



ОБЛАЧНЫЕ ВЫЧИСЛЕНИЯ

O'REILLY®

Джордж Риз



Cloud Application Architectures



George Reese

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Sebastopol • Taipei • Tokyo

Джордж Риз

ОБЛАЧНЫЕ ВЫЧИСЛЕНИЯ

Санкт-Петербург

«БХВ-Петербург»

2011

УДК 681.3.06
ББК 32.973.26-018.2
P49

Риз Дж.

P49 Облачные вычисления: Пер. с англ. — СПб.: БХВ-Петербург, 2011. — 288 с.: ил.
ISBN 978-5-9775-0630-4

Рассмотрена современная бизнес-модель, в которой вычислительные ресурсы предоставляются пользователям как услуги в сети Интернет. Дана практика разработки Web-приложений для развертывания в облачной среде и переноса в нее уже существующих приложений. Рассказано о приемах программирования и о навыках системного администрирования, которыми необходимо овладеть, чтобы успешно разрабатывать и сопровождать приложения, развертываемые в облаке.

Оценена целесообразность переноса существующих приложений в облачную среду как с технической, так и с экономической точек зрения. Описаны построение транзакционных Web-приложений и установка виртуальных серверов для их поддержки. Рассмотрены особенности подготовки плана аварийного восстановления в облачной среде. Показаны преимущества облачной инфраструктуры в области масштабирования приложений.

Для разработчиков

УДК 681.3.06
ББК 32.973.26-018.2

Группа подготовки издания:

Главный редактор	<i>Екатерина Кондукова</i>
Зам. главного редактора	<i>Евгений Рыбаков</i>
Зав. редакцией	<i>Григорий Добин</i>
Перевод с англ.	<i>Ольги Кокоровой</i>
Редактор	<i>Екатерина Капальгина</i>
Компьютерная верстка	<i>Натали Смирновой</i>
Корректор	<i>Наталья Першакова</i>
Оформление обложки	<i>Елены Беляевой</i>
Зав. производством	<i>Николай Тверских</i>

Authorized translation of the English edition of Cloud Application Architectures, ISBN: 978-0-596-15636-7, Copyright © 2009 George Reese. Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472. All rights reserved. Russian edition copyright © 2011 year by BHV – St.Petersburg. All rights reserved. This translation is published and sold by permission of O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

Авторизованный перевод английской редакции книги Cloud Application Architectures, ISBN: 978-0-596-15636-7, Copyright © 2009 George Reese. Издание O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472. Все права защищены. Русская редакция издания выпущена издательством БХВ – Петербург в 2011 году. Все права защищены. Перевод опубликован и продается с разрешения O'Reilly Media, Inc., собственника всех прав на публикацию и продажу издания.

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 30.12.10.

Формат 70×100^{1/16}. Печать офсетная. Усл. печ. л. 23,22.

Тираж 1500 экз. Заказ №

"БХВ-Петербург", 190005, Санкт-Петербург, Измайловский пр., 29.

Санитарно-эпидемиологическое заключение на продукцию № 77.99.60.953.Д.005770.05.09 от 26.05.2009 г. выдано Федеральной службой по надзору в сфере защиты прав потребителей и благополучия человека.

Отпечатано с готовых диапозитивов
в ГУП "Типография "Наука"
199034, Санкт-Петербург, 9 линия, 12

ISBN 978-0-596-15636-7 (англ.)
ISBN 978-5-9775-0630-4 (рус.)

© 2009 George Reese
© Перевод на русский язык "БХВ-Петербург", 2011

Оглавление

ОБ АВТОРЕ	1
ПРЕДИСЛОВИЕ ПЕРЕВОДЧИКА	3
ВВЕДЕНИЕ	5
Целевая аудитория этой книги	5
Структура книги	6
Соглашения, использованные в этой книге	7
Использование примеров кода	7
Safari® Books Online	8
Мы ждем ваших отзывов!	8
Благодарности	9
ГЛАВА 1. ОБЛАЧНЫЕ ВЫЧИСЛЕНИЯ	11
Облако	12
Программное обеспечение	12
Аппаратные средства	15
Преимущества облачной инфраструктуры	15
Аппаратная виртуализация	17
Облачное хранилище	19
Архитектуры облачных приложений	20
Концепция Grid Computing	20
Транзакционные вычисления	22
Значимость облачных вычислений	23
Возможности выбора для инфраструктуры IT	25
Экономические соображения	29
Капитальные затраты	29
Сравнение затрат	30
Нижняя граница	32
Модели облачных инфраструктур	34
Платформа как сервис	34
Инфраструктура как сервис	35
Частные облачные вычислительные среды	36
Комбинация моделей	36
Обзорная информация об Amazon Web Services	37
Amazon Elastic Cloud Compute (EC2)	38
Amazon Simple Storage Service (S3)	39
Amazon Simple Queue Service (SQS)	42

Amazon CloudFront	42
Amazon SimpleDB	43
ГЛАВА 2. ОБЛАЧНАЯ ОБРАБОТКА ДАННЫХ AMAZON	45
Amazon S3	45
Доступ к S3	46
Web-сервисы	46
BitTorrent	47
S3 в действии	48
Amazon EC2	49
Концепции EC2	50
Доступ к EC2	52
Установка экземпляра	53
Доступ к экземпляру	57
Группы безопасности	59
Зоны доступности	61
Статические IP-адреса	62
Хранение данных в EC2	64
Настройка тома EBS	65
Управление томами	67
Моментальные снимки	67
Управление AMI	69
ГЛАВА 3. ПЛАНИРОВАНИЕ ПЕРЕХОДА НА ОБЛАЧНУЮ ОБРАБОТКУ ДАННЫХ	71
Вопросы лицензирования программного обеспечения	71
Переход к модели затрат на облачную среду	74
Подходы к сравнению расценок	74
Пример анализа прибыли на инвестированный капитал	76
На чем еще позволяет экономить облачная инфраструктура	79
Уровни сервиса для облачных приложений	80
Доступность	80
Как оценить доступность вашей системы	81
Из чего складывается доступность?	83
Доступность облачных сервисов	83
Уровни сервиса в Amazon Web Services	84
Ожидаемая доступность в облачной среде	86
Надежность	88
Производительность	89
Организация кластеров или независимые узлы	89
Ограничения по производительности в EC2	90
Безопасность	91
Сложности, связанные с законодательными, организационно-правовыми аспектами и вопросами стандартизации	92
В облачной среде нет периметра	93
Профиль риска для S3 и других облачных хранилищ данных не апробирован ..	93
Ликвидация последствий чрезвычайных происшествий	94

ГЛАВА 4. ПОДГОТОВКА К ПЕРЕХОДУ НА ОБЛАЧНЫЕ ВЫЧИСЛЕНИЯ	97
Разработка Web-приложений	97
Состояние системы и защита транзакций	98
Проблема блокировок в памяти	99
Защита целостности транзакций с помощью хранимых процедур	101
Две альтернативы хранимым процедурам	103
Что происходит при сбоях серверов	106
Построение образов машин	108
Безопасность данных образа машины Amazon	108
Что входит в состав образа машины?	109
Пример образа машины MySQL	112
Философия Amazon AMI	113
Проектирование системы защиты конфиденциальной информации	114
Конфиденциальность в облачной среде	115
Управление шифрованием информации о кредитных картах	117
Обработка транзакции с использованием кредитной карты	118
Что происходит при компрометации системы электронной коммерции	119
Что происходит при компрометации процессора кредитных карт	119
Что происходит, если облачные сервисы Amazon не могут удовлетворить ваши потребности	120
Управление базами данных	123
Объединение в кластеры или репликация?	123
Применение кластеризации баз данных в облачной среде	125
Использование репликации баз данных в облачной среде	126
Использование репликации для повышения производительности	128
Управление главными ключами	130
Генерация глобально уникальных первичных ключей	130
Поддержка глобально уникальных случайных ключей	132
Резервное копирование баз данных	133
Типы резервных копий баз данных	133
Применение стратегии резервного копирования в облачной среде	135
ГЛАВА 5. БЕЗОПАСНОСТЬ	139
Безопасность данных	139
Управление данными	140
Что происходит в случае остановки деятельности облачного провайдера	141
Что происходит, когда судебная повестка вынуждает вашего облачного провайдера раскрыть ваши данные	141
Что происходит, если облачный провайдер не может обеспечить адекватную защиту собственной сети	142
Шифруйте все!	143
Шифрование сетевого трафика	143
Шифрование резервных копий	143
Шифрование файловых систем	144
Соответствие стандартам и нормативно-законодательным актам	145

Сетевая безопасность	148
Правила брандмауэра	149
Системы обнаружения сетевых вторжений	154
Предназначение сетевых систем обнаружения вторжений	154
Реализация системы обнаружения сетевых вторжений в облачной среде	156
Защита хостов	158
Усиление защиты системы	159
Антивирусная защита	159
Системы обнаружения вторжений на уровне хоста	160
Сегментация данных	162
Управление учетной информацией	163
Реакция на компрометацию	164
ГЛАВА 6. АВАРИЙНОЕ ВОССТАНОВЛЕНИЕ	167
Планирование процесса аварийного восстановления	168
Целевая точка восстановления (RPO)	170
Допустимое время восстановления (RTO)	171
Катастрофические события в облачной среде	172
Управление резервным копированием	172
Стратегия в отношении неизменных данных	174
Стратегия в отношении конфигурационных данных	174
Стратегия в отношении сохраняемых данных (резервное копирование баз данных)	176
Безопасность резервного копирования	179
Географическая избыточность	180
Пересечение границ зон доступности	181
Пересечение границ регионов	182
Организационная избыточность	184
Управление нештатными ситуациями	185
Мониторинг	186
Восстановление балансировщика нагрузки	187
Восстановление сервера приложений	188
Восстановление базы данных	188
ГЛАВА 7. МАСШТАБИРОВАНИЕ ОБЛАЧНОЙ ИНФРАСТРУКТУРЫ	191
Планирование мощностей	191
Ожидаемые потребности	192
Определение ожидаемых потребностей	193
Анализ неожиданностей	195
Влияние нагрузки	195
К вопросу об архитектуре приложений и баз данных	196
Шкалы масштабирования	197
Ценность ваших вычислительных мощностей	198
Простой мысленный эксперимент	199
Насколько различными могут оказаться исходы?	200

Масштабирование в облачной среде	201
Средства и системы мониторинга	202
Процесс выделения ресурсов в облачной среде	204
Управление превентивным масштабированием	205
Управление реактивным масштабированием	206
Рекомендуемый подход	207
Вертикальное масштабирование	208
ЗАКЛЮЧЕНИЕ.....	211
ПРИЛОЖЕНИЯ	217
ПРИЛОЖЕНИЕ 1. СПРАВОЧНАЯ ИНФОРМАЦИЯ ПО AMAZON WEB SERVICES	217
Справочная информация по командной строке Amazon EC2	217
Команда <i>ec2-add-group</i>	219
Команда <i>ec2-add-keypair</i>	219
Команда <i>ec2-allocate-address</i>	220
Команда <i>ec2-associate-address</i>	220
Команда <i>ec2-attach-volume</i>	221
Команда <i>ec2-authorize</i>	221
Команда <i>ec2-bundle-instance</i>	222
Команда <i>ec2-cancel-bundle-task</i>	223
Команда <i>ec2-confirm-product-instance</i>	223
Команда <i>ec2-create-snapshot</i>	223
Команда <i>ec2-create-volume</i>	224
Команда <i>ec2-delete-group</i>	224
Команда <i>ec2-delete-keypair</i>	225
Команда <i>ec2-delete-snapshot</i>	225
Команда <i>ec2-delete-volume</i>	225
Команда <i>ec2-deregister</i>	225
Команда <i>ec2-describe-addresses</i>	226
Команда <i>ec2-describe-availability-zones</i>	226
Команда <i>ec2-describe-bundle-tasks</i>	227
Команда <i>ec2-describe-group</i>	227
Команда <i>ec2-describe-image-attribute</i>	228
Команда <i>ec2-describe-images</i>	228
Команда <i>ec2-describe-instances</i>	229
Команда <i>ec2-describe-keypairs</i>	230
Команда <i>ec2-describe-regions</i>	230
Команда <i>ec2-describe-snapshots</i>	230
Команда <i>ec2-describe-volumes</i>	231
Команда <i>ec2-detach-volume</i>	231
Команда <i>ec2-disassociate-address</i>	232
Команда <i>ec2-get-console-output</i>	232

Команда <i>ec2-get-password</i>	233
Команда <i>ec2-modify-image-attribute</i>	233
Команда <i>ec2-reboot-instances</i>	234
Команда <i>ec2-release-address</i>	234
Команда <i>ec2-register</i>	234
Команда <i>ec2-reset-image-attribute</i>	235
Команда <i>ec2-revoke</i>	235
Команда <i>ec2-run-instances</i>	235
Команда <i>ec2-terminate-instances</i>	237
Рекомендации по использованию сервиса Amazon EC2	237
Шифрование на уровне файловой системы	238
Настройка RAID для использования множества томов EBS	241
ПРИЛОЖЕНИЕ 2. ОБЛАЧНЫЙ ХОСТИНГ GOGRID	244
Типы облаков	244
Облачные центры	246
Центры обработки данных в облачных средах	247
GoGrid и традиционные центры обработки данных	247
Горизонтальное и вертикальное масштабирование	248
Архитектуры для развертывания GoGrid	250
Web-приложения — в центре внимания	251
Сравнение подходов	252
Непосредственное сравнение	252
Практическое использование	254
Что лучше подойдет вам?	254
ПРИЛОЖЕНИЕ 3. КОМПАНИЯ RACKSPACE	255
Облачные серверы Rackspace	256
Сервис Cloud Files	257
Сервис Cloud Sites	259
Полная интеграция с обеспечением круглосуточной непрерывной технической поддержки	259
ПРИЛОЖЕНИЕ 4. ТЕРМИНОЛОГИЧЕСКИЙ ГЛОССАРИЙ	260
СПИСОК ЛИТЕРАТУРЫ И РЕСУРСОВ ИНТЕРНЕТА	269
Русскоязычные ресурсы Интернета	270
ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ	271

Об авторе

Джордж Риз (George Reese) является основателем двух компаний, расположенных в Миннеаполисе — enStratus Networks LLC (поставщик элитного инструментария, предназначенного для управления облачной инфраструктурой) и Valtira LLC (поставщик одноименной онлайн-платформы управления маркетингом). В течение последних 15 лет Джордж написал целый ряд книг об информационных технологиях, выпущенных издательством O'Reilly, в том числе: "MySQL Pocket Reference", "Database Programming with JDBC and Java" и "Java Database Best Practices".

На протяжении эры господства Интернета Джордж занимался созданием корпоративного инструментария для разработчиков и построением решений для маркетинга. Он являлся влиятельным участником процесса развития многопользовательских компьютерных игр из разряда так называемых "многопользовательских миров". Он является автором ряда библиотек Open Source MUD¹, и в 1996 году разработал первый драйвер JDBC — Open Source mSQL-JDBC. В последнее время Джордж занимается развертыванием транзакционных Web-приложений в облачной среде. До того как заняться разработкой Web-приложений, Джордж работал над моделированием краш-тестов для Национального управления по безопасности автомобильного движения США (National Highway Traffic Safety Administration, NHTSA).

Джордж имеет степень бакалавра прикладных наук (философия), полученную в колледже Бейтса (Bates College), г. Льюистон, штат Мэн (<http://www.bates.edu/>), и степень магистра бизнеса (M.B.A.), полученную в Школе Менеджмента Келлог (Kellogg School of Management), г. Эванстон, штат Иллинойс (<http://www.kellogg.northwestern.edu/>). В настоящее время он проживает в штате Миннесота с женой Моникой и дочерьми Кирой и Линдси.

В процессе работы над книгой автору помогали его друзья и коллеги Рэнди Байас (Randy Bias) из компании GoGrid и Эрик "Е. J". Джонсон (Eric "E. J" Johnson) из компании Rackspace. Во-первых, они рецензировали авторский текст, посвященный работе с Web-сервисами Amazon (Amazon Web Services), и, кроме того, они написали приложения, в которых рассказали о предложениях компаний GoGrid и Rackspace соответственно.

¹ MUD (Multi User Dungeon, Dimension или Domain, русский вариант — "Многопользовательский мир", МПМ). — *Прим. перев.*

- ◆ Приложение 2 "Облачный хостинг GoGrid" написано Рэнди Байесом.
Рэнди Байес — вице-президент по технологическим стратегиям GoGrid, подразделения ServePath (<http://www.servepath.com/>). Он является признанным экспертом в области облачных вычислений и инфраструктур центров обработки данных, обладающим более чем 20-летним опытом в области предоставления услуг Интернета (ISP), предоставления услуг хостинга и разработки крупномасштабных инфраструктур. На все эти темы он часто пишет в своем личном блоге: <http://neotactics.com/blog>.
- ◆ Приложение 3 "Компания Rackspace" написано Эриком Джонсоном.
Эрик Джонсон (Eric Johnson, или просто "E. J." — псевдоним, под которым он известен еще шире) является менеджером по развитию и разработке системы хранения данных Rackspace Cloud Files. К его дополнительной работе относятся исследования, проводимые в Racklabs, исследовательско-инженерном подразделении Rackspace.
В течение последних 15 лет Эрик работает исключительно с использованием технологий Open Source, в том числе таких, как системное администрирование UNIX, сетевые технологии, администрирование баз данных и разработка программного обеспечения. Он выполняет исследовательские работы в области воздушного сообщения и аэрокосмических технологий. В течение целого ряда лет он вносит свой вклад в сообщество Open Source, разрабатывая патчи для SSH, поддерживая пакеты для Arch Linux и разрабатывая технические руководства для DNS/Bind.
Эрик имеет степени бакалавра естественных наук (Bachelor of Science, B. S.) в области электротехники, полученную в Университете им. Дрекселя (Drexel University, Филадельфия, штат Пенсильвания, см. <http://www.drexel.edu/>), и магистра естественных наук (Master of Science, M. S.) в области компьютерных технологий, полученную в Политехническом институте Ренсселира (Rensselaer Polytechnic Institute, Трой, Нью-Йорк, см. <http://rpi.edu/>, http://en.wikipedia.org/wiki/Rensselaer_Polytechnic_Institute). В настоящее время вместе с семьей проживает в Сан-Антонио, штат Техас.

Предисловие переводчика

Облачные вычисления (cloud computing) — довольно известное на сегодняшний день явление, о котором говорится очень и очень много. Фактически это то, чем почти каждый из нас пользуется ежедневно. Это и почтовые сервисы, например, такие как Gmail, Yahoo! Mail, Webmail, Hotmail и др. Это и онлайн-текстовые редакторы — скажем, Zoho Writer или Документы Google. И даже онлайн-сервисы по работе с графикой — например, LunarPic или всем известное онлайн-приложение Google Picasa, не говоря уже о набравших популярность музыкальных и видеосервисах. Их существует великое множество — всевозможных развлекательных и офисных либо творческих приложений на базе облачных технологий. Но есть и рынок облачных вычислений, где действительно крутятся огромные деньги!

И именно этому рынку как раз и посвящена книга, которую вы держите в руках. В ней рассматривается современная бизнес-модель, в которой вычислительные ресурсы предоставляются пользователям как услуги в сети Интернет. На простом и понятном примере самого известного "облака" компании Amazon объяснены как основные концепции, так и нюансы организации облачных вычислений, дано большое количество практических советов и полезных идей. Эта книга является признанным на Западе бестселлером, и одновременно с этим — первой книгой на русском языке, посвященной облачным вычислениям.

Несколько слов о том, что вы найдете в этой книге, и чего вы в ней не найдете. Основное внимание в книге уделено рассмотрению реальных решений, позволяющих сэкономить капиталовложения и трудозатраты, повысить надежность вычислительной инфраструктуры — поэтому ее можно рекомендовать, в первую очередь, системным и сетевым архитекторам, администраторам, разработчикам ПО, работающим в предприятиях малого и среднего бизнеса, государственных учреждениях. Сотрудников отделов ИТ крупных предприятий должны заинтересовать описанные способы повышения производительности и отказоустойчивости их вычислительных мощностей.

Большая часть материалов книги излагается на примере облачных сервисов компании Amazon — одного из ведущих представителей облачных провайдеров. Но при этом рассказывается и об универсальных подходах, используемых, например, такими провайдерами, как GoGrid и Rackspace. К сожалению, объять необъятное невозможно, поэтому в книге не описываются, например, такие платформы,

как Microsoft Azure (<http://www.microsoft.com/windowsazure/>)¹ или Eucalyptus (Elastic Utility Computing Architecture Linking Your Programs To Useful Systems, открытый проект, реализующий облачные вычисления в пределах центра управления данными, на котором концентрируются разработчики Ubuntu Server²). Чтобы отчасти скомпенсировать этот недостаток, группа подготовки издания дополнила книгу списком литературы, в котором освещаются различные аспекты облачных вычислений, а также списком русскоязычных ресурсов Интернета, которые помогут читателям сориентироваться в различных аспектах облачных вычислений в соответствии со стоящими перед ними конкретными задачами. Эта информация поможет читателям узнать, как обстоят дела с развитием облачных вычислений в России, и подскажет им, как максимально эффективно применять облачные сервисы в практической повседневной работе.

Зато огромным достоинством этой книги является концентрация внимания на таких вопросах, которые, безусловно, волнуют всех — и рядовых пользователей, и клиентов интернет-магазинов, и системных архитекторов, и разработчиков ПО. Как несложно догадаться, это вопросы по поводу конфиденциальности информации, находящейся за пределами ваших компьютеров, будь то домашний компьютер рядового пользователя Интернета или сервер компании. Ведь если вы перемещаете свою IT-деятельность в чужой центр обработки данных, вы кладете ее в "черный ящик". Применительно к облачным сервисам — вам не просто непонятно, что внутри, как оно работает, но и где ваши данные расположены физически. Наконец, еще один важный вопрос — а будет ли этот облачный сервис продолжать работать и завтра, и после завтра, и в более отдаленном будущем. Сможете ли вы вообще извлечь свои данные в случае наступления каких-либо форс-мажорных обстоятельств. Все эти вопросы изложены достаточно подробно, и это одно из важных достоинств книги, которые делают ее интересной для всех, кто интересуется облачными вычислениями.

¹ Подробнее о Microsoft Azure см. <http://www.osp.ru/pcworld/2010/09/13004178/>, <http://www.azuresupport.com/>. — *Прим. перев.*

² См. интервью с Сореном Хансеном (Søren Hansen) из команды виртуализации Ubuntu в журнале Linux Format (апрельский номер 2010 года), подробнее см. <https://help.ubuntu.com/community/Eucalyptus>, <http://packages.ubuntu.com/ru/lucid/eucalyptus-common>. — *Прим. перев.*

Введение

Облачными вычислениями (Cloud computing) на сегодняшний день интересуются все, и это действительно перспективное направление! Облачные вычисления представляют собой высокоэффективный инструмент повышения прибыли и расширения каналов продаж для независимых производителей программного обеспечения (Independent Software Vendors), операторов связи и VAR-посредников, расширяющих возможности существующих продуктов с целью их перепродажи конечным пользователям. Облачный подход позволяет организовать динамическое предоставление услуг, когда пользователи могут производить оплату по факту и регулировать объем своих ресурсов в зависимости от реальных потребностей без долгосрочных обязательств.

Благодаря облачным вычислениям компании могут предоставлять конечным пользователям удаленный динамический доступ к услугам, вычислительным ресурсам и приложениям (включая операционные системы и инфраструктуру) через Интернет. Вычислительные облака состоят из тысяч серверов, размещенных в физических и виртуальных центрах обработки данных, обеспечивающих работу десятков тысяч приложений, которые одновременно используют миллионы пользователей. По сравнению с традиционным подходом, облачные вычисления позволяют управлять более крупными инфраструктурами, обслуживать различные группы пользователей в пределах одного облака. Они обеспечивают огромный потенциал роста, но при этом дают возможность добиваться существенной экономии на масштабах.

Индустрия облачных вычислений стремительно развивается, и, по прогнозам аналитиков, к 2012 году на ее долю будет приходиться 9 % всех расходов на информационные инфраструктуры и услуги. Каким бы ни был бизнес вашей компании — разработка программного обеспечения, предоставление хостинга (hosting) или услуг связи, маркетинг или реклама — ваши клиенты наверняка тоже ожидают от вас движения в этом направлении.

Целевая аудитория этой книги

Я написал эту книгу, ориентируясь на технологов различных уровней подготовки и находящихся на всех ступенях "карьерной лестницы". Книга будет полезна

разработчикам, которым требуется писать код для приложений, входящих в облачные инфраструктуры, системным архитекторам, занятым проектированием систем для работы в облачных инфраструктурах, IT-менеджерам, отвечающим за перенос систем в облака.

В книге нет больших фрагментов кода, но время от времени я все же привожу примеры кодирования, демонстрирующие мой подход к решению различных задач. В основном я программирую на Java и Python и использую базы данных MySQL, хотя иногда прибегаю и к таким базам данных, как SQL Server или Oracle. Вместо того, чтобы приводить большие фрагменты кода на Java, я предпочел излагать методики, подходящие для любого языка программирования.

Если вы проектируете, строите или поддерживаете Web-приложения для развертывания в облаке, значит, эта книга — для вас.

Структура книги

Глава 1 универсальна и предназначена для широкого круга читателей. В ней описывается то, что я понимаю под "облаком", и поясняю, почему облако имеет такую ценность для предприятий и организаций. Эта глава дает материал на таком уровне и таким языком, чтобы финансовые директора ваших компаний (Chief Financial Officers, CFO) без проблем могли ее прочесть и понять, в чем ценность и важность облаков.

В *главе 2* приводится небольшой ознакомительный курс об облаке Amazon.

Цель данной книги заключается в предоставлении наилучших примеров, иллюстрирующих использование облачной инфраструктуры, вне зависимости от того, конкретно какую реализацию вы используете. На основании собственного опыта я могу сказать, что на данный момент именно облако Amazon и Amazon Web Services занимают самую внушительную рыночную нишу. По этой причине я считаю, что необходимо дать клиентам представление о том, как быстрее всего начать работу с облаком Amazon, а также предоставить наиболее общие сведения, необходимые для обсуждения дальнейших тем, освещаемых в данной книге, плюс определения базовых терминов, которыми мы впоследствии будем оперировать.

Если вас интересуют другие реализации облачной инфраструктуры, то и эту информацию я тоже предоставляю. Мне оказали помощь и содействие некоторые друзья и коллеги из таких компаний, как Rackspace и GoGrid. Эрик "Е. J". Джонсон (Eric Johnson) из Rackspace выступил в качестве научного редактора книги и консультанта по вопросам, касающимся совместимости описываемого ПО с предложениями от компании Rackspace, а Ренди Байас (Randy Bias) из GoGrid сделал то же самое для предложений от своей компании.

Глава 3 содержит предварительную информацию, необходимую для того, чтобы разработать план вашей компании по переходу на облачную инфраструктуру, проанализировать этот план и приступить к его осуществлению.

Главы 4—7 предоставляют подробную информацию о построении Web-приложений для работы в облаке. Глава 4 описывает архитектуры, основанные на транзакциях Web-приложений, и изменения, которые они должны претерпеть в облаке. Глава 5 описывает основные проблемы, связанные с концепциями безопасности облачных вычислений. В главе 6 рассказывается о том, как наилучшим образом спланировать и подготовить процедуры аварийного восстановления и обеспечить аварийное восстановление в кратчайшие сроки. Наконец, в главе 7 мы обсудим, как облачные вычисления влияют на перспективы масштабирования приложений, включая автоматическое масштабирование Web-приложений.

Соглашения, использованные в этой книге

В книге используются следующие соглашения по оформлению текста:

- ◆ *курсив* — так выделяются новые термины, впервые введенные в книге;
- ◆ **полужирный шрифт** — так выделяются Web-адреса и адреса электронной почты;
- ◆ моноширинный шрифт с засечками — так выделяются имена файлов и папок, их содержимое, вывод команд, и, в общем случае, — все, что может содержаться в программах;
- ◆ **полужирный моноширинный шрифт с засечками** — так выделены команды и любой другой текст, который вводится с клавиатуры пользователем, а также фрагменты кода или содержимого файлов, которые обсуждаются в данный момент;
- ◆ **полужирный моноширинный шрифт курсивного начертания** — так выделяется текст, который должен быть заменен на информацию, предоставляемую пользователем.

Использование примеров кода

Основное предназначение этой книги — помочь вам выполнить вашу работу. В общем случае, вы можете применять код, приведенный в этой книге, в ваших программах и документации. За исключением тех случаев, когда вы копируете большие объемы кода, вам не обязательно обращаться в издательство O'Reilly за разрешением на копирование. Например, для использования небольших фрагментов приведенного в книге кода получения разрешения не требуется. С другой стороны, распространение примеров из книг O'Reilly на CD-ROM разрешения требует. Если вам нужно ответить на вопрос, и в вашем ответе вы цитируете данную книгу, то на это разрешения не требуется. Включение больших объемов текста книги в сопроводительную документацию к вашим программным продуктам разрешения требует.

Мы очень ценим, но не требуем в обязательном порядке указания полной информации об этой книге. Как правило, при ссылке на книгу указывается ее название, автор, издатель и ISBN, например: *"Cloud Application Architectures by George Reese"*. Copyright 2009 George Reese, 978-0-596-15636-7.

Если вы чувствуете, что объем цитирования выходит за рамки правомерного использования, обозначенного в этом разделе, вы можете обращаться за разрешением на использование материалов по следующему адресу: permissions@oreilly.com.

Safari® Books Online

Если вы видите на обложке вашей любимой книги, посвященной информационным технологиям, значок Safari® Books Online, это означает, что данная книга доступна через сервис O'Reilly Network Safari Bookshelf.

Safari предлагает даже лучшее решение, чем электронные книги (e-books). Safari® Books Online — это виртуальная библиотека, которая позволяет с легкостью искать нужную информацию по тысячам книг об информационных технологиях, копировать и вставлять в ваши тексты образцы кода, скачивать главы из книг и быстро находить ответы на интересующие вас вопросы, получая наиболее точную и актуальную информацию. Попробуйте бесплатно протестировать этот сервис по следующему адресу: <http://my.safaribooksonline.com>.

Мы ждем ваших отзывов!

Редакция O'Reilly проверила информацию, приведенную в этой книге, и протестировала образцы кода, но никто из нас не безупречен, поэтому ошибки и недосмотры все же случаются. Если вы обнаружите какие-нибудь ошибки или неточности, а также захотите поделиться с нами своими пожеланиями на будущее, пожалуйста, пишите нам:

O'Reilly Media, Inc.

1005 Gravenstein Highway North

Sebastopol, CA 95472

800-998-9938 (в США или Канаде)

707-829-0515 (международное или местное отделение)

707-829-0104 (факс)

У нас есть Web-страница, заведенная для этой книги, на которой мы публикуем списки опечаток, примеры и дополнительную информацию. Эта страница имеет следующий адрес:

<http://www.oreilly.com/catalog/9780596156367>

Если вы хотите задать технический вопрос или что-либо прокомментировать, пишите по следующему адресу:

bookquestions@oreilly.com

Дополнительную информацию о наших книгах, конференциях, программному обеспечению, центрах ресурсов (Resource Centers) и сети O'Reilly (O'Reilly Network) можно получить на нашем Web-сайте:

<http://www.oreilly.com>

Благодарности

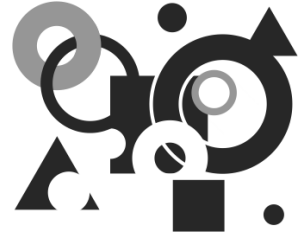
В этой книге рассматривается такое количество научных дисциплин и такое множество технологий, что я просто не смог бы написать ее полностью самостоятельно.

В первую очередь я хотел бы выразить безмерную благодарность за помощь, которую оказали мне Рэнди Байас (Randy Bias) из компании GoGrid и Эрик "Е. J". Джонсон (Eric "E. J" Johnson) из компании Rackspace. Мой опыт работы с облачной инфраструктурой ограничивался работой с Web-сервисами Amazon (Amazon Web Services), поэтому Рэнди и Эрик потратили довольно долгое время на рецензирование и проверку фрагментов книги, посвященных AWS. Также они написали приложения, в которых рассказали о предложениях компаний GoGrid и Rackspace соответственно.

Далее, я крайне признателен всем, кто вычитывал главы книги и предоставлял подробные комментарии: Джону Оллспо (John Allspaw), Джеффу Барру (Jeff Barr), Кристоферу Хоффу (Christopher Hoff), Тео Шлосснейглу (Theo Schlossnagle) и Джеймсу Уркхарту (James Urquhart). Каждый из них обладает уникальным опытом и привнес в книгу ценные технические материалы. Благодаря их труду и критическим замечаниям книга стала намного полезнее, чем могла бы оказаться.

В дополнение к этому, еще целый ряд специалистов предоставил свои замечания по избранным частям этой книги, в том числе: Дэвид Бэгли (David Bagley), Морган Кэтлин (Morgan Catlin), Майк Хорвот (Mike Horwath), Моника Риз (Monique Reese), Стейси Релофс (Stacey Roelofs) и Джон Вьегга (John Viega).

Наконец, я выражаю отдельную благодарность сотрудникам O'Reilly: Энди Орэму (Andy Oram) и Изабель Канкл (Isabel Kunkle). Я неоднократно говорил и писал это, и не только в данной книге, но я должен упомянуть это и здесь: благодаря их редактированию я стал гораздо лучшим писателем.



ГЛАВА 1

Облачные вычисления

Отличительным признаком любого "модного" слова, которое у всех на слуху, является то, что оно может передавать не реальное понимание его точного значения, а лишь видимость этого понимания. Термин "облако" (cloud) является как раз таким словом — многие люди просто произносят его, не понимая его истинного смысла.

Оно используется в самых разных, не согласующихся между собой контекстах, и часто им обозначают совершенно разные вещи и понятия. В одном из обсуждений люди могут говорить о Google Gmail; в другом — о Web-сервисе Amazon Elastic Compute Cloud (Amazon EC2) — но здесь, по крайней мере, термин "облако" все же присутствует в названии сервиса.

С другой стороны, термин "облачные вычисления" (cloud computing) является "модным словечком" ничуть не более, чем им является термин Web. Облачные вычисления представляют собой результат эволюции множества различных технологий, которые в комбинации изменили организационный подход к построению информационной инфраструктуры предприятия. Как и в случае с Web (а это произошло чуть более десятилетия тому назад), в облачных вычислениях нет абсолютно ничего фундаментально нового. Многие из технологий, вошедших в состав Web, существовали десятилетиями до появления компании Netscape Communications, которая объединила их и сделала универсально доступными. По аналогии с этим событием, большинство технологий, которые в совокупности и формируют облачные вычисления, существовали уже в течение десятков лет. Чтобы появился термин "облачные вычисления", потребовалось, чтобы компания Amazon объединила их и сделала их комбинацию массово доступной.

Цель данной книги заключается в том, чтобы дать возможность разработчикам Web-приложений на базе транзакций использовать облачную инфраструктуру в качестве средства для развертывания их приложений. Поэтому в книге основное внимание уделяется таким облакам, как Amazon EC2, и в меньшей степени — другим, таким как Google Gmail. Тем не менее, обсуждение мы начнем с описания общей инфраструктуры для облачных вычислений.

Облако

Облако — это не просто новейший модный термин, применяющийся для описания интернет-технологий удаленного хранения данных. Хотя Интернет представляет собой основной фундамент, необходимый для облака, облако — это нечто большее, чем просто Интернет. *Облако* — это то, куда вы обращаетесь, когда вам требуется использовать ту или иную технологию, и пользуетесь ею до тех пор, пока она вам нужна, и ни минутой дольше. Для этого вам не требуется ничего устанавливать на вашем компьютере и вы не платите за технологию, когда вы не пользуетесь ею.

Облако может означать как программное обеспечение, так и инфраструктуру. Оно может быть приложением, доступ к которому предоставляется через Web, или сервером, к которому вы обращаетесь в точности тогда, когда вам это требуется. Вне зависимости от того, является ли сервис программным или аппаратным, вам нужен критерий, позволяющий определить, является ли этот сервис облачным. Этот критерий формулируется так:

"Если для доступа к сервису вы можете зайти в любую библиотеку или интернет-кафе, сесть за любой компьютер, не предъявляя никаких особых требований ни к операционной системе, ни к браузеру, и получить доступ к сервису, то этот сервис является облачным".

Я сформулировал три условия, по которым в дальнейшем обсуждении мы будем определять, является ли тот или иной сервис облачным:

- ◆ сервис доступен через Web-браузер (непатентованный и не являющийся собственностью той или иной фирмы) или при помощи специального интерфейса прикладного программирования (API) для доступа к Web-сервисам;
- ◆ чтобы начать пользоваться сервисом не требуется никаких затрат капитала;
- ◆ вы платите только за то, чем вы пользуетесь, и оплачиваете только то время, в течение которого вы этим пользуетесь.

Я не считаю, что три перечисленных условия полностью закрывают вопрос, но они представляют собой твердое основание, позволяющее вести дальнейшее обсуждение, и полностью отражают мое понимание облачных сервисов, которые рассматриваются в данной книге.

Если вам не нравится мое сжатое и концентрированное определение облачных сервисов, вы можете обратиться к определению, которое Джеймс Гавернор (James Governor) дал в своем блоге: запись называется "15 Ways to Tell It's Not Cloud Computing", а найти ее можно по следующему адресу: <http://www.redmonk.com/jgovernor/2008/03/13/15-ways-to-tell-its-not-cloud-computing>.

Программное обеспечение

Как я уже упоминал ранее, облачные сервисы подразделяются на программные сервисы и сервисы инфраструктуры. С точки зрения зрелости технологии, про-

граммное обеспечение в облаке развито значительно лучше, чем аппаратная составляющая.

Программное обеспечение как услуга (Software as a Service, SaaS), в сущности, представляет собой термин, призванный обозначать программную составляющую в облаке. Большинство систем SaaS, хотя и не все, являются облачными системами.

SaaS представляет собой модель развертывания ПО на основе Web, благодаря которой программное обеспечение оказывается полностью доступным через Web-браузер. Для пользователей систем SaaS не имеет значения, где установлено программное обеспечение, какую операционную систему оно использует и на каком языке оно написано — PHP, Java или .NET. И, самое главное — вам самим нет необходимости устанавливать что-либо и где-либо.

Например, Gmail представляет собой не что иное, как программу электронной почты, доступную через браузер. Она предоставляет все те же функциональные возможности, что и Apple Mail или Outlook, но при этом для пользования ею не нужен так называемый "толстый" клиент¹. Даже если ваш домен не получает почту через Gmail, вы все равно можете пользоваться Gmail для доступа к вашей почте.

SalesForce.com — это еще один вариант SaaS. SalesForce.com представляет собой систему управления отношениями с клиентами (CRM), предназначенную для крупного предприятия. Она предназначена для специалистов из отделов продаж и позволяет им отслеживать потенциальных заказчиков и менеджеров по закупкам, определять их место в организации процесса продаж, управлять всем процессом заключения сделки — от первого контакта до заключения договора и проведения расчета, и т. д. Как и в случае с Gmail, вам не требуется устанавливать никакое дополнительное программное обеспечение для получения доступа к SalesForce.com: достаточно только ввести адрес SalesForce.com в браузере, завести учетную запись, и можно приступать к работе.

Системы SaaS обладают некоторыми определяющими характеристиками, в том числе:

- ◆ *Доступность через Web-браузер.* Программное обеспечение типа SaaS никогда не требует установки каких-либо дополнительных программ на ваш ноутбук или настольный компьютер. Доступ к системам SaaS осуществляется через Web-браузер с использованием открытых стандартов или универсальный плагин браузера. Облачные вычисления и патентованное ПО, являющееся собственностью какой-либо компании, просто не сочетаются между собой.
- ◆ *Доступность по требованию.* Чтобы получить доступ к системе SaaS, вам не требуется полностью проходить процесс сделки в классическом понимании этого слова. Как только вы заведете учетную запись, вы сможете получать дос-

¹ "Толстый" клиент (fat client, thick client, иногда rich client) — в архитектуре "клиент-сервер" представляет собой приложение, предоставляющее (в противовес "тонкому" клиенту) расширенные функциональные возможности, вне зависимости от центрального сервера. При таком подходе сервер выполняет роль хранилища данных, а вся работа по обработке и представлению этих данных переносится на клиентский компьютер. — *Прим. перев.*

туп к нужному вам программному обеспечению в любой момент и из любой географической точки.

- ❖ *Условия оплаты зависят от использования.* SaaS не требует никаких вложений в инфраструктуру или какой-то особой настройки, поэтому вам не потребуются начинать работу с серьезных капиталовложений. Платить вы будете только за те виды сервиса, которыми вы пользуетесь, и только за фактическое время использования. Если сервис вам больше не нужен, вы просто прекращаете за него платить.
- ❖ *Минимальные требования к инфраструктуре IT.* Если вам не требуется покупать никаких серверов и нет необходимости строить сеть, то зачем вам вообще нужна какая-то инфраструктура IT? Хотя для конфигурирования систем SaaS и может иногда требоваться некоторый минимальный уровень технических знаний (например, для управления DNS в Google Apps), но этот уровень не выходит за рамки, характерные для обычного опытного пользователя. Высококвалифицированный IT-администратор для этого не требуется.

Одной из особенностей некоторых развернутых систем SaaS, которую я нарочно пока пропустил, является так называемая "многоарендная архитектура" (multitenancy)¹. Многие поставщики SaaS хвастаются тем, что их системы предоставляют "многоарендные услуги", а некоторые даже утверждают, что "многоарендная" архитектура является основным требованием к системе SaaS.

Многоарендное приложение представляет собой программу-сервер, которая поддерживает развертывание множества клиентов, одновременно пользующихся одним и тем же экземпляром программы-сервера. Эта модель предлагает поставщику SaaS очевидные преимущества в том, что позволяет обеспечить для конечных пользователей следующие возможности:

- ❖ поддержку большего количества клиентов на меньшем количестве аппаратных компонентов;
- ❖ ускоренную и упрощенную процедуру развертывания обновлений для приложений и обновления для системы безопасности;
- ❖ в целом более надежную архитектуру.

¹ "Многоарендная архитектура" (multitenancy) — это специальный прием программирования или архитектурное решение, поддерживающее одновременное использование одного и того же экземпляра программы несколькими клиентами. В основе этого термина лежит слово "tenant", которое буквально означает "жилец" или "арендатор". Вообще-то при описании multitenancy сразу же возникает ассоциация с таким бытовым явлением, как "коммунальная квартира", и, наверное, можно было бы назвать такую архитектуру действительно "коммунальной". Однако это слово уже "занято" — оно уже использовано для перевода термина utility computing (принцип предоставления информационных сервисов как коммунальных услуг или "коммунальные вычисления"). Впрочем, в отличие от физической коммунальной квартиры, виртуальная облачная "квартира" имеет раздвижные перегородки, и ее конфигурацию можно менять по мере необходимости. Таким образом, термин multitenancy можно интерпретировать как технологическое решение, позволяющее нескольким пользователям независимо друг от друга разделять один и тот же ресурс, не нарушая при этом конфиденциальности и защиты принадлежащих им данных. — *Прим. перев.*

Максимальный выигрыш для конечного пользователя заключается в косвенном уменьшении платы за сервис, более быстром доступе к новым функциональным возможностям и (иногда) ускоренном исправлении слабых мест в системе безопасности. Однако за счет того, что базовым принципом облачных вычислений является незаинтересованность конечного пользователя в архитектуре приложения, лежащего в основе используемого сервиса, значимость многоарендной архитектуры в перспективе будет уменьшаться.

Как будет показано в следующем разделе, технологии виртуализации в принципе делают архитектурные преимущества многоарендности неактуальными.

Аппаратные средства

В общем случае, большинству людей концептуально гораздо сложнее понять и принять аппаратные средства в облаке, нежели программное обеспечение в облаке. Аппаратные средства — это нечто материальное, что вы можете потрогать: это ваша собственность, и вы не сдаете ее напрокат. Если ваш сервер пострадает в результате пожара — это бедствие касается только вас. Многим людям тяжело смириться с тем, что их аппаратные средства принадлежат не только им, и отказаться от возможности увидеть и физически потрогать свое собственное оборудование.

Если аппаратные средства находятся в облаке, вы запрашиваете новый "сервер" по мере необходимости. Обычно он будет вам доступен менее, чем через 10 минут. Когда вы заканчиваете с ним работать, вы освобождаете ресурс, и он возвращается в облако. При этом вы не имеете ни малейшего представления, какой физический сервер предоставляет вам облачный сервис, и, скорее всего, вы не знаете даже его конкретного физического местоположения.

БАРЬЕР СТАРЫХ ОЖИДАНИЙ

Для меня, как для поставщика облачных сервисов, труднее всего ответить на вопрос: "Так где же находятся наши серверы?". Фактически честный ответ на этот вопрос будет таким: "Я не знаю — может быть, где-нибудь на восточном побережье США или где-нибудь в Западной Европе", но многие клиенты, получив такой ответ, почувствуют себя крайне неудобно. Однако это отсутствие информации о физическом местоположении ваших серверов позволяет почувствовать интересное ощущение физической безопасности, поскольку для мотивированных злоумышленников это делает почти невозможной физическую атаку на ваше оборудование с целью его компрометации.

Преимущества облачной инфраструктуры

Задумайтесь о таких факторах, о которых вы должны заботиться, если вы имеете аппаратные средства в собственности и должны обеспечивать их работу.

◆ Мощность серверов стала недостаточной?

Планирование необходимой мощности и обеспечение ресурсами всегда играет важную роль. Облачные вычисления упрощают для вас решение следующих двух

проблем, которые вам необходимо решать в том случае, если аппаратные средства находятся в вашей собственности: что делать, если вы неправильно оценили возможности вашего оборудования (чересчур оптимистично или слишком пессимистично)? Что делать, если у вас нет финансовых средств для расширения бизнеса на тот момент, когда встает проблема покупки нового оборудования? Если вы управляете собственной инфраструктурой, вам потребуется собрать большие суммы наличными на каждую новую сеть устройств хранения данных (Storage Area Network, SAN) или каждый приобретаемый сервер. Кроме того, вам потребуется довольно значительное время на ввод в эксплуатацию — от момента принятия решения до размещения заказа, его оплаты, физической доставки, приемки, монтажа, инсталляции ПО, тестирования и, наконец, ввода в эксплуатацию.

◆ Что произойдет в случае возникновения проблемы?

Естественно, любой высококачественный сервер имеет некоторый резерв, позволяющий безболезненно пережить некоторые типичные аппаратные проблемы. Но, даже если у вас есть, например, резервный жесткий диск, предназначенный на замену отказавшего диска в составе массива RAID, все равно кто-то должен заменить отказавший диск, управлять процедурой RMA¹, а затем установить новый диск на сервер. Для этого требуется не только время, но и высокая квалификация, и при этом вам необходимо провести все эти работы в сжатые сроки, чтобы избежать полного выхода сервера из строя.

◆ Что произойдет в случае чрезвычайной ситуации?

Если сервер окончательно выйдет из строя, то, если только вы не имеете инфраструктуры с высокой степенью отказоустойчивости, вы столкнетесь с бедствием, и все ваши сотрудники должны будут действовать в условиях форс-мажорных обстоятельств, предпринимая все усилия для устранения последствий возникшей аварийной ситуации. В таких случаях вам остается только надеяться на то, что у вас есть качественная и актуальная резервная копия, а ваш план аварийного восстановления проработан достаточно хорошо, чтобы провести восстановление в кратчайшие возможные сроки. Восстановление — это почти всегда ручной процесс.

◆ Что делать, если сервер вам больше не нужен?

Вполне возможно, что ваши потребности в мощности используемого оборудования со временем изменятся. Кроме того, вполне возможно, что вам необходимо будет вывести из эксплуатации устаревший и обесцененный сервер. Что делать с этим старым сервером? Даже если вы хотите его полностью списать, все равно кто-то должен будет этим заниматься. Если же сервер обесценился не полностью, ваша компания понесет убытки из-за машины, которая совершенно бесполезна для вашего бизнеса.

¹ RMA (Return Merchandise Authorization) — процедура возврата неисправного или некачественного изделия поставщику или производителю. Когда вам нужно заменить дефектный компонент, вы обычно должны выполнить определенную последовательность действий по взаимоотношениям с поставщиком (составить заявку на возврат и замену, переслать поставщику, затем провести переговоры о ремонте/замене и, наконец, получить в качестве замены исправное изделие).

❖ Как решить вопросы с недвижимостью и электроснабжением?

Если вы управляете собственной инфраструктурой (или даже имеете собственную стойку у вашего интернет-провайдера), вам может потребоваться оплачивать недвижимость или электроэнергию, не используя при этом большую часть оплачиваемых ресурсов. Это мало того, что неэкологично, но и представляет собой абсолютно непродуктивную трату денег.

При использовании облачной инфраструктуры ни один из следующих аспектов не представляет проблемы, потому что:

- ❖ вы добавляете мощностей в облачную инфраструктуру только на тот момент, когда она вам нужна, и ни минутой раньше. Вы не несете никаких затрат, ассоциированных с выделением ресурсов, поэтому вам нет необходимости беспокоиться о синхронизации потребностей в мощностях и бюджетными нуждами. Наконец, вы можете нарастить мощность за считанные минуты, и это позволит вам не потерять лицо даже в сложных ситуациях;
- ❖ вам вообще нет необходимости беспокоиться об аппаратных средствах, на которых все это работает. Вы можете никогда не узнать о том, что физический сервер, на котором фактически выполнялась ваша работа, полностью отказал. Если вы правильно подберете инструментарий, вы можете добиться автоматического восстановления после сложнейших аварийных ситуаций, а ваша команда в это время может попросту спать;
- ❖ если ваши потребности в мощностях изменяются, вам может потребоваться другая виртуальная аппаратная конфигурация. В этом случае вы просто отказываетесь от вашего виртуального сервера и получаете другой. При этом вам нет необходимости задумываться о перепродаже, утилизации или беспокоиться о вреде для окружающей среды;
- ❖ вам нет необходимости платить за недвижимость и электроэнергию, которыми вы не пользуетесь. Поскольку вы используете лишь часть гораздо больших аппаратных мощностей, нежели вам нужно, вы повышаете эффективность использования физического пространства, необходимого для обеспечения ваших потребностей в обработке информации. Более того, вы не платите за всю стойку серверов, потребляющих электроэнергию, где большинство циклов центрального процессора являются холостыми.

Аппаратная виртуализация

Аппаратная виртуализация (hardware virtualization) представляет собой технологию, которая позволяет большинству поставщиков услуг облачных вычислений, в том числе и Amazon Web Services (AWS), предлагать свои услуги¹. Если у вас

¹ Существуют и другие подходы к реализации облачной инфраструктуры, включая предоставление аппаратных средств по требованию через такие компании, как AppNexus и NewClouds. Кроме того, некоторые поставщики услуг, скажем, такие, как GoGrid (см. приложение 2), предлагают гибридные решения.

есть компьютер Mac, на котором вы запускаете Windows или Linux в таких эмуляторах, как Parallels или Fusion, то вы используете технологию виртуализации, аналогичную той, что применяется для реализации облачных вычислений. Благодаря виртуализации администратор ИТ может подразделить физический сервер на любое количество виртуальных серверов, каждый из которых работает под управлением собственной операционной системы и каждому из которых выделяются такие ресурсы, как память, CPU, участки дискового пространства. Некоторые технологии виртуализации даже позволяют вам перемещать работающие экземпляры виртуальных серверов с одного физического сервера на другой. С точки зрения пользователя или приложения, которые работают на виртуальном сервере, не существует никаких возможностей определить, является ли сервер, на котором они работают, виртуальным или физическим.

Ряд виртуализационных технологий, доступных на рынке, используют различные подходы к проблеме виртуализации. Решение Amazon представляет собой расширение популярной системы виртуализации на основе открытого кода, которая называется Xen¹. Xen предоставляет компонент, называемый *гипервизором* (hypervisor), на котором могут работать одна или несколько гостевых операционных систем. Гипервизор создает уровень аппаратных абстракций, который позволяет операционным системам совместно использовать ресурсы физического сервера, при этом гостевые операционные системы не имеют физического доступа к этим ресурсам, как "своим", так и "чужим".

Общеизвестное возражение против виртуализации — особенно со стороны тех, кто работал с эмуляторами на настольных компьютерах под управлением настольных операционных систем — заключается в том, что виртуализированные системы существенно теряют в производительности. Но в облаке это общепринятое предубеждение против виртуализации не имеет под собой оснований по следующим причинам:

- ◆ даже пониженная производительность оборудования вашего поставщика облачных услуг, по всей вероятности, окажется лучше, чем максимальная производительность вашего потребительского сервера, и это при том условии, что он оптимальным образом настроен;
- ◆ технологии виртуализации, предназначенные для крупных предприятий, например, такие как Xen и VMware (<http://www.vmware.com/>), используют так называемую *паравиртуализацию* (paravirtualization), а также возможности *аппаратной виртуализации* (hardware-assisted virtualization capabilities), разработанные различными производителями процессоров. Это позволяет им добиться почти такой же производительности, как и при работе приложений на "живом железе".

¹ Официальный сайт проекта: <http://www.xen.org/>, подробные описания см. по адресам: http://wiki.xensource.com/xenwiki/Xen_on_4_app_servers, <http://ru.wikipedia.org/wiki/Xen>, <http://www.vmg.ru/citrix-xen>. — Прим. перев.

Облачное хранилище

Абстрагирование от аппаратных средств в облаке осуществляется не только благодаря замене физических серверов виртуальными. Виртуализации подлежат и системы физического хранения данных.

Облачное хранилище позволяет вам перебрасывать данные в облако, и при этом не беспокоиться о том, как именно они хранятся, и не задумываться об их резервном копировании. Когда данные, перемещенные в облако, понадобятся вам снова, вам достаточно будет просто обратиться в облако и получить свои данные. При этом вы можете не знать, как хранятся эти данные, где они хранятся, или того, что происходит с тем или иным оборудованием, когда вы периодически перемещаете данные в облако и извлекаете их оттуда.

Как и в случае с другими элементами облачных вычислений, на рынке предлагается несколько подходов к облачному хранилищу. Говоря общими словами, они связаны с разбиением ваших данных на небольшие цепочки и хранение их на множестве серверов. Цепочки данных снабжаются индивидуально вычисленными контрольными суммами, так, чтобы данные можно было быстро восстановить, вне зависимости от того, что могло бы произойти в течение времени хранения с накопителями, физически хранящими данные, и скомпрометировать облако.

Я не раз был свидетелем того, как люди, начинающие работать с облаком, пытались обращаться с облачным хранилищем так, как если бы оно представляло собой сетевую накопитель. С точки зрения принципа работы облачное хранилище принципиально отличается от традиционных сетевых накопителей, и служит принципиально другим целям. Облачное хранилище имеет тенденцию работать гораздо медленнее и имеет гораздо бóльшую степень структурированности, вследствие чего его использование в качестве оперативного хранилища данных непрактично, вне зависимости от того, работает ли приложение, использующее эти данные, в облаке или где-то еще.

Вообще говоря, облачное хранилище не подходит для оперативного использования облачными приложениями на базе транзакций. В последующих главах мы подробнее обсудим роль облачного хранилища в инфраструктуре приложений, работающих на базе транзакций. Пока же об облачном хранилище можно думать примерно так же, как об аналоге резервной копии на ленточном носителе, но, в отличие от системы резервного копирования с ленточным приводом, при работе в облаке вам не нужны ни привод, ни ленты.

ПРИМЕЧАНИЕ

Компания Amazon недавно выступила с новым коммерческим предложением, которое называется Amazon CloudFront. Amazon CloudFront использует Amazon S3 в качестве средства для реализации сети, предназначенной для распределения информационного содержимого. Идея, на которой основывается Amazon CloudFront, заключается в репликации вашего информационного содержимого по периферии сети. Хотя облачное хранилище Amazon S3 не подходит для использования с Web-приложениями на базе транзакций в оперативном режиме, в будущем оно обещает стать ключевым компонентом систем быстрого распределения статического контента.

Архитектуры облачных приложений

Мы могли бы еще долго обсуждать технологии виртуализации (кстати, знаете ли вы, что вы можете комбинировать и совместно использовать не менее пяти различных типов виртуализации?), но центральной темой этой книги являются не эти вопросы, а написание приложений, которые наилучшим образом используют преимущества облачных вычислений. Именно к этой теме мы сейчас и перейдем.

Концепция Grid Computing

Концепция Grid Computing¹ представляет собой архитектуру приложений, предоставляющую самый простой метод перехода к облачной архитектуре. Приложения, использующие *грид-технологии*, представляют собой ПО, интенсивно потребляющее ресурсы процессора. *Грид-приложения* разбивают операции по обработке данных на небольшие последовательности элементарных операций, которые могут выполняться изолированно.

В общем, если вы когда-либо принимали участие в проекте **SETI@home**, то вы уже принимали участие в грид-вычислениях. Проект по поиску внеземных цивилизаций SETI (Search for Extra-Terrestrial Intelligence) использует радиотелескопы, ведущие постоянное наблюдение за активностью в космосе. Эти радиотелескопы собирают гигантские объемы данных, которые затем нуждаются в обработке с целью поиска сигналов, отличающихся от естественных — обнаружение сигнала, который может иметь искусственное происхождение, может служить признаком деятельности внеземной цивилизации. Обработка такого объема информации на одном компьютере потребует такого длительного времени, что вполне может стать, человечество к завершению этих вычислений действительно уже освоит межзвездные перелеты. Но вот множество компьютеров, каждый из которых использует для решения этой проблемы свои холостые циклы CPU, могут справиться с задачей намного быстрее.

Эти компьютеры, на которых работает приложение **SETI@home** (возможно, включая и ваш собственный компьютер), формируют сеть или "решетку" (grid). Каждый раз, когда у них есть свободные такты процессора, компьютеры направляют запросы серверам проекта SETI на получение наборов данных для обработки. Они обрабатывают наборы данных и передают результаты обратно серверам проекта SETI. Результаты, полученные одним из участников проекта, перепроверяются повторной обработкой на компьютерах других участников, а затем производится дальнейшая проверка наиболее интересных результатов [1].

¹ В русскоязычных ресурсах встречается и такое написание, как "грид" (англ. *grid* — решетка, сеть) — согласованная, открытая и стандартизованная компьютерная среда, которая обеспечивает гибкое, безопасное, скоординированное разделение вычислительных ресурсов и ресурсов хранения информации, которые являются частью этой среды, в рамках одной виртуальной организации. См., например: <http://tinyurl.com/37mx4jn>, <http://gridclub.ru/>. — Прим. перев.

В 1999 г. участники программы SETI предложили использовать холостые циклы настольных компьютеров обычных пользователей для обработки данных проекта. Коммерческие и правительственные организации сформировали сеть из суперкомпьютеров для выполнения той же задачи. Впоследствии были созданы пулы серверов (так называемые "серверные фермы", *server farms*) для формирования сети, выполняющей такие задачи, как рендеринг видео (визуализация изображений и обсчет последовательностей кадров). Как суперкомпьютеры, так и серверные фермы оказались очень дорогими подходами к грид-вычислениям и требовали крупных капиталовложений.

Облачная инфраструктура серьезно упрощает и удешевляет создание грид-приложений. Если у вас есть данные, нуждающиеся в обработке, вы просто запрашиваете сервер для обработки этих данных. После завершения обработки этот сервер может быть остановлен или может запросить для обработки новую порцию данных.

На рис. 1.1 проиллюстрировано, как работает грид-приложение. Сначала сервер или кластер серверов получает набор данных, которые нуждаются в обработке. На первом шаге эти данные передаются в очередь сообщений (1). Другие узлы, которые часто называются серверам обработки (или, как в случае **SETI@home** — другими настольными компьютерами), наблюдают за очередью сообщений (2) и ждут появления новых наборов данных. Когда набор данных появляется в очереди сообщений, он обрабатывается первым же компьютером, который его обнаружил, и результаты отсылаются обратно в очередь сообщений (3), откуда они считываются сервером или кластером серверов (4). Оба компонента могут работать независимо друг от друга, и каждый из них может работать даже в том случае, если второй компонент не работает ни на одном компьютере.

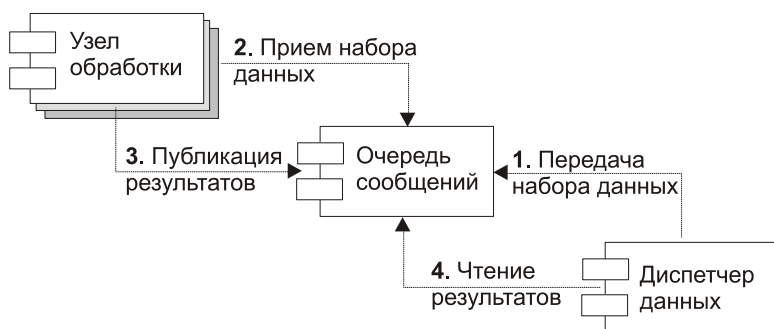


Рис. 1.1. Архитектура грид-приложений отделяет основное приложение от узлов, осуществляющих обработку данных

Здесь приходят на помощь облачные вычисления, потому что при их использовании вы не должны иметь никаких серверов в личной собственности, и когда у вас нет данных для обработки, вам не требуется никаких серверов вообще. Таким образом, вы можете масштабировать выделяемые вам вычислительные мощности,