





# ОГЛАВЛЕНИЕ

<b>ПРЕДИСЛОВИЕ</b> .....	<b>10</b>
<b>БЛАГОДАРНОСТИ</b> .....	<b>12</b>
<b>ОБ ЭТОЙ КНИГЕ</b> .....	<b>14</b>
Структура книги .....	15
Графическое выделение и загрузка исходного кода .....	15
<b>АВТОР В СЕТИ</b> .....	<b>16</b>
<b>ОБ АВТОРЕ</b> .....	<b>17</b>
<b>ОБ ИЛЛЮСТРАЦИИ НА ОБЛОЖКЕ</b> .....	<b>18</b>
<b>ЧАСТЬ 1.</b>	
<b>Нadoop – каркас распределенного программирования</b> .....	<b>19</b>
<b>ГЛАВА 1. Введение в Hadoop</b> .....	<b>21</b>
1.1. Зачем написана книга «Hadoop в действии»? .....	22
1.2. Что такое Hadoop? .....	23
1.3. Сравнение Hadoop с другими распределенными системами .....	24
1.4. Сравнение СУБД на основе SQL с Hadoop .....	26
1.5. Знакомство с MapReduce .....	29
1.5.1. Масштабирование простой программы вручную .....	30
1.5.2. Масштабирование той же программы с помощью MapReduce .....	33
1.6. Подсчет слов с помощью Hadoop – ваша первая программа .....	36
1.7. История Hadoop .....	43
1.8. Резюме .....	44
1.9. Ресурсы .....	45

<b>ГЛАВА 2. Запуск Hadoop .....</b>	<b>46</b>
2.1. Структурные элементы Hadoop .....	46
2.1.1. NameNode .....	47
2.1.2. DataNode .....	47
2.1.3. Secondary NameNode .....	49
2.1.4. JobTracker .....	49
2.1.5. TaskTracker .....	50
2.2. Настройка SSH для кластера Hadoop .....	52
2.2.1. Определение общей учетной записи .....	52
2.2.2. Проверка правильности установки SSH .....	52
2.2.3. Генерация пары ключей .....	53
2.2.4. Распространение открытого ключа и проверка возможности входа в систему .....	53
2.3. Запуск Hadoop .....	54
2.3.1. Локальный (автономный) режим .....	55
2.3.2. Псевдораспределенный режим .....	56
2.3.3. Полностью распределенный режим .....	58
2.4. Веб-интерфейс для мониторинга кластера .....	62
2.5. Резюме .....	63
<b>ГЛАВА 3. Компоненты Hadoop .....</b>	<b>65</b>
3.1. Работа с файлами в системе HDFS .....	65
3.1.1. Основные команды для работы с файлами .....	66
3.1.2. Чтение и запись в HDFS из программы .....	71
3.2. Анатомия MapReduce-программы .....	74
3.2.1. Типы данных в Hadoop .....	76
3.2.2. Распределитель .....	78
3.2.3. Редуктор .....	79
3.2.4. Разбивка — направление выхода распределителя ....	80
3.2.5. Комбинатор — локальная редукция .....	81
3.2.6. Подсчет слов с помощью готовых классов распределителя и редуктора .....	81
3.3. Чтение и запись .....	83
3.3.1. Интерфейс InputFormat .....	85
3.3.2. Интерфейс OutputFormat .....	91

3.4. Резюме .....	93
-------------------	----

## **ЧАСТЬ 2.**

<b>Hadoop в действии .....</b>	<b>95</b>
--------------------------------	-----------

### **Глава 4. Создание простых MapReduce-программ . 97**

4.1. Получение набора данных о патентах .....	98
4.1.1. Данные о цитировании патентов .....	99
4.1.2. Данные об описаниях патентов .....	101
4.2. Определение шаблона MapReduce-программы .....	102
4.3. Подсчет всякой всячины .....	108
4.4. Адаптация к изменениям в API Hadoop .....	114
4.5. Интерфейс Hadoop Streaming .....	118
4.5.1. Интерфейс Streaming и команды Unix .....	119
4.5.2. Streaming и скрипты .....	120
4.5.3. Интерфейс Streaming и пары ключ/значение .....	126
4.5.4. Интерфейс Streaming и пакет Aggregate .....	131
4.6. Повышение производительности с помощью комбинаторов.....	137
4.7. Упражнения .....	142
4.8. Резюме .....	144
4.9. Дополнительные ресурсы .....	145

### **ГЛАВА 5. Углубленное изучение MapReduce ..... 147**

5.1. Сцепление задач MapReduce .....	148
5.1.1. Последовательное сцепление задач MapReduce .....	148
5.1.2. Сцепление задач MapReduce со сложными зависимостями.....	148
5.1.3. Включение в цепочку шагов пред- и постобработки	149
5.2. Соединение данных из разных источников.....	154
5.2.1. Соединение на стороне редуктора .....	155
5.2.2. Построение реплицированных соединений с помощью класса DistributedCache .....	166
5.2.3. Полусоединение: соединение на стороне редуктора с фильтрацией на стороне распределителя .....	171
5.3. Создание фильтра Блума .....	173

5.3.1. Что делает фильтр Блума?.....	173
5.3.2. Реализация фильтра Блума .....	176
5.3.3. Фильтр Блума в Hadoop версии 0.20+ .....	184
5.4. Упражнения .....	184
5.5. Резюме .....	187
5.6. Дополнительные ресурсы .....	187
<b>ГЛАВА 6. Практическое программирование .....</b>	<b>189</b>
6.1. Разработка MapReduce-программ .....	190
6.1.1. Локальный режим.....	191
6.1.2. Псевдораспределенный режим.....	197
6.2. Мониторинг и отладка в производственном кластере .....	203
6.2.1. Счетчики .....	203
6.2.2. Пропуск плохих записей .....	205
6.2.3. Перезапуск сбойных заданий с помощью IsolationRunner .....	210
6.3. Оптимизация производительности .....	211
6.3.1. Уменьшение сетевого трафика с помощью комбинатора.....	212
6.3.2. Уменьшение объема выходных данных.....	212
6.3.3. Использование сжатия .....	213
6.3.4. Повторное использование JVM .....	216
6.3.5. Наблюдаемое исполнение.....	217
6.3.6. Переработка кода и модификация алгоритмов.....	219
6.4. Резюме .....	221
<b>ГЛАВА 7. Сборник рецептов .....</b>	<b>222</b>
7.1. Передача нестандартных параметров задаче .....	222
7.2. Получение информации о конкретном задании.....	226
7.3. Разбиение на несколько выходных файлов .....	227
7.4. Ввод и вывод в базу данных .....	234
7.5. Сортировка выходных данных .....	236
7.6. Резюме .....	238
<b>ГЛАВА 8. Администрирование Hadoop .....</b>	<b>239</b>
8.1. Практическая настройка параметров .....	240

8.2. Проверка состояния системы .....	243
8.3. Установка прав доступа .....	245
8.4. Управление квотами .....	246
8.5. Включение корзины .....	247
8.6. Удаление узлов DataNode.....	247
8.7. Добавление узлов DataNode .....	249
8.8. Управление узлами NameNode и Secondary NameNode...	250
8.9. Восстановление после сбоя узла NameNode.....	252
8.10. Проектирование топологии сети и осведомленность о стойках .....	254
8.11. Планирование задач, поступающих от нескольких пользователей.....	257
8.11.1. Организация нескольких узлов JobTracker.....	257
8.11.2. Справедливый планировщик .....	258
8.12. Резюме .....	261

### ЧАСТЬ 3.

## **Hadoop в реальной жизни ..... 263**

### **ГЛАВА 9. Эксплуатация Hadoop в облаке ..... 265**

9.1. Введение в Amazon Web Services.....	266
9.2. Настройка AWS .....	267
9.2.1. Получение учетных данных для аутентификации в AWS.....	268
9.2.2. Получение командных утилит .....	271
9.2.3. Подготовка пары ключей для работы с SSH .....	273
9.3. Настройка Hadoop в EC2 .....	275
9.3.1. Задание параметров защиты .....	275
9.3.2. Конфигурирование типа кластера .....	276
9.4. Запуск MapReduce-программ в среде EC2.....	278
9.4.1. Перенос своего кода в кластер Hadoop .....	279
9.4.2. Доступ к данным из кластера Hadoop .....	279
9.5. Очистка и останов экземпляров EC2 .....	285
9.6. Amazon Elastic MapReduce и другие службы AWS.....	285
9.6.1. Amazon Elastic MapReduce.....	286

9.6.2. AWS Import/Export.....	287
9.7. Резюме .....	288
<b>ГЛАВА 10. Программирование с помощью Pig .....</b>	<b>289</b>
10.1. Научитесь думать по-свински.....	290
10.1.1. Язык описания потоков данных.....	290
10.1.2. Типы данных .....	291
10.1.3. Определенные пользователем функции .....	291
10.2. Установка Pig .....	291
10.3. Запуск Pig .....	293
10.3.1. Управление оболочкой Grunt .....	294
10.4. Изучение языка Pig Latin с помощью Grunt .....	295
10.5. Учимся говорить на Pig Latin.....	302
10.5.1. Типы данных и схемы.....	302
10.5.2. Выражения и функции .....	304
10.5.3. Реляционные операторы .....	307
10.5.4. Оптимизация исполнения.....	317
10.6. Определяемые пользователем функции .....	317
10.6.1. Использование UDF.....	318
10.6.2. Создание UDF .....	319
10.7. Работа со скриптами.....	322
10.7.1. Комментарии .....	322
10.7.2. Подстановка параметров .....	323
10.7.3. Режим многозапросного исполнения .....	324
10.8. Pig в действии: отыскание похожих патентов.....	326
10.9. Резюме .....	332
<b>ГЛАВА 11. Hive и другие .....</b>	<b>333</b>
11.1. Hive .....	334
11.1.1. Установка и настройка Hive.....	335
11.1.2. Примеры запросов .....	338
11.1.3. Детали языка HiveQL .....	342
11.1.4. Hive: подводя итоги .....	352
11.2. Другие проекты, связанные с Hadoop.....	353
11.2.1. HBase .....	353

11.2.2. ZooKeeper .....	353
11.2.3. Cascading .....	354
11.2.4. Cloudera .....	354
11.2.5. Katta .....	355
11.2.6. CloudBase.....	355
11.2.7. Aster Data и Greenplum.....	356
11.2.8. Hama и Mahout .....	356
11.2.9. search-hadoop.com.....	356
11.3. Резюме .....	357
<b>ГЛАВА 12. Примеры применения .....</b>	<b>358</b>
12.1. Преобразование 11 миллионов изображений из архива газеты New York Times .....	358
12.2. Добыча данных в компании China Mobile .....	360
12.3. Рекомендование лучших веб-сайтов на StumbleUpon....	367
12.3.1. Как мы пришли к распределенной обработке в StumbleUpon .....	368
12.3.2. HBase и StumbleUpon .....	369
12.3.3. Другие применения Hadoop на сайте StumbleUpon.....	379
12.4. Построение аналитической системы для внутрикорпоративного поиска – проект IBM ES2 .....	381
12.4.1. Архитектура ES2 .....	386
12.4.2. Робот ES2.....	387
12.4.3. Аналитические средства в ES2 .....	390
12.4.4. Выводы .....	400
12.4.5. Библиография.....	401
<b>ПРИЛОЖЕНИЕ. Команды HDFS.....</b>	<b>403</b>
<b>ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ .....</b>	<b>408</b>





# ПРЕДИСЛОВИЕ

Уже давно данные очаровывают меня. Еще на младших курсах, изучая электротехнику, я открыл для себя цифровую обработку сигналов и прикипел к ней всей душой. Я понял, что музыку, фотографии и множество других вещей можно рассматривать как данные. Оказалось, что создавать и усиливать эмоциональные переживания можно с помощью вычислений. Ничего интереснее я даже представить себе не мог.

Со временем передо мной раскрылись новые, не менее удивительные аспекты данных. Вот уже несколько лет, как я занимаюсь социальными сетями и большими массивами данных. Именно большие массивы бросают вызов моему интеллекту. Статистический анализ данных я освоил уже давно, а для работы с новыми типами данных были нужны «всего лишь» новые математические методы. Конечно, это не просто, но по крайней мере я этому учился, а уж в ресурсах для пополнения знаний недостатка и вовсе нет. С другой стороны, для работы с большими объемами данных необходим совсем другой системный подход и новые способы программирования. Этому меня не учили, но – и это даже важнее – не один я оказался в таком положении. Знания о практических методах обработки больших массивов данных сродни черной магии. Существует немало инструментов и приемов масштабируемой обработки данных, в частности, кэширование (например, с помощью программы memcached), репликация, секционирование и, конечно, MapReduce/Hadoop. Последние несколько лет я потратил на приобретение и совершенствование навыков в этой области.

Лично для меня наиболее сложной оказалась средняя часть кривой обучения. Поначалу не так уж трудно найти блоги с вводным материалом и презентации, объясняющие, как написать пример уровня «Здравствуй, мир». А приобретя начальные знания, вы уже можете задавать дополнительные вопросы в списках рассылки, общаться со специалистами на различных встречах и конференциях и даже самостоятельно читать исходный код. Но вот в середине зияет гигантский

провал —аппетит уже разыгрался, а как сформулировать последующие вопросы, неясно. Эта проблема стоит особенно остро для таких новейших технологий, как Nadoor. Необходимо некое упорядоченное изложение, которое начиналось бы с примера «Здравствуй, мир» и доводило читателя до точки, в которой он смог бы без особых усилий применить Nadoor на практике. Именно так я и представляю себе эту книгу. К счастью, я обнаружил, что серия «In Action», выходящая в издательстве Manning, отлично согласуется с моей целью, и к тому же там есть прекрасные редакторы, помогавшие мне на всем протяжении работы.

Я получал удовольствие, когда писал книгу, и надеюсь, что она станет для вас началом увлекательного путешествия в мир Nadoor.



## ОБ ЭТОЙ КНИГЕ

Hadoop – это каркас с открытым исходным кодом, в котором реализован алгоритм распределения и редукции MapReduce, лежащий в основе подхода Google к организации запросов к распределенным наборам данных, которые и составляют Интернет. В связи с таким определением возникает очевидный вопрос: что такое *распределение* (map) и зачем нужна последующая *редукция* (reduce)? Зачастую массивные наборы данных с трудом поддаются анализу и опросу традиционными средствами, особенно когда сами запросы весьма сложны. По существу, алгоритм MapReduce разбивает запрос и набор данных на несколько частей – это этап распределения. Отдельные компоненты запроса можно обработать параллельно, а затем *свести* полученные результаты воедино (*редуцировать*).

Читатель этой книги узнает, как использовать Hadoop и писать MapReduce-программы. Книга предназначена для программистов, архитекторов и руководителей проектов, связанных с обработкой большого объема данных в офлайн-режиме. Будет описано, как получить копию Hadoop, как организовать кластер и как писать программы анализа. Мы начнем с применения Hadoop в конфигурации по умолчанию к решению нескольких простых задач, например, об анализе изменения частоты вхождения слов в корпус документов; это позволит уяснить основные идеи Hadoop и MapReduce. Далее мы перейдем к базовым концепциям MapReduce-приложений, разрабатываемых с помощью Hadoop, и по ходу дела изучим компоненты каркаса, применение Hadoop к широкому спектру задач анализа данных и многочисленным примерам Hadoop в действии.

Алгоритм MapReduce сложен как концептуально, так и в плане реализации, а от пользователей каркаса Hadoop требуется изучить все средства и хитрости, необходимые для работы с ним. В этой книге не просто рассказывается о том, как запустить Hadoop; прочитав ее, вы научитесь писать полезные программы с использованием MapReduce.

Предполагается, что читатель хотя бы немного владеет языком Java, поскольку именно на нем написано большинство примеров.

Знакомство с основами математической статистики (например, гистограммами и корреляцией) поможет разобраться в более сложных примерах обработки данных.

### Структура книги

Эта книга состоит из 12 глав, разбитых на три части.

Часть 1 состоит из трех глав и представляет собой введение в каркас Hadoop. Здесь излагаются те базовые сведения, которые необходимо знать для понимания и использования каркаса. Описывается, из каких аппаратных компонентов состоит кластер Hadoop, рассказывается об установке и конфигурировании системы. Также в части 1 дается общее представление о каркасе MapReduce и приводится пример первой MapReduce-программы.

Часть 2 «Hadoop в действии» состоит из пяти глав, в которых описаны практические навыки, необходимые для составления и запуска программ обработки данных в среде Hadoop. Здесь мы рассмотрим многочисленные примеры применения Hadoop к анализу набора данных о патентах, в том числе и такие нетривиальные алгоритмы, как фильтр Блума. Мы также поговорим о приемах программирования и администрирования, чрезвычайно полезных при работе с Hadoop в производственной среде.

Часть 3 «Hadoop в реальной жизни», в которую входят последние четыре главы, посвящена обширной экосистеме, сложившейся вокруг Hadoop. Существуют облачные службы, которые позволяют обойтись без покупки собственного оборудования для создания кластера. Имеются также дополнительные пакеты, предлагающие высокоуровневые абстракции, надстроенные над MapReduce. Наконец, мы рассмотрим несколько примеров практического применения Hadoop к решению реальных задач бизнеса.

В приложении приведен список команд HDFS.

### Графическое выделение и загрузка исходного кода

Исходный код в листингах и в основном тексте набран моноширинным шрифтом. Многие листинги сопровождаются аннотациями, в которых излагаются важные концепции. В некоторых случаях в листингах присутствуют нумерованные маркеры, с которыми соотносятся последующие пояснения.

Код всех приведенных в книге примеров можно скачать с сайта издательства по адресу [www.manning.com/HadoopinAction](http://www.manning.com/HadoopinAction).



## **АВТОР В СЕТИ**

Приобретение книги «*Hadoop в действии*» открывает бесплатный доступ к закрытому форуму, организованному издательством Manning Publications, где вы можете оставить свои комментарии к книге, задать технические вопросы и получить помощь от автора и других пользователей. Получить доступ к форуму и подписаться на список рассылки можно на странице [www.manning.com/HadoopinAction](http://www.manning.com/HadoopinAction). Там же написано, как зайти на форум после регистрации, на какую помощь можно рассчитывать, и изложены правила поведения в форуме.

Издательство Manning обязуется предоставлять читателям площадку для общения с другими читателями и автором. Однако это не означает, что автор обязан как-то участвовать в обсуждениях; его присутствие на форуме остается чисто добровольным (и не оплачивается). Мы советуем задавать автору хитроумные вопросы, чтобы его интерес к форуму не угасал!

Форум автора в сети и архивы будут доступны на сайте издательства до тех пор, пока книга не перестанет печататься.



## ОБ АВТОРЕ

В настоящее время Чак Лэм занят организацией социальной сети для пользователей мобильных устройств под названием *RollCall*. Это будет социальный секретарь для активных людей.

Раньше Чак работал техническим руководителем проекта в компании RockYou. В этой должности он разрабатывал социальные приложения и инфраструктуру обработки данных, рассчитанную на сотни миллионов пользователей. Он применял A/B тестирование и статистический анализ для оптимизации скорости распространения социального приложения среди пользователей (его *виральности*). Ему удавалось значительно – иногда на порядок – повысить частоту посещения страниц.

Обработкой больших объемов данных Чак заинтересовался во время работы над кандидатской диссертацией в Стэнфордском университете. Он узнал о том, какой серьезный эффект большие массивы данных могут оказать на машинное обучение, и начал изучать последствия. В его диссертации «Вычислительные методы сбора данных» впервые были исследованы новые подходы к сбору данных для машинного обучения – с применением идей, заимствованных из программ с открытым исходным кодом и онлайн-овых игр.



# Часть 1

## **Hadoop – каркас распределенного программирования**

**В** части 1 закладываются основы для понимания и использования Hadoop. Мы расскажем, из каких аппаратных компонентов состоит кластер Hadoop, а также опишем процедуру установки и конфигурирования, необходимую для создания работоспособной системы. Мы дадим общее представление о каркасе MapReduce, после чего вы напишете и запустите свою первую MapReduce-программу.







# ГЛАВА 1.

## Введение в Hadoop

### *В этой главе:*

- Основы составления масштабируемых, распределенных программ для обработки данных.
- Что представляют собой Hadoop и MapReduce.
- Создание и запуск простой MapReduce-программы.

В современном мире данные окружают нас со всех сторон. Мы загружаем на серверы видео, делаем фотографии с помощью мобильных телефонов, посылаем текстовые сообщения друзьям, изменяем их статус в системе Facebook, оставляем комментарии на разных сайтах, щелкаем по рекламным баннерам и т. д. Компьютеры со своей стороны также порождают и хранят все больше и больше данных. Возможно, даже эта книга представлена у вас в электронном виде, и читаете вы ее на экране своего компьютера. И уж точно факт покупки этой книги отражен в базе данных какого-нибудь розничного продавца<sup>1</sup>.

Проблемы, вызванные экспоненциальным ростом данных, первыми ощутили на себе такие находящиеся на переднем крае технологий компании, как Google, Yahoo, Amazon и Microsoft. Им приходилось просеивать терабайты и петабайты данных, чтобы понять, какие веб-сайты популярны, какие книги пользуются спросом, какая реклама находит отклик у пользователей. Имеющиеся инструменты оказались не приспособлены к обработке столь больших объемов данных. Google стала первой компанией, начавшей пропагандировать систему *MapReduce*, которую применила для масштабирования обработки данных. Эта система вызвала большой интерес, так как многие другие компании уже столкнулись с аналогичными проблемами, а заново изобретать собственный инструментарий было нерентабельно.

<sup>1</sup> Вы ведь читаете легальную копию, правда?

Дуг Каттинг разглядел открывающиеся возможности и принялся за разработку версии MapReduce с открытым исходным кодом, которую назвал Hadoop. Вскоре Yahoo и другие компании объединили усилия в поддержку этого начинания. Ныне Hadoop составляет основную часть вычислительной инфраструктуры многих работающих в веб компаний, в частности, Yahoo, Facebook, LinkedIn и Twitter. Примериваются к ней и более традиционные организации, например, средства массовой информации и телекоммуникационные компании. В главе 12 описаны примеры использования Hadoop в таких компаниях, как New York Times, China Mobile и IBM.

Hadoop и, более общо, технологии распределенной обработки больших массивов данных, быстро становятся важным умением для широкого круга программистов. Сегодня программисту для эффективной работы необходимо знать о реляционных базах данных, сетях и безопасности, хотя еще пару десятков лет назад все это считалось факультативными навыками. Так и базовые знания о распределенной обработке данных скоро станут неременной частью багажа любого программиста. Ведущие университеты, например Стэнфордский и Карнеги-Меллона, уже ввели в программу факультетов информатики курс по изучению Hadoop. Эта книга поможет вам, программисту-практику, быстро освоить Hadoop и начать использовать его для обработки собственных наборов данных.

Эта глава содержит несколько более формальное введение в каркас Hadoop, рассматриваемый в контексте распределенных систем и систем обработки данных. Мы дадим обзор модели программирования MapReduce. Простой пример программы подсчета слов с помощью имеющихся инструментов поможет уяснить, какие проблемы возникают при обработке больших массивов данных. Затем вы напишете программу, основанную на использовании Hadoop, что позволит лучше оценить простоту этого каркаса. Мы также поговорим об истории Hadoop и некоторых перспективах парадигмы MapReduce. Но сначала позвольте мне вкратце объяснить, для чего я написал эту книгу и чем она может быть вам полезна.

## 1.1. Зачем написана книга «Hadoop в действии»?

На своем опыте я убедился, что поначалу Hadoop завораживает своими возможностями, но дело с трудом продвигается дальше написания элементарных примеров. Документация на официальном сайте

Hadoop довольна полная, но найти ясные ответы на прямо поставленные вопросы не всегда легко.

Решение данной проблемы и составляет задачу этой книги. Я не стану углубляться в скучные технические подробности, а поделюсь информацией, которая позволит вам быстро создавать полезный код. Попутно я расскажу о более сложных вопросах, часто возникающих на практике.

## 1.2. Что такое Hadoop?

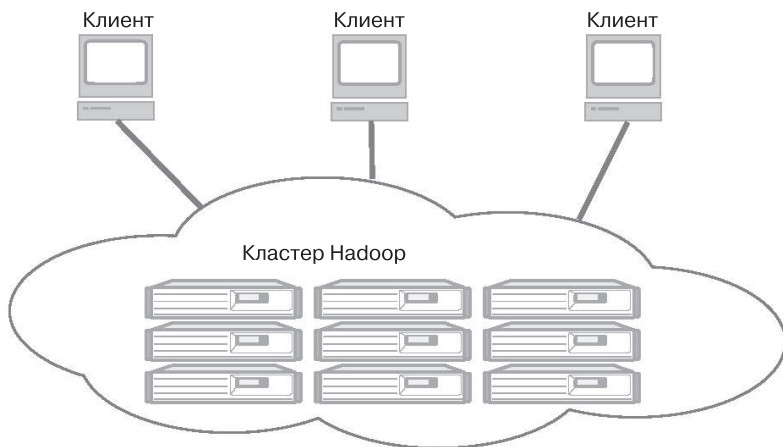
Говоря формально, Hadoop – это каркас с открытым исходным кодом, предназначенный для создания и запуска распределенных приложений, обрабатывающих большие объемы данных. Распределенные вычисления – это широкая и многогранная область, но у Hadoop есть ряд важных отличительных особенностей, а именно:

- ❑ *Доступность* – Hadoop работает на крупных кластерах, собранных из стандартных компьютеров, или в вычислительном облаке, например на базе службы Elastic Compute Cloud (EC2), предлагаемой компанией Amazon.
- ❑ *Надежность* – поскольку Hadoop должен работать на стандартном оборудовании, его архитектура разработана с учетом возможности частых отказов. Большинство отказов можно обработать так, что характеристики кластера будут ухудшаться постепенно.
- ❑ *Масштабируемость* – Hadoop масштабируется линейно, то есть при увеличении объема данных достаточно добавить новые узлы в кластер.
- ❑ *Простота* – Hadoop позволяет пользователю быстро создавать эффективный параллельный код.

Доступность и простота Hadoop дают ему конкурентное преимущество в деле написания и запуска больших распределенных программ. Даже студент колледжа может быстро собрать собственный недорогой кластер Hadoop. С другой стороны, надежность и масштабируемость делают Hadoop подходящим средством даже для таких ответственных задач, которые решаются в компаниях Yahoo и Facebook. Поэтому Hadoop популярен как в академических кругах, так и в промышленности.

На рис. 1.1 показано, как пользователь взаимодействует с кластером Hadoop. Как видите, кластер Hadoop представляет собой на-

бор стандартных компьютеров, объединенных в сеть, физически расположенную в одном месте<sup>2</sup>. Как хранение, так и обработка данных происходят внутри этого «облака» машин. Разные пользователи управляют кластеру Hadoop вычислительные «задания» с клиентских машин, например со своих настольных компьютеров, которые могут находиться далеко от кластера.



**Рис. 1.1.** Кластер Hadoop состоит из многих машин, которые хранят и параллельно обрабатывают большие наборы данных. Клиентские компьютеры посылают в это вычислительное облако задания и получают результаты.

Не все распределенные системы конфигурируются так, как показано на рис. 1.1. Краткое знакомство с другими распределенными системами поможет лучше понять философию Hadoop.

## 1.3. Сравнение Hadoop с другими распределенными системами

Закон Мура верно служил нам несколько десятилетий, но теперь создание все более и более мощных серверов не обязательно дает оптимальное решение крупномасштабных задач. Быстро набирает

<sup>2</sup> Хотя это и необязательно, обычно кластер Hadoop состоит из более-менее однородных машин на базе процессоров x86 под управлением ОС Linux. И почти всегда они находятся в одном центре обработки данных, часто даже в общем наборе стоек.

популярность альтернативный подход – объединение множества недорогих стандартных машин в функционально единую *распределенную систему*.

Чтобы понять, почему распределенные системы (масштабирование по горизонтали) оказались популярнее монолитных серверов (масштабирование по вертикали), надо принять во внимание производительность и цену современных технологий ввода/вывода. Высокпроизводительному компьютеру с четырьмя каналами ввода/вывода, производительностью 100 МБ/с каждый, потребуется три часа, чтобы только *прочитать* набор данных объемом 4 ТБ! Если же воспользоваться Hadoop, то такой же набор данных можно разбить на меньшие блоки (обычно размером 64 МБ), распределенные по нескольким машинам кластера с помощью распределенной файловой системы Hadoop Distributed File System (HDFS). Благодаря довольно скромной репликации кластерные машины могут читать этот набор данных параллельно, достигая гораздо более высокой пропускной способности. И при этом целый кластер, собранный из стандартных машин, оказывается дешевле одного высокопроизводительного сервера!

Сказанное выше объясняет, почему кластер Hadoop оказывается эффективнее монолитных систем. А теперь сравним Hadoop с другими архитектурами распределенных систем. Одним из примеров может служить система SETI@home, в которой хранители экрана, работающие на компьютерах, разбросанных по всему земному шару, совместно решают задачу поиска внеземных цивилизаций. В этой системе имеется центральный сервер, на котором хранятся полученные из космоса радиосигналы, и этот сервер по Интернету рассылает клиентским компьютерам задачи поиска аномальных признаков. При таком подходе данные перемещаются на места выполнения вычислений (хранители экрана в настольных компьютерах). По завершении вычислений результаты возвращаются в систему хранения данных.

Hadoop отличается от систем, подобных SETI@home, взглядом на данные. SETI@home основана на повторяющихся операциях пересылки данных между клиентами и серверами. Это хорошо работает для задач, требующих большого объема вычислений, но не годится, когда нужно обработать много данных, поскольку их объем настолько велик, что перемещение становится неэффективным. В основу Hadoop положена идея приближения кода к данным, а не наоборот. Взгляните еще раз на рис. 1.1 – как видите, хранение данных и выполнение вычислений происходит внутри кластера Hadoop. Клиенты