS)* 5, no. 4: 14.
- Wen, Y., M. Chen, G. Lu, H. Lin, L. He, and S. Yue. 2013. Prototyping an open environment for sharing geographical analysis models on cloud computing platform 6, no. 4: 356–382.
- Yang, C., M. Goodchild, Q. Huang et al. 2011a. Spatial cloud computing: How can the geospatial sciences use and help shape cloud computing? *International Journal of Digital Earth* 4, no. 4: 305–329.
- Yang, C., Y. Xu, and D. Nebert. 2013. Redefining the Possibility of Digital Earth and Geosciences with Spatial Cloud Computing, *International Journal of Digital Earth* 6, no. 4: 297–312.

- Yang, C., D. Nebert, and D. R. Fraser Taylor. 2011b. Establishing a sustainable and cross-boundary geospatial cyberinfrastructure to enable polar research. *Computers & Geosciences* 37, no. 11: 1721–1726.
- Yang, C., R. Raskin, M. Goodchild, and M. Gahegan. 2010. Geospatial cyberinfrastructure: Past, present and future. *Computers, Environment and Urban Systems* 34, no. 4: 264–277.
- Yue, P., H. Zhou, J. Gong, and L. Hu. 2013. Geoprocessing in cloud computing platforms—A comparative analysis. *International Journal of Digital Earth* 6, no. 4, 404–425.



---

# Индекс

---

## **а**

- Amazon Elastic Cloud Computing, 265
- Amazon Elastic MapReduce (EMR), 29
- Образ машины Amazon (AMI), 55, 85, 148
- Amazon S3, 299
- Amazon Simple E-mail Service, 182
- Amazon Web Services (AWS), 55–66, 181–187
- Amazon CloudFront, 183
  - прикладной уровень, 182
  - архитектура, 181–182
  - авторизация доступа к сети, 55–57
  - Бесклассовая междоменная маршрутизация, 57 совместимость, 183
  - стоимость, 184–185
  - создать AMI из запущенного экземпляра, 65
  - межсервисный уровень, 182
    - развертывание приложения, 65
  - развертывание и интерфейс, 183 Elastic Beanstalk, 183
  - неудач, 187
  - обратная связь от сообщества, 186–187
  - брандмауэр, 56, 63
  - общие характеристики EC2, 182–185
  - Программное обеспечение ГИС как услуга, 115 Приложение Hello Cloud, 63
  - HTTP-доступ, 57
  - гипервизоры, 183
  - слой строительных блоков инфраструктуры, 182
  - установить и настроить веб-сервер, 63
  - запустить экземпляр, 57
  - список основных клиентов, 185–186
  - авторизация в экземпляре, 60
  - основные пользователи и общие комментарии, 185–187
  - созревание, 186
  - Модель экземпляров по требованию, поддерживает 184 ОС, 184
  - уровень физической инфраструктуры, 181–182
  - слой строительных блоков платформы, 182 файла с открытым ключом, 61
  - Файлы PuTTY, 61
  - надежность, 183
  - Удаленный рабочий стол, 55
  - Модель зарезервированных экземпляров, масштабируемость 184, 182–183 Secure Shell, 55
  - безопасность, 268
  - зарегистрироваться в Amazon AWS, модель 55 спотовых экземпляров, 184
  - уровень инструментов и API, 182
  - передать файл Hello Cloud на экземпляр, 63
  - сложность использования, 187
  - AMI; См. Образ машины Amazon
  - Apache Tomcat, 266
  - Apache VCL, 30
  - Apple iCloud, 299
  - Интерфейс прикладного программирования (API), 13
  - ArcGIS в облаке, 111–124
  - ArcGIS Online, 112–114 ArcGIS for Server, 114–115 AMI ArcGIS Server, 114
  - ArcGIS Server Cloud Builder, 114
  - ArcLogistics, 115
  - Карты Bing, 113

## 312 Индекс

---

- Business Analyst Online, 115  
Community Analyst, 115–116  
контентных сервисов, 113  
функций, 113–114  
геокодирование, 113  
геометрия, 114  
геопоиск, 113  
Централизованное управление услугами ГИС и доставка, 114  
Программное обеспечение ГИС как услуга, 115–116  
размещенных сервисов, 113  
имиджевые услуги, 113  
интеграция с предприятием приложения, 114  
артографические сервисы, 113  
мобильный ГИС-сервис, 116  
ответ по запросу для карты ГИС-инструменты, 114  
проблемы, 123  
маршрутизация, 113  
сводка раздела, сводка 116–117, сводка 122–123  
сервисы топографических карт, 113  
вариантов использования, 117–122  
добавить данные, 118  
добавить слои, 118  
Карта аэрофотоснимков, наводнение городского совета Брисбена, 121  
Обычное рабочее изображение, 120–121  
выбрать базовые карты, 118  
National Geographic World Карта, 121  
Государственные парки Пенсильвании Зритель, 121  
опубликовать и поделиться, 119  
региональный анализ штата Орегон с использованием ArcGIS Online, 117–119  
сохранить карту, 119  
сводка раздела, 122  
модификация символа, 119  
варианты использования ArcGIS for Server, 120–121  
Карта улиц мира, 121  
зачем географической информационной системе нужно облако, 111–112  
облачная ГИС, 111  
общих рабочих изображений приложение, 112  
Географическая информационная система, 111
- Примеры ГИС, которым нужно облако, 112  
традиционное программное обеспечение, 111  
ArcGIS Server, 266  
Архитектура; См. Облачные вычисления архитектура, концепции и характеристики ArcLogistics, 115  
Полномочия на эксплуатацию (ATO), 268  
AWS; См. Amazon Web Services Azure; См. Windows Azure В  
Рекомендации по сравнительному анализу, открытые-источник, 236  
сравнения решений IaaS, 237–238  
определенные действия пользователя и интерфейсы, 239  
Большие данные, 276–280  
наличие, 7  
способности принимать, 4  
вызов, 24  
CLN Challenge, 126  
пример с Climate @ Home, 277  
разрыв между MapReduce и научные данные, 279–280  
Распределенная визуализация на базе графического процессора в облачных сервисах, 278  
распределенных файловых систем Hadoop, 278  
представлений, 276–277  
задач управления, 276  
метаданных, пространственно-временные индекс и распределенный файл система, 277  
доступ по запросу, 279  
параллельные вычисления с облаком услуги, 278  
пиксельная визуализация, обработка 276–277, 98, 276  
оставшиеся проблемы и будущее исследования, 278–280  
масштабируемость управления, 156  
решений, 277–278  
хранение и вычисления вместимость, 279  
технологии хранения, 41  
передача, 15  
конечная цель, 276  
задача визуализации, 276  
Веб-сервис обработки, 29

- Карты Bing, 113, 154 Сервер BOINC, 144, 149–151  
разрешить доступ к сети, 149  
настроить эластичное блочное хранилище, 150  
настроить баланс нагрузки или масштабируемость, 151  
создать проект BOINC и настроить BOINC сервер, 150  
создать рабочие единицы и запустить проект, 151  
установить серверное программное обеспечение BOINC, 150  
установить пакеты программного обеспечения, 150  
запустить экземпляр, 149  
загрузите изображение ModelE VirtualBox и файлы конфигурации ModelE в экземпляр, 150  
Приложения без полей, 98  
Коробка, 299  
Бизнес-аналитик онлайн, 115  
С  
CacheBench, 244  
Служба каталогов для Интернета (CSW), 126  
CDN; См. Сети доставки контента  
Центр интеллектуальных пространственных вычислений по науке о воде / энергии, 241  
CentOS, 195  
Центральный процессор (ЦП), 8, 33, 243  
Характеристики; См. Облачные вычисления  
архитектура, концепции и характеристики  
Выбор облачных сервисов (в сторону модели затрат на облачные вычисления), 93–108.  
особенности и требования приложений, 97–98  
больших данных, обработка, 98  
приложений без полей, 98  
интенсивных коммуникаций приложений, 98  
интенсивных вычислений приложений, 98  
одновременных интенсивных приложения, 98  
приложения с интенсивным использованием данных, 98  
правил ценообразования для облачных платформ, 96–97  
сети доставки контента, 97  
затраты на хранение данных / базу данных, 97  
стоимость передачи данных, 97  
стоимость использования VM, 96  
Плата за адрес интернет-протокола, 97  
режим оплаты по мере использования, 96  
зарезервированный режим, 96  
предоставление мощности облачной службы и измерения, 94–96  
вычислительные возможности, 94  
степень настройки и гибкость / масштабируемость, 95  
геолокации облака инфраструктуры, 96  
ИТ-безопасность и конфиденциальность, 94–95  
управляемость, удобство использования и служба поддержки, 95  
надежность и надежность, 95  
Соглашения об уровне обслуживания, 95  
факторы, влияющие на облачный сервис отбор, 94–98  
особенности приложения и требования, 97–98  
правила ценообразования облачной платформы, 96–97  
емкость облачной службы обеспечения и измерения, 94–96  
важность и проблемы выбор облачных сервисов, 93–94  
особенности приложения, 93  
перспектива облачных сервисов, 94  
экономическая перспектива, 94  
виртуальная машина конфигурация, 93  
подробные соображения, 104–106  
метрики облачных сервисов, 104  
вычислительная мощность провизия, 105  
правильность и точность оценочные модели, 105–106  
расчетная модель, 105  
интерактивность и визуализация функции консультативного инструмента, 106  
системы мониторинга, 104  
рентабельность инвестиций, 105  
удовлетворение ограничений, 105  
пространственно-временные воздействия, 105  
подходящие модели оценки и основные принципы выбора, 105  
актуальная информация облачных сервисов, 106

## 314 Индекс

---

- проблемы, 107  
выбор облачных сервисов (Earth Science Information Partners), 98–104  
Описание требований к приложению, 102  
архитектура советника, 99  
облачные информационные базы данных, 99  
Индивидуальное приложение, 100, 101  
Приложение для хранения данных, 100  
общий рабочий процесс для выбора облачной службы, 99–102  
Получить решения, 102  
Печать, 103–104  
покупательские предпочтения, 101  
Простое вычислительное приложение, 100  
Простое веб-приложение, 100  
Создатель решений, 99  
Оценщик решений, 99  
Таблицы и диаграммы, 103  
вариант использования, 102–104  
Веб-интерфейс, 99  
резюме, 106–107  
Пользователь вычислений, 8, 143  
Citrix, 28  
Бесклассовая междоменная маршрутизация (CIDR), 57  
Модель клиент / сервер, 36  
Изменение климата, 5–6  
влияющие факторы, 5  
гипотетический сценарий, 6  
ударов, 6  
многомасштабное моделирование, 6  
Климат @ Home, 8, 201, 277  
Climate@Home, поддержка облаков, 143–160  
архивирование наборов данных, 156  
фон, 143–145  
BOINC сервер, 144, 145  
создание масштабируемого и эластичного хранилища данных, 146  
проблемы, 146–147  
вычислительные ресурсы гражданина, 143  
моделирование климата, 143  
Dgural, 144  
обработка распределенных данных, 146  
Институт космических исследований имени Годдарда НАСА ModelE, 143  
обеспечение надежной и устойчивой киберинфраструктуры централизованного управления, 146  
пространственный веб-портал, 144  
Виртуальный образ, 145  
ПО для виртуализации, 145  
большие данные, масштабируемость управления, 156  
BOINC-сервер, развертывание, 149–151  
разрешить доступ к сети, 149  
настроить Elastic Block Storage, 150  
настроить баланс нагрузки или масштабируемость, 151  
создать проект BOINC и настроить BOINC сервер, 150  
создать рабочие единицы и запустить проект, 151  
установить программное обеспечение сервера BOINC, 150  
установить программные пакеты, 150  
запустить экземпляр, 149  
загрузите ModelE VirtualBox image и файлы конфигурации ModelE для экземпляра, 150  
climatethome\_re.xml, 159  
облачное хранилище, 156  
развертывание и оптимизация, 147–153  
Образ машины Amazon, 148  
Выбор AMI для BOINC-сервера, 152  
резервное копирование данных и виртуальных машин, 151–152  
развертывание сервера BOINC, 149–151  
развертывание и настройка нескольких серверов, 152  
развертывание нескольких компонентов, 152  
развертывание пространственного веб-портала, 147–149  
отличие от общих шагов в главе 5, 152–153  
общий рабочий процесс развертывания, 147–151  
обработка вычислений и интенсивность сети, 153  
балансировка нагрузки и автоматическое масштабирование, 152  
База данных MySQL, 148

- конфигурация портала, 151  
особые соображения, 151–152  
    E4M20a\_000040\_wu.xml, 158–159  
проблемы, 158  
пространственный веб-портал,  
развертывание,  
147–149  
разрешить доступ к сети, 147  
настроить эластичный блок  
Хранение, 148  
настроить баланс нагрузки или  
масштабируемость, 149  
создать АМІ из запущенного  
например, 149  
установить программные пакеты, 148  
запустить экземпляр, 148  
восстановить базу данных, 148  
запустите сервер Apache2 и Tomcat и  
настройте веб-портал, 149  
передать исходные коды и базу  
данных системы Climate @ Home, 148  
резюме, 157–158  
системные демонстрации, 153–155  
геовизуальный аналитический  
портлет, 154  
обзор пространственного веб-портала,  
153–154  
портлет управления ресурсами, 155  
использование вычислительной  
техники, вып., 157  
Climate @ Home, тест с  
использованием данных и  
вычислений, 207–211  
очистка и выпуск экземпляра, 211  
Требования к Climate @ Home  
computing, 207–208  
сбор и анализ файлов журналов, 211  
настройка вычислительной среды,  
208–209  
настройка среды тестирования, 209–  
210  
Приложение на базе Linux, 208  
Модель E, 207  
тестовый дизайн, 208  
тестирование нескольких запусков с  
помощью сценария оболочки Linux на  
всех машинах, 210  
анализы результатов, 211 рабочий  
процесс тестирования, 208–211  
экземпляры виртуальных машин, 208  
CloudApp, 299  
Архитектура, концепции и  
характеристики облачных вычислений,  
19–31  
характеристики, 22–23  
широкополосная сеть и независимый  
доступ от устройств, 22  
размеренный сервис, 23  
по запросу, 22  
быстрая эластичность, 23  
надежность, 23  
объединение ресурсов, 22–23  
самообслуживание с минимальным  
взаимодействием с провайдерами, 22  
виртуализация, общий ресурс  
модель на базе, 23  
архитектура облачных вычислений, 20–  
21  
бизнес-модели, 20  
IaaS, 21  
Общая архитектура NIST, 20  
PaaS, 21  
игроков в облачных вычислениях, 20  
SaaS, 21  
коммерческие облака, 27–29  
управление взаимоотношениями с  
клиентами, 27–28  
Межсетевой экран, 28  
GoGrid, 28–29  
Google App Engine, 27  
IBM SmartCloud, 29 лет  
Joyent, 28 лет  
Стойка, 28  
Платформа Salesforce, 27  
SmartOS, 28  
Verizon SaaS, 29 лет  
виртуализация, технология Xen для, 28,  
29  
виртуальная частная сеть, 28  
концепты, 19–20  
фон, 19  
все как услуга, 19  
сеточные вычисления, 19  
неоднородность облачных сервисов, 19  
модели развертывания и типы облаков,  
25–27  
коммерческое облако, 25  
общественные облака, 26  
гибридные облака, 27  
частные облака, 26  
публичные облака, 25

## 316 Индекс

- облачные решения с открытым исходным кодом, 29–30  
Amazon ElasticMapReduce, 29 лет  
Apache VCL, 30  
Файловая система Google, 30  
Файловая система Tianwang, 30  
TPlatform, 30  
Xen Cloud Platform, 29–30  
проблемы, 30  
обзор ресурсов облачных вычислений, 27–30  
коммерческие облака, 27–29  
облачные решения с открытым исходным кодом, 29–30  
служебные модели, 23–25  
приложения, 24  
проблема больших данных, 24  
Сервисный режим DaaS, 24  
все как услуга, 24  
Gmail, 24  
Google Планета Земля, 24  
Сервисный режим IaaS, 24  
Сервисный режим PaaS, 24  
Сервисный режим SaaS, 24  
резюме, 30  
Исследования облачных вычислений для наук о Земле и приложений, 295–309  
развитие видения 21 века для приложений геонаук, 295–297  
Application Vision, 296–297  
запросы фундаментальной геопространственной науки, 295–296  
симуляторы глобальных военных конфликтов, 296  
Google Планета Земля, 296  
Gore Digital Earth Vision, 296 г.  
интеграция наук о Земле с другими областями науки для новых открытий, 296  
Виртуальная Земля/Карты Bing, 296  
Virtual Geographic  
Окружающая среда, 296  
проблемы, 306  
резюме, 305–306  
синергетический прогресс социальных наук и облачных вычислений, 303–305  
управление облаком, 303  
облачный охват, 303  
данные, контролируемые экспортом, 304  
Федеральная программа управления рисками и авторизацией, 304  
глобальное сотрудничество, 305  
Закон Сарбейнса-Оксли, 304  
безопасность и правила, 304  
визуальная аналитика, 305  
технологические достижения, 297–303  
оценка и отбор облаков, 297–298  
критерии измерения облачных сервисов, 297  
управление ресурсами облачной службы, 298–299  
облачные сервисы хранения, 299  
программирование в ограничениях, 299  
Общественные облака, 299  
резервное копирование и синхронизация данных, 299  
Целевая группа по распределенному управлению, 300  
Глобальный вид поставщика, 297  
совместимость, 299–301  
новая визуализация и интерактивные системы, 301  
моделирование и доступ в реальном времени, 302–303  
надежность и доступность, 302  
Возврат инвестиций, 297  
узкое место масштабируемости, 299  
недоступность услуги, 302  
сторонний аудит, 297  
Общая стоимость владения, 297  
Стоимость использования, 297  
Windows Server  
Виртуализация, 300  
Облачный диск, 299  
ГИС с поддержкой облака, 111  
CloudFront, 182, 183  
Облачные сервисы, 179–199  
Amazon Web Services, 181–187  
Amazon CloudFront, 183  
прикладной уровень, 182  
архитектура, 181–182  
совместимость, 183  
стоимость, 184–185  
межсервисный уровень, 182  
развертывание и интерфейс, 183  
Эластичный Beanstalk, 183  
Неудачи, 187  
обратная связь от сообщества, 186–187

- общие характеристики EC2, 182–185
- гипервизоры, 183
- слой строительных блоков
- инфраструктуры, 182
- список основных заказчиков, 185–186
- основные пользователи и общие комментарии, 185–187
- созревание, 186
- Модель экземпляров по требованию, 184
- Поддерживает ОС, 184
- уровень физической инфраструктуры, 181–182
- слой строительных блоков платформы, 182
- надежность, 183
- Модель зарезервированных экземпляров, масштабируемость 184, 182–183
- Модель спотовых экземпляров, 184
- уровень инструментов и API, 182
- сложность использования, 187
- автоматическое масштабирование и балансировка нагрузки, 180
- Azure, 188–193
- архитектура, 188–189
- возможности, 189
- совместимость, 190
- стоимость, 191
- развертывание и интерфейс, 191
- обратная связь от сообщества, 193
- Плоское сетевое хранилище, 189
- общие характеристики, 189–191
- высокопроизводительные вычисления, 190
- гипервизоры, 191
- список основных заказчиков, 192
- основные пользователи и общие комментарии, 192–193
- созревание, 193
- Поддерживает ОС, 191
- надежность, 191
- масштабируемость, 189–190
- система самообслуживания, 192
- сложность использования, 193
- сравнение соображений готовности для трех платформ, 197
- географическое присутствие, 179
- конкурентное преимущество, 179
- выбор поставщиков услуг, 179
- услуги смены, 179
- виртуализация, 179
- Nebula, 194–196
- архитектура, 194
- CentOS, 195
- совместимость, 195
- компоненты, 194
- стоимость, 195
- развертывание и интерфейс, 195
- обратная связь от сообщества, 196
- общие характеристики, 195
- гипервизоры, 195
- Дистрибутивы Linux, 195
- список основных заказчиков, 196
- основные пользователи и общие комментарии, 196
- созревание, 196
- НАСА, 194
- Поддерживает ОС, 195
- надежность, 195
- масштабируемость, 195
- Ubuntu, 195
- сложность использования, 196
- проблемы, 198
- соглашение об уровне обслуживания, 179, 180
- оборотность, 180
- типовой SLA, 180
- резюме, 197–198
- пользовательские интерфейсы и доступ к серверов, 180
- CloudSleuth, 297
- CloudStack, 224, 225–228
- архитектура, 225–226
- облачная модель, 227
- совместимость, 227
- стоимость, 227
- развертывание и интерфейс 227
- отзыв сообщества, 228
- общие характеристики, 226–227
- гипервизоры, 227
- монтажные сложности, 228
- список клиентов, 227
- основные пользователи и общие комментарии, 227–228
- зрелость, 228
- Поддержка ОС, 227
- надежность, 227
- масштабируемость, 226
- Кластерный экземпляр, 85
- Коммерческое облако, 25
- Приложение общего рабочего изображения (COP), 112, 120
- Приложения с интенсивным общением, 98

- Облака сообщества, 26  
Обеспечение вычислительной мощности, 105  
Модель вычислительной архитектуры, 36  
Вычислительная нагрузка, 98, 280–284  
облачные CPU вычисления, 281  
облачные вычисления на GPU, 282  
стоимость, 283  
интеллектуальный анализ данных для информации / знаний, 280  
Цифровая модель местности, 280  
пример с интерполяцией цифровой модели оценки, 280–281  
совместимый и умный каркас, 283  
сеть, 283  
извлечение параметров, 280  
моделирование явлений, 280  
Остающиеся проблемы и будущие исследования, 283–284  
решения, 281–282  
Вычислительные технологии, 35–36  
модель клиент / сервер, 36  
модель вычислительной архитектуры, 36  
противоречия, 35  
парадигма распределенных вычислений, 35  
Концепции; См. Архитектура, концепции и характеристики облачных вычислений.  
Параллельная интенсивность, 98, 284–289.  
синхронизация данных и производительность, 288  
эластичность, 287  
пример с Информационным центром GEOSS, 285  
глобальная доставка контента, 286  
Кэширование памяти, 288–289  
Остающиеся проблемы и будущие исследования, 288–289  
раствор, 286–288  
пространственно-временная индексация, 287–288  
используя пространственно-временные принципы, 288  
Команды кондора, 87  
Программирование в ограничениях, 299  
Сети доставки контента (CDN), 97  
Контент-сервисы, 113  
Заявление КС; См. Стандартное приложение для рабочего изображения  
Модель расчета стоимости, 105  
Модель стоимости; См. Раздел Выбор облачных сервисов (в сторону модели затрат на облачные вычисления)  
ЦПУ; См. Центральный процессор  
CRM; См. Управление взаимоотношениями с клиентами  
CSW; См. Службу каталогов для Интернета  
Платформа CUDA (NVIDIA), 278  
Кучевые облака, 299  
Управление взаимоотношениями с клиентами (CRM), 27–28  
D  
Сервисный режим DaaS, 24  
Веб-приложения, управляемые базами данных, 78–84  
разрешить доступ к сети, 79  
создать новый AMI из запущенного экземпляра, 84  
настройка, 81  
развернуть приложение, 81  
Веб-сайт на основе Drupal, 81, 84  
DynamoDB, 78  
поле googlemap, 83  
запустить экземпляр, 79  
войти в экземпляр, 79  
Консоль MySQL, 80  
Служба реляционной базы данных, 78  
Secure Shell, 79  
настроить среды, 79  
SimpleDB, 78  
передать файлы на экземпляр, 80–81  
Системы управления базами данных (СУБД), 76  
Приложения с интенсивным использованием данных, 98  
DataNodes, 42  
СУБД; См. Раздел Системы управления базами данных  
Dell, 28  
DFS; Раздел Распределенная файловая система  
Цифровая модель рельефа (DEM), 77, 280  
Цифровая модель местности (DTM), 280  
Распределенные вычисления, 10–11, 35  
Распределенная файловая система (DFS), 39–42, 277  
Распределенная файловая система Apache Hadoop, 41–42  
технологии хранения больших данных, 41  
большие куски, 41  
облачное хранилище, 39  
DataNodes, 42  
Файловая система Google, 40–41

- введение, 40  
множественная репликация данных, 41  
не локальное кеширование, 41  
Целевая группа по распределенному управлению (DMTF), 300  
DMTF; См. Раздел Dtopbox Целевой группы распределенного управления, 39, 299.  
Веб-сайт на основе Drupal, 81, 84  
DTM; См. Цифровая модель местности Прогнозирование пыльных бурь, поддержка облаков, 161–176.  
фон, 161  
приложений, 161  
глобальное изменение климата, 161  
мотивация, 161  
негативные воздействия пыльных бурь, 161  
проблемы, 162–164  
разрушительные события, 163  
прогноз пыльной бури, 163  
высокопроизводительные вычисления, 162  
использование облачных вычислений, 163  
Негидростатическая мезомасштабная модель - модель пыли, 162  
имитация пыльной бури явления, 162  
Подключение к глобальной сети, 164  
демонстрация, 169–174  
рентабельность, 173–174  
хабуб, 169  
спектакль, 170–173  
Пыльная буря в Фениксе, 169  
результат моделирования, 170  
трехчасовое прогнозирование, 172  
развертывание и оптимизация, 164–169  
список авторизованных ключей, 166  
разрешить доступ к сети, 164  
автомасштабирование, 169  
настроить головной узел и вычислительный узел, 165–166  
настроить и протестировать модель, 167  
настроить виртуальную кластерную среду, 167–168  
создать АМІ и вычислительный узел, 165  
создать АМІ из запущенного экземпляра, 167  
создать том EBS и прикрепить его к головному узлу, 166  
развернуть модель в каталоге экспорта NFS, 166  
экспортировать каталог NFS на вычислительный узел, 166  
общий рабочий процесс, 164–167  
интенсивность общения, 169  
вычислительная нагрузка на обработку, 169  
установить программные зависимости, 165  
Ошибка «Недостаточная емкость», 168  
запустить экземпляр, 164  
слабосвязанное вложенное моделирование и облачные вычисления, 168–169  
Интерфейс передачи сообщений, 165  
смонтировать том в каталог экспорта NFS, 166  
Сетевая файловая система, 165  
особые соображения, 167–169  
SSH к экземпляру, 165  
краткое изложение отличий от общих шагов в главе 5, 169  
Параллельная виртуальная файловая система версии 2, 174  
проблемы, 175  
резюме, 175  
Прогнозирование пыльных бурь, проверка готовности облаков на 255–256 гг.  
сравнение облачного промежуточного программного обеспечения, 256  
вычислительные требования для прогнозирования пыльной бури, 255  
НПС против облаков, 255, 256  
сравнение решений с открытым исходным кодом, 255  
тестовый дизайн, 255  
анализы результатов тестов, 256  
тестовый рабочий процесс, 256  
технология виртуализации, 255, 256  
Прогнозирование пыльной бури, тестирование облачности, 211–216  
добавление дополнительных экземпляров, 214, 215  
изображения модели окружающей среды здания пыльной бури, 213  
проверка результатов каждой платформы и анализ результатов, 214  
настройка облачной платформы, 214  
разное количество вычислительных узлов от разных сервисов, 212  
требования к вычислениям для прогнозирования пыльной бури, 211–212

## 320 Индекс

---

- влияние гиперпоточности, 212
- экземпляры рециклинга, 215
- запуск сценария тестирования на каждой платформе, 214
- одинаковое количество вычислительных ресурсов от разных сервисов, 212
- стартовало такое же количество экземпляров, 214
- дизайн теста, 212–213
- анализы результатов анализов, 215–216
- рабочий процесс тестирования, 213–215
- передача сценария тестирования в головной узел, 214
- написание сценария тестовой оболочки, 214
- DynamoDB, 78
- Е
- Наблюдения Земли (EO), 76, 125
- Информационные партнеры по наукам о Земле (ESIP), 98–104
- Описание требований к приложению, 102
- архитектура советника, 99
- Консультации по внедрению облака, 98–104
- облачные информационные базы данных, 99
- Индивидуальное приложение, 100, 101
- Приложение для хранения данных, 100
- общий рабочий процесс для выбора облачной службы, 99–102
- Получить решения, 102
- Печать, 103–104
- покупательские предпочтения, 101
- Простое вычислительное приложение, 100
- Простое веб-приложение, 100
- Создатель решений, 99
- Оценщик решений, 99
- Таблицы и диаграммы, 103
- вариант использования, 102–104
- Веб-интерфейс, 99
- ОВОС; См. Управление энергетической информации США.
- Эластичный Beanstalk, 183
- Магазин эластичных блоков (EBS), 53, 127, 148
- EMR; См. Amazon Elastic MapReduce
- Enabling technologies, 33–48 computing technologies, 35–36
- модель клиент / сервер, 36
- модель вычислительной архитектуры, 36
- противоречия, 35
- парадигма распределенных вычислений, 35
- распределенная файловая система, 39–42
- Распределенная файловая система Apache Hadoop, 41–42
- технологии хранения больших данных, 41
- большие куски, 41
- облачное хранилище, 39
- DataNodes, 42
- Файловая система Google, 40–41
- введение, 40
- множественная репликация данных, 41
- не локальное кеширование, 41
- аппаратные достижения, 33–35
- жесткие диски, 34
- высокопроизводительные вычисления, 33
- Модель «Инфраструктура как услуга», 34
- многоядерные и многоядерные технологии, 33–34
- многоядерные технологии, 33
- нетворкинг, 34
- умные устройства, 35
- хранение, 34–35
- ленточные системы, 34
- проблемы, 47
- сервис-ориентированная архитектура, 44–45
- пособие, 44
- повторное использование компонентов, 44
- существующая системная интеграция, 44
- независимость от языка и платформы, 45
- резюме, 46
- виртуализация, 36–39
- полная виртуализация, 37
- аппаратная виртуализация, 37
- реализация, 37
- Виртуальная машина на основе ядра, 38
- паравиртуализация, 37
- консолидация серверов, 36
- SmartOS, 39
- решения, 38–39
- VirtualBox, 38
- VMWare, 39
- Зен, 38
- Web X.0, 42–45
- Расширяемый язык разметки, 43
- протокол передачи гипертекста, 42
- Intelligence Web, 43
- Вычисления "сеть как платформа", 42

- Сервис-Ориентированная Архитектура, 44–45
- услуги через Интернет, 44
- Протокол простого доступа к объектам, 43
- Веб 1.0, 42
- Веб 2.0, 42
- Веб 3.0, 42
- Веб 4.0, 43
- Веб-сервисы, 43–44
- Язык описания веб-сервисов, 43
- ЕО; См. Наблюдение за Землей
- ESIP; См. Информацию о науках о Земле
- Партнеры
- Эвкалипт, 224, 228–230
- архитектура, 228–229
- облачная модель, 229
- совместимость, 230
- стоимость, 230
- развертывание и интерфейс, 230
- обратная связь от сообщества, 230
- общие характеристики, 229–230
- гипервизоры, 230
- монтажные сложности, 230
- список клиентов, 230
- основные пользователи и общие комментарии, 230
- зрелость, 230
- Поддержка ОС, 230
- надежность, 230
- масштабируемость, 229
- Все как услуга (XaaS), 19, 24, 44
- Расширяемый язык разметки (XML), 43, 45
- F
- Facebook, 28, 153
- Федеральный комитет географических данных (FGDC), 261
- Федеральный закон об управлении информационной безопасностью (FISMA), 20, 52, 94–95, 268
- Федеральная программа управления рисками и авторизацией (FedRAMP), 304
- Конфигурация межсетевого экрана, 63
- по умолчанию, 63
- сервисная, 28
- Плоское сетевое хранилище (FNS), 189–190
- Полная виртуализация, 37
- грамм
- GAE; См. Раздел Администрирование общих служб Google App Engine (GSA), 261.
- Инициатива GeoCloud, 261–272
- обсуждение, 270–271
- аварийное переключение, резервирование, 271
- процесс утверждения безопасности, 270
- развертывание программного обеспечения, 270
- время развертывания, 270
- Федеральный комитет географических данных, 261
- Администрация общих служб, 261
- Действия GeoCloud, 264–268
- Amazon Elastic Cloud Computing, 265
- Интерфейс прикладного программирования, 266
- приложений, 266
- базовые платформы, 266
- создать прототип платформы, 265–267
- документ и обнаружение, 267–268
- Пользовательский интерфейс, 266
- Подтвердить с заявками агентства, 267
- Архитектура GeoCloud, 262–264
- серверы приложений, 263
- фреймворки и библиотеки, 264
- Информационная служба ГЕОСС, 264
- Инструмент LarvaMap, 264
- Национальная инвентаризация водно-болотных угодий, 264
- Уровни PaaS, 263–264
- активаторы платформы, 263–264
- время выполнения, 264
- Топологически интегрированное географическое кодирование и привязка, 264
- Безопасность GeoCloud, 268–269
- Безопасность Amazon Web Services, 268
- Полномочия на эксплуатацию, 268
- Операция безопасности GeoCloud, 269
- экономия на инфраструктуре, 262
- эксплуатационные расходы в облаке, 269–270
- экономия платформы, 262

## 322 Индекс

- проблемы, 271  
сводка, 271  
Геокодирование, 113  
Географическая информационная система (ГИС), 111; Также ArcGIS в  
облачное приложение Geoscience бросает  
вызов вычислительной инфраструктуре,  
3–17  
преимущества и недостатки,  
14–15  
преимущества облачных вычислений, 14  
большие данные, передача, 15  
проблемы, 14–15  
окончательная потеря контроля над  
вычислительными ресурсами, 15  
рождение облачных вычислений, 10–13  
интерфейс прикладного  
программирования, 13  
совместное использование вычислений и  
экономия средств, 12  
распределенные вычисления, 10–11  
эластичность услуг, 13  
появление облачных вычислений, 12–13  
аппаратная виртуализация, 11  
аппаратная виртуализация  
промежуточного программного  
обеспечения, 12  
услуги по запросу, 11–12  
надежность, 12  
рентабельность инвестиций, 11  
Соглашения об уровне обслуживания, 12  
проблемы и возможности, 3–7  
большие данные, 4  
изменение климата, 5–6  
аварийное реагирование, 4–5  
энергия, 3–4  
глобальный энергетический рынок,  
прозрачность, 4  
опасные материалы, 4  
Ураган Катрина, 4  
устойчивое развитие, 6–7  
изменение климата, 5–6  
влияющие факторы, 5  
гипотетический сценарий, 6  
ударов, 6  
многомасштабное моделирование, 6  
потребности новой вычислительной  
инфраструктуры, 8–10  
центральный процессор, 8  
гражданские вычисления, 8  
графический процессор, 8  
высокопроизводительные вычисления, 8  
улучшение доступности, 10  
обеспечение достаточной  
вычислительной мощности, 8–9  
отвечает в режиме реального времени, 9  
экономия бюджета, 10  
экономия энергии, 9–10  
всплеск вычислительных потребностей, 8  
резкий рост доступа к вычислительным  
ресурсам, 9  
служебные вычисления, 10  
проблемы, 15  
Соглашения об уровне обслуживания,  
услуги по запросу, 12  
резюме, 15  
устойчивое развитие, 6–7  
большие данные, доступность, 7  
задачи, 7  
компромиссы между конкурирующими  
целями, 7  
Приложения для геофизических  
исследований, облачность, 75–91  
облачные приложения для геолого-  
геофизических исследований, 77–78  
развертывание приложения, 78  
настройка среды, 77  
общие компоненты для приложений  
геонаук, 75–77  
системы управления базами данных, 76  
Цифровая интерполяция модели  
рельефа, 77  
Данные наблюдения Земли, 76  
высокопроизводительные вычисления, 76–  
77  
серверное программирование, 75–76  
пространственные базы данных, 76  
веб-приложения, управляемые базами  
данных, 78–84  
разрешить доступ к сети, 79  
создать новый API из запущенного  
экземпляра, 84  
настройка, 81  
развернуть приложение, 81  
Веб-сайт на основе Drupal, 81, 84  
DynamoDB, 78  
поле googlemap, 83  
запустить инстанс, 79  
войти в экземпляр, 79  
Консоль MySQL, 80  
Служба реляционной базы данных, 78  
Secure Shell, 79  
настроить среды, 79

- SimpleDB, 78
- передать файлы на экземпляр, 80–81
- Приложения HPC, 84–90
- разрешить доступ к сети, 84–85
- экземпляр кластера, 85
- Команды кондора, 87
- настроить промежуточное ПО на всех узлах для обеспечения связи, 88
- создать новый AMI из работающего головного узла, 88
- установить пакеты промежуточного программного обеспечения, 85
- Комплект разработчика Java, 87
- запустить инстанс, 85
- запускать другие экземпляры из нового AMI, 88
- передать данные матрицы высот и код интерполяции, 89
- проблемы, 90
- сводка, 90
- варианты использования, 78–90
- веб-приложения, управляемые базами данных, 78–84
- типичные приложения HPC, 84–90
- Геопоиск, 113
- Информационная служба GEOCC, 201, 264, 285
- Информационная служба GEOSS, облачная поддержка, 125–142
- фон, 125–126
- большие данные, 126
- Служба каталогов для Интернета, 126
- проблемы, 126–127
- одновременный доступ, 127
- Системы наблюдения Земли, 125
- Магазин эластичных блоков, 127
- Области общественной выгоды, 125
- пространственно-временной поиск и полнотекстовый поиск, 126
- Геологическая служба США, 125
- развертывание и оптимизация, 127–135
- разрешить доступ к сети, 127
- автоматическое масштабирование, 132–134
- Параметр AutoScaling MaxSize, 134
- настроить баланс нагрузки или масштабруемость, 130
- настроить сервлет для CLN, 129
- создать AMI из запущенного экземпляра, 130
- резервное копирование данных, 130–131
- отличие от общих шагов в главе 5, 134–135
- рабочий процесс общего развертывания, 127–130
- установочный пакет, 128
- запустить экземпляр, 127
- балансировка нагрузки, 132
- триггер ресурса, 133
- восстановить базу, 129
- особые соображения, 130–134
- запустить службу и настроить приложение, 130
- Tomcat, 128
- передать код и данные CLN, 129
- экономические преимущества, 137–138
- проблемы, 138
- резюме, 138
- демонстрация системы, 135–136
- местный поиск, 135–136
- удаленный поиск, 136
- технические преимущества, 138
- шаблон для создания функции автомасштабирования, 138–141
- Информационная служба GEOSS, тест готовности к облаку для, 254–255
- Требования к вычислительным средствам Информационного центра, 254
- дизайн тестирования, рабочий процесс и анализ, 254–255
- Информационный центр GEOCC, одновременное использование теста интенсивности, 202–207
- анализировать результаты JMeter, 205
- требования центра обмена информацией для компьютерных услуг, 202–203
- Графический интерфейс пользователя, 205
- установка CLN на разные облачные сервисы, 204
- тесты балансировки нагрузки и масштабируемости, 204
- тесты балансировки нагрузки и автомасштабирования, 206
- матрица теста одновременных поисковых запросов, 203
- матричный тест, 206
- работает JMeter, 205
- настройка запроса CSW GetRecords, 204
- настройка плана тестирования JMeter, 205
- тестовый дизайн, 203–204

## 324 Индекс

- анализы результатов анализов, 206–207  
рабочий процесс тестирования, 204–206  
Информационная служба ГЕОСС,  
Архитектура GeoCloud, 264  
GFS; См. Файловую систему Google  
ГИС; См. Географическая информационная  
система  
Программное обеспечение ГИС как услуга,  
115–116  
Глобальное сотрудничество, 305  
Симуляторы глобального военного  
конфликта, 296  
Глобальный вид поставщика, 297  
Gmail, 24  
GoGrid, 28–29  
Google App Engine (GAE), 27  
Google Диск, 39  
Драйвер Google, 299  
Google Планета Земля, 24, 154, 296  
Файловая система Google (GFS), 30, 40–41  
Google Карты, 154  
Gore Digital Earth Vision, 296 г.  
Графический интерфейс пользователя (GUI),  
52, 205  
Блок обработки графики (GPU),  
8, 33, 278  
Грид-вычисления, 19  
GSA; См. Управление общих служб (GSA), 261  
GUI См. Графический интерфейс  
пользователя  
ЧАС  
Распределенная файловая система Hadoop  
(HDFS), 41–42, 278  
Жесткие диски, 34  
Аппаратные достижения, 33–35  
жесткие диски, 34  
высокопроизводительные вычисления, 33  
Модель «Инфраструктура как услуга», 34  
многоядерные и многоядерные технологии,  
33–34  
многоядерные технологии, 33  
нетворкинг, 34  
умные устройства, 35  
хранение, 34–35  
ленточные системы, 34  
Аппаратная виртуализация, 11, 12, 37  
Аппаратная виртуальная машина (HVM), 85  
HDFS; См. Приложение Hello Cloud для  
распределенной файловой системы Hadoop,  
63  
Hewlett-Packard (HP), 26  
Высокопроизводительные вычисления  
(HPC), 8, 19, 212, 233  
Высокопроизводительные вычисления  
приложения, 84–90  
разрешить доступ к сети, 84–85  
экземпляр кластера, 85  
Команды кондора, 87  
настроить промежуточное ПО  
на всех узлах, чтобы включить  
связь, 88  
создать новый АМІ из запущенного  
головной узел, 88  
установить пакеты промежуточного программного  
обеспечения, 85  
Комплект разработчика Java, 87  
запустить экземпляр, 85  
запускать другие экземпляры из нового  
АМІ, 88  
передать данные матрицы высот и  
код интерполяции, 89  
Хостинговые услуги, 113  
HP; См. Hewlett-Packard  
HPC; См. Высокая производительность  
вычисление  
Папка HTML, 65  
HTTP; См. Протокол передачи гипертекста  
HVM; См. Аппаратную виртуальную машину  
Гибридные облака, 27  
Протокол передачи гипертекста (HTTP),  
42, 57  
Технология Hyperthreading, 212  
Гипервизоры, 227, 230, 233  
я  
IaaS; См. Инфраструктуру как услугу  
IBM SmartCloud, 29 лет  
iCloud, 39 лет  
МЭА; См. Международное энергетическое  
агентство  
IS; См. Информационные службы Интернета  
Имидж сервисы, 113  
Инфраструктура как услуга (IaaS), 21  
облачные сервисы, 179  
модель, 34  
сервисный режим, 24  
Intelligence Web, 43  
Интенсивность данных, вычислений,  
одновременный доступ и  
пространственно-временные модели  
(хендлинг), 275–293  
большие данные, 276–280  
пример с Climate @ Home,  
277  
разрыв между MapReduce и  
научные данные, 279–280

- Распределенная визуализация на базе GPU в облачных сервисах, 278
- Распределенная файловая система Nadoop, 278
- Введение, 276–277
- управление вызовом, 276
- метаданные, пространственно-временной индекс и распределенный файл
- система, 277
- доступ к большим данным по запросу, 279
- параллельные вычисления с облачными сервисами, 278
- пиксельная визуализация, 276–277
- вызов обработки, 276
- Остающиеся проблемы и будущие исследования, 278–280
- решения, 277–278
- вычислительные мощности, 279
- конечная цель, 276
- задача визуализации, 276
- вычислительная мощность, 280–284
- облачные CPU вычисления, 281
- облачные вычисления на GPU, 282
- стоимость, 283
- интеллектуальный анализ данных для информации / знаний, 280
- Цифровая модель местности, 280
- пример с интерполяцией цифровой модели оценки, 280–281
- интероперабельная и интеллектуальная структура, 283
- сеть, 283
- извлечение параметров, 280
- моделирование явлений, 280
- Остающиеся проблемы и будущие исследования, 283–284
- решения, 281–282
- одновременная интенсивность, 284–289
- синхронизация данных и производительность, 288
- эластичность, 287
- пример с GEOSS
- Информационная служба, 285
- глобальная доставка контента, 286
- Кэширование памяти, 288–289
- Остающиеся проблемы и будущие исследования, 288–289
- раствор, 286–288
- пространственно-временная индексация, 287–288
- используя пространственно-временные принципы, 288
- человеческие знания, 275
- интенсивности, 275
- проблемы, 290–291
- пространственно-временная интенсивность, 289–290
- сбор данных, 289
- система аварийного реагирования, 289
- природные явления и научные исследования, 289
- пространственно-временные данные, 289
- сводка, 290
- Международное энергетическое агентство (МЭА), 3
- Информационные службы Интернета (IIS), 53, 54, 68
- Плата за IP-адрес, 97
- Я
- Ява, 266
- Комплект разработчика Java (JDK), 87
- Среда выполнения Java (JRE), 251
- JMeter, 205
- Джойнт, 28 лет
- К
- Виртуальная машина на основе ядра (KVM), 38, 195, 243
- Л
- Инструмент LarvaMap, 264
- Законодательство
- Федеральный закон об управлении информационной безопасностью, 20, 52, 94–95, 268
- Закон Сарбейнса-Оксли, 304
- Модель климата на базе Linux, 145
- распределения, Nebula, 195
- сценарий оболочки, 210
- VM, 53
- Пакет LLCbench, 244
- Локальные сети (ЛВС), 243
- М
- Картографические сервисы, 113
- Матричный тест, 206
- Тестирование памяти, 247
- Кэширование памяти, 288–289

## 326 Индекс

- Интерфейс передачи сообщений (MPI), 165  
Микеланджело, 192  
Microsoft; См. Также Карты Windows  
Azur Bing, 154  
SQL Server, 76, 190  
WebMatrix, 190  
Windows Server, 114  
Симуляторы военных конфликтов (глобальный), 296  
MPI; См. Интерфейс передачи сообщений MySQL, 79, 266  
консоль, 80  
база данных, 148  
N  
НАСА, 28  
Годдард Институт космических исследований ModelE, 143  
Туманность, 194  
Мировой ветер, 154  
Национальный институт стандартов и технологий (NIST), 20, 241, 297  
Национальная инвентаризация водно-болотных угодий, 264  
Туманность, 194–196  
архитектура, 194  
CentOS, 195  
совместимость, 195  
компоненты, 194  
стоимость, 195  
развертывание и интерфейс, 195  
обратная связь от сообщества, 196  
общие характеристики, 195  
гипервизоры, 195  
Дистрибутивы Linux, 195  
список основных заказчиков, 196  
основные пользователи и общие комментарии, 196  
созревание, 196  
НАСА, 194  
Поддерживает ОС, 195  
надежность, 195  
масштабируемость, 195  
Ubuntu, 195  
сложность использования, 196  
Данные NetCDF, 154  
Сетевая файловая система (NFS), 165  
Вычисления «Сеть как платформа», 42  
NFS; См. Network File System Nimbus, 225, 234–236.  
архитектура, 234–235  
облачная модель, 235  
совместимость, 235  
стоимость, 235  
клиентов, 235  
развертывание и интерфейс, 235  
обратная связь от сообщества, 236  
общие характеристики, 235  
гипервизоры, 235  
сложность монтажа, 236  
основные пользователи и общие комментарии, 235–236  
зрелость, 235  
Поддержка ОС, 235  
надежность, 235  
масштабируемость, 235  
NIST; См. Негидростатическая мезомасштабная модель (NMM)  
Национального института стандартов и технологий - модель пыли, 162  
ДАННЫЕ NTT, 28  
O  
Модель экземпляров по требованию, 184  
Услуги по запросу, 11–12  
OpenGeo, 266  
Служба каталогов Open Geospatial Consortium (OGC) для Интернета, 285  
OpenNebula, 225, 231–234  
архитектура, 231  
облачная модель, 231  
совместимость, 231  
стоимость, 233  
клиентов, 233  
развертывание и интерфейсы, 233  
обратная связь от сообщества, 233–234  
234  
общие характеристики, 231–233  
гипервизоры, 233  
монтажные сложности, 234  
основные пользователи и общие комментарии, 233–234  
зрелость, 233  
Поддержка ОС, 233  
надежность, 233  
масштабируемость, 231  
Решения для облачных вычислений с открытым исходным кодом, 29–30, 223–240  
Amazon Elastic MapReduce, 29 лет  
Apache VCL, 30  
CloudStack, 224, 225–228

- архитектура, 225–226
- облачная модель, 227
- совместимость, 227
- стоимость, 227
- развертывание и интерфейс 227
- отзыв сообщества, 228
- общие характеристики, 226–227
- гипервизоры, 227
- монтажные сложности, 228
- список клиентов, 227
- основные пользователи и общие комментарии, 227–228
- зрелость, 228
- Поддержка ОС, 227
- надежность, 227
- масштабируемость, 226
- Эвкалипт, 224, 228–230
- архитектура, 228–229
- облачная модель, 229
- совместимость, 230
- стоимость, 230
- развертывание и интерфейс, 230
- обратная связь от сообщества, 230
- общие характеристики, 229–230
- гипервизоры, 230
- монтажные сложности, 230
- список клиентов, 230
- основные пользователи и общие комментарии, 230
- зрелость, 230
- Поддержка ОС, 230
- надежность, 230
- масштабируемость, 229
- Файловая система Google, 30
- ключевой вызов, 223
- самые популярные доступные VIM, 223
- Нимбус, 225, 234–236
- архитектура, 234–235
- облачная модель, 235
- совместимость, 235
- стоимость, 235
- клиентов, 235
- развертывание и интерфейс, 235
- обратная связь от сообщества, 236
- общие характеристики, 235
- гипервизоры, 235
- сложность монтажа, 236
- основные пользователи и общие комментарии, 235–236
- зрелость, 235
- Поддержка ОС, 235
- надежность, 235
- масштабируемость, 235
- OpenNebula, 225, 231–234
- архитектура, 231
- облачная модель, 231
- совместимость, 231
- стоимость, 233
- клиентов, 233
- развертывание и интерфейсы, 233
- обратная связь от сообщества, 233–234
- общие характеристики, 231–233
- гипервизоры, 233
- монтажные сложности, 234
- основные пользователи и общие комментарии, 233–234
- зрелость, 233
- Поддержка ОС, 233
- надежность, 233
- масштабируемость, 231
- соображения сравнительного анализа с открытым исходным кодом, 236
- сравнения решений IaaS, 237–238
- определенные действия пользователя и интерфейсы, 239
- проблемы, 239
- резюме, 236
- Файловая система Tianwang, 30
- TPlatform, 30
- менеджер виртуальной инфраструктуры, 223
- Xen Cloud Platform, 29–30
- Кластерная файловая система Oracle Automatic Storage Management, 40
- п
- PaaS, 21
- слои, 263–264
- серверы приложений, 263
- фреймворки и библиотеки, 264
- активаторы платформы, 263–264
- время выполнения, 264
- сервисный режим, 24
- Параллельная виртуальная файловая система версии 2 (ПВФС2), 174
- Паравиртуализация, 37
- Режим оплаты по факту, 96
- КПК; См. Раздел Персональные цифровые помощники Персональные цифровые помощники (КПК), 22.
- Пиксельная визуализация, 276
- PostgreSQL, 129

- Частные облака, 26  
Проблемы ArcGIS в облаке, 123  
выбор облачных сервисов (в сторону модели затрат на облачные вычисления), 107  
Climate @ Home, поддержка облаков, 158  
архитектура облачных вычислений, концепции и характеристики, 30  
исследования облачных вычислений для науки Земли и приложений, 306  
облачные сервисы, 198  
прогнозирование пыльной бури, облачность, 175  
стимулирующие технологии, 47  
Инициатива GeoCloud, 271  
прикладные задачи геонаук для вычислительной инфраструктуры, 15  
приложения для изучения земли, облачность, 90  
Информационная служба ГЕОСС, облачность, 138  
обработка интенсивностей данных, вычислений, одновременно-временных шаблонов, 290–291  
решения для облачных вычислений с открытым исходным кодом, 239  
тестирование готовности облачных сервисов, 217  
тестирование готовности облачных вычислений с открытым исходным кодом  
решения, 257  
с использованием облачных вычислений, 72–73  
Публичные облака, 25  
Файл открытого ключа, 61  
PVFS2; См. Параллельная виртуальная файловая система версии 2  
р  
Стойка, 28  
Оперативная память (RAM), 243  
Подгруппа RATA Cloud Metrics, 297  
RDS; См. Раздел Служба реляционной базы данных Red Hat Enterprise Linux AS / ES, 114.  
Система Redhat Linux, 61  
Служба реляционной базы данных (RDS), 53, 78  
Удаленный рабочий стол, 55  
Исследования в области наук о Земле и приложений, 295–309  
развитие видения 21 века для приложений геонаук, 295–297  
Application Vision, 296–297  
запросы фундаментальной геопространственной науки, 295–296  
симуляторы глобальных военных конфликтов, 296  
Google Планета Земля, 296  
Gore Digital Earth Vision, 296 г.  
интеграция наук о Земле с другими областями науки для новых открытий, 296  
Виртуальная Земля / Карты Bing, 296  
Virtual Geographic  
Окружающая среда, 296  
проблемы, 306  
резюме, 305–306  
синергетический прогресс социальных наук и облачных вычислений, 303–305  
управление облаком, 303  
облачный охват, 303  
данные, контролируемые экспортом, 304  
Федеральная программа управления рисками и авторизацией, 304  
глобальное сотрудничество, 305  
Закон Сарбейнса-Оксли, 304  
безопасность и правила, 304  
визуальная аналитика, 305  
технологические достижения, 297–303  
оценка и отбор облаков, 297–298  
критерии измерения облачных сервисов, 297  
управление ресурсами облачной службы, 298–299  
облачные сервисы хранения, 299  
программирование в ограничениях, 299  
Кучевые облака, 299  
резервное копирование и синхронизация данных, 299  
Целевая группа по распределенному управлению, 300  
Глобальный вид поставщика, 297  
совместимость, 299–301  
новая визуализация и интерактивные системы, 301

- моделирование и доступ в реальном времени, 302–303  
надежность и доступность, 302  
Возврат инвестиций, 297  
узкое место масштабируемости, 299  
недоступность услуги, 302  
сторонний аудит, 297  
Общая стоимость владения, 297  
Стоимость использования, 297  
Windows Server  
Виртуализация, 300  
Модель зарезервированных экземпляров, 184  
Рентабельность инвестиций (ROI), 11, 105, 297  
Рейтинг, 28  
Маршрутная, 113  
S  
SaaS, 21, 24  
Платформа Salesforce, 27  
Закон Сарбейнса-Оксли (SOX), 304 SBA;  
См. Разделы "Социальные выгоды" Secure Shell, 55, 79.  
Семантический Drupal, 266  
Серверное программирование, 75–76  
Соглашение об уровне обслуживания (SLA), 23, 179  
Amazon EC2, 23, 157, 183  
выбор облачных сервисов на основе, 298  
определенная надежность, 197  
степень исполнения, 299  
описание, 180  
оборотность, 180  
типичная, 180  
Сервис-ориентированная архитектура (SOA), 44–45  
пособие, 44  
повторное использование компонентов, 44  
существующая системная интеграция, 44  
независимость от языка и платформы, 45  
СЭС; См. Простую службу электронной почты  
SETI@Home project, 8  
SimpleDB, 78  
Простая электронная почта (SES), 182  
Простой протокол доступа к объектам (МБЛО), 43  
SkyDrive, 39, 299  
SLA; См. Соглашение об уровне обслуживания SmartOS, 28, 39  
SOA; См. Сервис-ориентированная архитектура  
МБЛО; См. Простой протокол доступа к объектам. Социальные науки, синергетический прогресс облачных вычислений и 303–305  
управление облаком, 303  
облачный охват, 303  
данные, контролируемые экспортом, 304  
Федеральная программа управления рисками и авторизацией, 304  
глобальное сотрудничество, 305  
Закон Сарбейнса-Оксли, 304  
безопасность и правила, 304  
визуальная аналитика, 305  
Области социальной выгоды (SBA), 125  
SOX; См. Базы данных Sarbanes-Oxley Spatial, 76  
Развертывание пространственного веб-портала, 147–149 гг.  
разрешить доступ к сети, 147  
настроить Elastic Block Storage, 148  
настроить баланс нагрузки или масштабируемость, 149  
создать AMI из запущенного экземпляра, 149  
установить программные пакеты, 148  
запустить экземпляр, 148  
восстановить базу данных, 148  
запустите сервер Apache2 и Tomcat и настройте веб-портал, 149  
передать исходные коды и базу данных системы Climate @ Home, 148  
обзор, 153–154  
Пространственно-временная индексация, 287–288  
Пространственно-временная интенсивность, 289–290  
сбор данных, 289  
система аварийного реагирования, 289  
природные явления и научные исследования, 289  
пространственно-временные данные, 289  
Пик вычислительных потребностей, 8  
Модель спотовых экземпляров, 184  
База данных SQL Azure, 190  
SugarSync, 39, 299  
SUSE Linux Enterprise Server, 114  
Устойчивое развитие, 6–7  
большие данные, доступность, 7  
задачи, 7  
компромиссы между конкурирующими целями, 7

- Т
- Ленточные системы, 34
- ТСО; См. Общую стоимость владения
- ТСР; См. Протокол управления передачей, тестирование готовности облачных сервисов, 201–221.
- Climate@Home, тест с использованием данных и вычислений, 207–211
- очистка и выпуск экземпляра, 211
- Требования к Climate @ Home computing, 207–208
- сбор и анализ файлов журналов, 211
- настройка вычислительной среды, 208–209
- настройка среды тестирования, 209–210
- Приложение на базе Linux, 208
- Модель E, 207
- тестовый дизайн, 208
- тестирование нескольких запусков с помощью сценария оболочки Linux на всех машинах, 210
- анализы результатов анализов, 211
- рабочий процесс тестирования, 208–211
- экземпляры виртуальных машин, 208
- прогнозирование пыльной бури, тестирование облачности, 211–216
- добавление дополнительных экземпляров, 214, 215
- изображения модели окружающей среды здания пыльной бури, 213
- проверка результатов каждой платформы и анализ результатов, 214
- настройка облачной платформы, 214
- разное количество вычислительных узлов от разных сервисов, 212
- требования к вычислениям для прогнозирования пыльной бури, 211–212
- влияние гиперпоточности, 212
- экземпляры рециклинга, 215
- запуск сценария тестирования на каждой платформе, 214
- одинаковое количество вычислительных ресурсов от разных сервисов, 212
- стартовое такое же количество экземпляров, 214
- дизайн теста, 212–213
- анализы результатов анализов, 215–216
- рабочий процесс тестирования, 213–215
- передача сценария тестирования в головной узел, 214
- написание сценария тестовой оболочки, 214
- Информационный центр ГЕОСС, одновременное использование теста интенсивности, 202–207
- анализировать результаты JMeter, 205
- требования центра обмена информацией для компьютерных услуг, 202–203
- Графический интерфейс пользователя, 205
- установка CLN на разные облачные сервисы, 204
- тесты балансировки нагрузки и масштабируемости, 204
- тесты балансировки нагрузки и автомасштабирования, 206
- матрица теста одновременных поисковых запросов, 203
- матричный тест, 206
- работает JMeter, 205
- настройка запроса CSW GetRecords, 204
- настройка плана тестирования JMeter, 205
- тестовый дизайн, 203–204
- анализы результатов анализов, 206–207
- рабочий процесс тестирования, 204–206
- проблемы, 217
- резюме, 216
- тестовая среда, 201–202
- конфигурация вычислительной службы, 202
- сеть, 201–202
- Проверка готовности решений облачных вычислений с открытым исходным кодом, 241–259
- заказчики и оценки программных решений, 242
- прогнозирование пыльной бури, тест на готовность облаков, 255–256
- сравнение облачного промежуточного программного обеспечения, 256
- вычислительные требования для прогнозирования пыльной бури, 255
- НРС против облаков, 255, 256
- сравнение решений с открытым исходным кодом, 255
- тестовый дизайн, 255
- анализы результатов тестов, 256

- тестовый рабочий процесс, 256
- технология виртуализации, 255, 256
- общие приложения, тестирование, 251–254
- сборка образов, 252
- собрать и анализировать результаты, 253
- установить и настроить JRE, 252
- установить и настроить MySQL, 252
- введение тестовых аспектов, 251
- Среда выполнения Java, 251
- Производительность JRE, 251
- Производительность MySQL, 252
- запустить тест, 253
- запускать разные типы виртуальных машин на разных решениях, 253
- дизайн теста, 251–252
- рабочий процесс тестирования, 252–253
- тест производительности JRE, 253
- тест производительности MySQL, 253
- анализы результатов тестов, 253–254
- общая характеристика выбранных решений, 242
- Информационная служба GEOSS, тест готовности к облаку для, 254–255
- вычислительные требования центра обмена информацией, 254
- дизайн тестирования, рабочий процесс и анализ, 254–255
- Характеристики облачных вычислений NIST, 241
- проблемы, 257
- резюме, 256
- тестовая среда, 241–243
- центральный процессор, 243
- Виртуальная машина на основе ядра, 243
- локальные сети, 243
- Объем оперативной памяти, 243
- тесты облачных операций, 243–244
- Патч-файл UBench, 257–258
- виртуальные вычислительные ресурсы, тесты, 244–251
- создание образа с помощью различных программных пакетов для тестирования виртуальных вычислительных ресурсов, 246
- CacheBench, 244
- собрать и анализировать результаты, 249
- Тестирование процессора, 246, 247
- Производительность процессора, 245
- Тестирование ввода-вывода, 246, 248
- Производительность ввода / вывода, 245
- тестирование памяти, 247, 248
- производительность иерархии памяти, 235
- сетевой бенчмаркинг, 247, 248
- сетевая производительность, 245
- выполнение теста, 247
- запуск различных типов виртуальных машин, 247
- тест-дизайн, 245–246
- анализы результатов анализов, 249–251
- рабочий процесс тестирования, 246–249
- Протокол управления коробкой передач, 244
- Протокол дейтаграмм пользователя, 244
- TFS; См. Раздел Сторонний аудит файловой системы Tianwang, стр. 297
- ПОРОГИ, 266
- Файловая система Tianwang (TFS), 30
- ТИГР; См. Топологически интегрированное географическое кодирование и привязку к Tomcat, 128, 149.
- Топографические картографические службы, 113
- Топологически интегрированное географическое кодирование и привязка (TIGER), 264
- Общая стоимость владения (TCO), 297
- TPlatform, 30
- Протокол управления передачей (TCP), 244
- Twitter, 153
- U
- Ubuntu, 114, 195
- Управление энергетической информации США (EIA), 3
- Протокол дейтаграмм пользователя (UDP), 244
- Геологическая служба США (USGS), 125
- USGS; См. Геологическая служба США с использованием облачных вычислений, 51–73.
- Amazon Web Services, 55–66
- разрешить доступ к сети, 55–57
- Бесклассовая междоменная маршрутизация, 57
- создать AMI из запущенного экземпляра, 65
- развернуть приложение, 65

## 332 Индекс

- брандмауэр, 56, 63
- Приложение Hello Cloud, 63
- Доступ HTTP, 57
- установить и настроить веб-сервер, 63
- запустить экземпляр, 57
- войти в экземпляр, 60
- файл открытого ключа, 61
- Файлы PuTTY, 61
- Удаленный рабочий стол, 55
- Secure Shell, 55
- зарегистрируйтесь в Amazon AWS, 55
- передать файл Hello Cloud на экземпляр, 63
- требования к приложению, 51
- развертывание веб-приложения в облачных сервисах, 55–70
  - Amazon Web Services, 55–66
  - Windows Azure, 66–70
  - процессы развертывания, 51
  - обсуждение, 70–72
  - разрешить доступ к сети, 70
- создать образ работающего экземпляра, 72
- развернуть приложение, 72
- установить и настроить веб-сервер, 72
- запустить экземпляр, 72
- войти в экземпляр, 72
- зарегистрироваться в облаке, 70
- перенос файлов приложения на экземпляр, 72
- улучшения, 51
- популярные облачные сервисы, 51–53
- Amazon AWS и Windows Azure, 52–53.
- Магазин эластичных блоков, 53
- графические пользовательские интерфейсы, 52
- Информационные службы Интернета, 53
- введение, 51–52
- Linux VM, 53
- служба реляционной базы данных, 53
- проблемы, 72–73
- резюме, 72
- вариант использования (простое веб-приложение), 53–54
  - HTML-дизайн для веб-приложения
- Hello Cloud, 53–54
- Информационные службы Интернета, 54
- Веб-серверы, 54
- Windows Azure, 66–70
- разрешить доступ к сети, 67
- сделать снимок работающей виртуальной машины, 70
- создать виртуальную машину, 66
- развернуть приложение, 70
- войти в виртуальную машину, 68
- зарегистрироваться в Windows Azure, 66
- перенос файлов Hello Cloud на виртуальную машину, 70
- включить IIS, 68
- Коммунальные вычисления, 10
- Стоимость использования, 297
- V
- Verizon CaaS, 29
- VGE; См. Виртуальную географическую среду
- VIM; См. Менеджер виртуальной инфраструктуры VirtualBox. 38
- Виртуальная кластерная среда, 167
- Виртуальные вычислительные ресурсы, тесты, 244–251
- создание образа с помощью различных программных пакетов для тестирования виртуальных вычислительных ресурсов, 246
- CacheBench, 244
- собирать и анализировать результаты, 249
- Тестирование процессора, 246, 247
- Производительность процессора, 245
- Тестирование ввода-вывода, 246, 248
- Производительность ввода / вывода, 245
- тестирование памяти, 247, 248
- производительность иерархии памяти, 235
- сетевой бенчмаркинг, 247, 248
- сетевая производительность, 245
- выполнение теста, 247
- запуск различных типов виртуальных машин, 247
- тест-дизайн, 245–246
- анализы результатов анализов, 249–251
- рабочий процесс тестирования, 246–249
- Протокол управления коробкой передач, 244
- Протокол дейтаграмм пользователя, 244
- Виртуальная Земля / Карты Bing, 296
- Виртуальная географическая среда (VGE), 296
- Менеджер виртуальной инфраструктуры (VIM), 223
- Виртуализация, 36–39
- голый металл, 212
- вычислительные ресурсы, 13

- географическое присутствие, 179  
оборудование, 11, 12  
Аппаратная виртуальная машина, 85  
реализация, 37  
полная виртуализация, 37  
аппаратная виртуализация, 37  
паравиртуализация, 37  
открытые технологии, 256  
консолидация серверов, 36  
программное обеспечение, 145  
решения, 38–39  
Виртуальная машина на основе ядра, 38  
SmartOS, 39  
VirtualBox, 39  
VMWare, 39  
Зен, 38  
Windows Azure, 300  
Технология Xen для, 28  
Виртуальная машина (VM), 241  
создание, 66  
вход в, 68  
Виртуальная частная сеть (VPN), 28  
Визуальная аналитика, 305  
Визуализация, пиксельная, 276  
VMWare, 39  
VPN; См. Виртуальная частная сеть  
W  
WAN-соединение; См. Широкую  
зону  
Подключение к сети  
Язык описания веб-служб  
(WSDL), 43  
Web X.0, 42–45  
Расширяемый язык разметки, 43  
протокол передачи гипертекста, 42  
Intelligence Web, 43  
Вычисления «сеть как платформа»,  
42  
сервис-ориентированная  
архитектура, 44–45  
услуги через Интернет, 44  
Протокол простого доступа к  
объектам, 43  
Web 1.0, 42  
Web 2.0, 42  
Web 3.0, 42  
Web 4.0, 43  
Веб-сервисы, 43–44  
Язык описания веб-сервисов,  
43  
Подключение к глобальной сети (WAN),  
164  
Windows Azure, 66–70, 188–193, 286  
архитектура, 188–189  
разрешить доступ к сети, 67  
возможности, 189  
сделать снимок работающей виртуальной  
машины, 70  
совместимость, 190  
стоимость, 191  
создать виртуальную машину, 66  
развернуть приложение, 70  
развертывание и интерфейс, 191  
обратная связь от сообщества, 193  
Плоское сетевое хранилище, 189  
общие характеристики, 189–191  
высокопроизводительные вычисления, 190  
гипервизоры, 191  
список основных заказчиков, 192  
войти в виртуальную машину, 68  
основные пользователи и общие  
комментарии, 192–193  
созревание, 193  
Поддерживает ОС, 191  
надежность, 191  
масштабируемость, 189–190  
система самообслуживания, 192  
зарегистрироваться в Windows Azure, 66  
перенос файлов Hello Cloud на  
виртуальную машину, 70  
включить IIS, 68  
сложность использования, 193  
Windows Mobile, ArcGIS for, 116  
Виртуализация Windows Server, 300  
WSDL; См. Язык описания веб-сервисов  
Xen  
XaaS; Смотри на все как на услугу Xen, 38  
Xen Cloud Platform (XCP), 29–30  
XML; См. Расширяемый язык разметки

