

TURING 图灵程序设计丛书

[PACKT]
PUBLISHING



最佳Hadoop入门图书，进入大数据处理的绝好途径，不容错过！

Hadoop

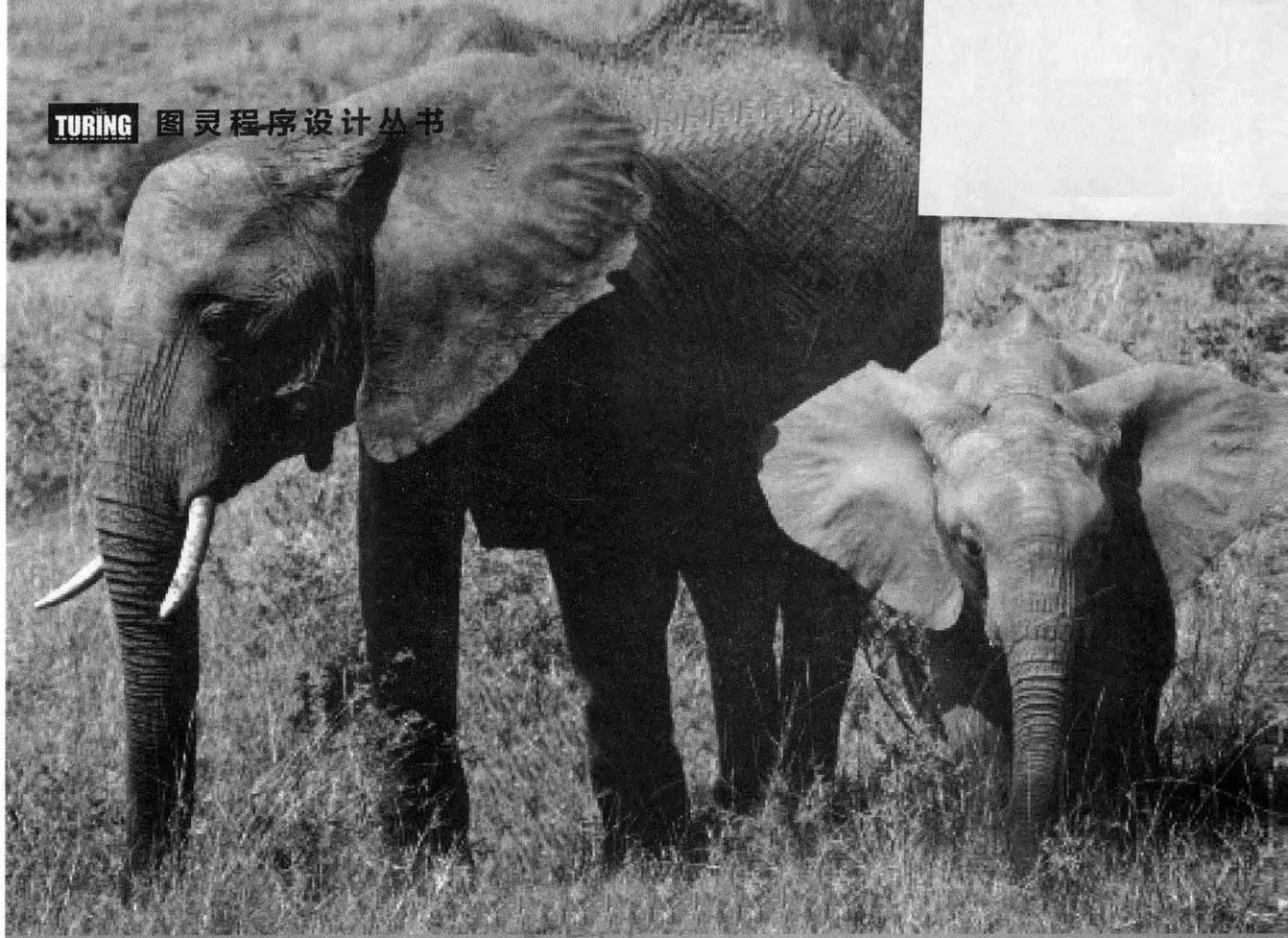
Hadoop Beginner's Guide

基础教程

[英] Garry Turkington 著
张治起 译

人民邮电出版社
POSTS & TELECOM PRESS

TURING 图灵程序设计丛书



最佳Hadoop入门图书，进入大数据处理的绝好途径，不容错过！

Hadoop Hadoop Beginner's Guide 基础教程

[英] Garry Turkington 著
张治起 译

人民邮电出版社
北京

图书在版编目(CIP)数据

Hadoop基础教程 / (英) 特金顿 (Turkington, G.)
著 ; 张治起译. -- 北京 : 人民邮电出版社, 2014. 1
(图灵程序设计丛书)
书名原文: Hadoop beginner's guide
ISBN 978-7-115-34133-4

I. ①H… II. ①特… ②张… III. ①数据处理软件—
教材 IV. ①TP274

中国版本图书馆CIP数据核字(2013)第302148号

内 容 提 要

本书包括三个主要部分:第1~5章讲述了Hadoop的核心机制及Hadoop的工作模式;第6~7章涵盖了Hadoop更多可操作的内容;第8~11章介绍了Hadoop与其他产品和技术的组合使用。本书目的在于帮助读者了解什么是Hadoop, Hadoop是如何工作的, 以及如何使用Hadoop从数据中提取有价值的信息, 并用它解决大数据问题。

本书适用于有软件开发经验的技术人员。



-
- ◆ 著 [英] Garry Turkington
 - 译 张治起
 - 责任编辑 李 瑛
 - 责任印制 焦志炜
 - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京鑫正大印刷有限公司印刷
 - ◆ 开本: 800×1000 1/16
印张: 20.25
字数: 478千字 2014年1月第1版
印数: 1-3 500册 2014年1月北京第1次印刷
著作权合同登记号 图字: 01-2013-4680号

定价: 65.00元

读者服务热线: (010)51095186转600 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京崇工商广字第 0021 号

版权声明

Copyright © 2013 Packt Publishing. First published in the English language under the title *Hadoop Beginner's Guide*.

Simplified Chinese-language edition copyright © 2014 by Posts & Telecom Press. All rights reserved.

本书中文简体字版由Packt Publishing授权人民邮电出版社独家出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

版权所有，侵权必究。

前 言

本书目的在于帮助读者理解Hadoop，并用它解决大数据问题。能使用Hadoop这样的数据处理技术进行工作是令人激动的。对大规模数据集进行复杂分析的能力，曾一度被大公司和政府机构所垄断，而现在通过免费的OSS（open source software，开源软件）即可获得这种能力。

该领域看上去有些复杂，并且变化节奏很快，所以理解这方面的基本知识让人望而生畏。这就是本书诞生的意义所在，它帮助读者了解什么是Hadoop，Hadoop是如何工作的，以及如何使用Hadoop从数据中提取有价值的信息。

除了解释Hadoop的核心机制，本书也会用几章内容来学习其他相关技术，这些技术要么用到了Hadoop，要么需要与Hadoop配套使用。我们的目标是，让读者不仅理解Hadoop是什么，还要理解如何在更宽泛的技术设施中使用Hadoop。

本书中提到的另一种技术是云计算的应用，尤其是AWS（Amazon Web Services，亚马逊网络服务）产品。本书中，我们将展示如何使用这些服务来承载Hadoop工作负载。这就意味着，读者无需购买任何物理硬件，就能处理大规模数据。

本书内容

本书包括3个主要部分：第1~5章讲述了Hadoop的核心机制及Hadoop的工作模式；第6~7章涵盖了Hadoop更多可操作的内容；第8~11章介绍了Hadoop与其他产品和技术的组合使用。每章内容具体如下。

第1章“绪论”。简要介绍了产生Hadoop和云计算的背景。如今看来，这些技术是如此重要。

第2章“安装并运行Hadoop”。指导读者完成本地Hadoop集群的安装，并运行一些示例作业。为了进行对比，在托管于亚马逊服务的Hadoop上执行同样的任务。

第3章“理解MapReduce”。深入研究Hadoop运行原理，揭示了MapReduce作业的执行方式，并展示了如何使用Java API编写MapReduce应用程序。

第4章“开发MapReduce程序”。通过对一个中等规模数据集案例的学习研究，演示如何着手处理和分析新数据源。

第5章“高级MapReduce技术”。介绍一些更复杂的应用MapReduce解决问题的方法，Hadoop似乎并不直接适用于这些问题。

第6章“故障处理”。详细检查Hadoop备受赞誉的高可用性和容错能力，通过强制结束进程和故意使用错误数据等方式故意制造破坏，以检验Hadoop在上述情况下的表现。

第7章“系统运行与维护”。从更具操作性的角度讲解Hadoop，这对于Hadoop集群的管理人员非常有用。本章在介绍一些最佳做法的同时，也描述了如何预防最糟糕的操作性灾难，因此系统管理员可以高枕无忧了。

第8章“Hive：数据的关系视图”。介绍Apache Hive，它允许使用类似SQL的语法对Hadoop数据进行查询。

第9章“与关系数据库协同工作”。探讨Hadoop如何与现有数据库融合，特别是如何将数据从一个数据库移到另一个数据库。

第10章“使用Flume收集数据”。介绍如何使用Apache Flume从多个数据源收集数据，并传送到Hadoop这样的目的地。

第11章“展望未来”。以更广泛的Hadoop生态系统概述结束全书，重点突出其他产品和技术潜在价值。此外，还提供了一些如何参与Hadoop社区并获得帮助的方法。

准备工作

本书每章内容分别介绍该章用到的Hadoop相关软件包。但是，无论哪章内容都要用到运行Hadoop集群的硬件设施。

在最简单的情况下，一台基于Linux的主机可作为运行几乎全书所有示例的平台。任何可运行Linux命令行的先进操作系统都可满足需求，文中假设读者使用的是Ubuntu的最新发布版。

后面几章中的一些例子确实需要在多台机器上运行，所以读者需要拥有至少4台主机的访问权限。虚拟机也是可以的。虽然对于产品来讲，它们并非理想选择，但完全能够满足学习和研究的需要。

本书中，我们还将研究AWS，读者可以在EC2实例上运行所有示例。本书中，我们将更多地着眼于AWS针对Hadoop的具体用途。任何人都可以使用AWS服务，但需要一张信用卡进行注册！

目标读者

我们认为，本书读者想要学习更多关于Hadoop的实践知识。本书的主要受众是，有软件开发

经验却没有接触过Hadoop或类似大数据技术的人员。

对于那些想知道如何编写MapReduce应用的开发者，假设他们能轻松地编写Java程序并熟悉Unix命令行接口。本书还包含几个Ruby程序示例，但这些通常只是说明语言的独立性，并不要求读者成为一位Ruby专家。

对于架构师和系统管理员而言，本书也具有重大价值。它解释了Hadoop的工作原理，Hadoop在更广阔的系统架构中所处的位置，以及如何管理Hadoop集群。这类受众对一些复杂技术可能缺乏直接兴趣，如第4章和第5章。

约定

本书中，有几个经常出现的标题。

为了明确说明如何完成一个程序或任务，本书使用下面的格式详细描述操作步骤。

实践环节：标题

- (1) 操作1
- (2) 操作2
- (3) 操作3

通常，需要一些额外的解释帮助读者理解这些指令，因此紧随其后的是下面的原理分析。

原理分析

这部分内容对任务运行原理或刚完成的指令进行解释说明。

本书还包含一些其他的学习辅助标记，包括：

随堂测验：标题

这是一些简短的多选题，目的在于帮助读者测试对相关内容的理解是否正确。

一展身手：标题

这部分内容设置一些实际问题，并为读者提供一些验证所学内容的方法。

本书使用多种文字风格来区分不同种类的信息。下面是一些例子还有对其意义的解释。

代码块设置如下：你也许注意到，我们使用Unix命令`rm`而不是DOS `del`命令移除Drush目录。

```
# * Fine Tuning
#
key_buffer = 16M
key_buffer_size = 32M
max_allowed_packet = 16M
thread_stack = 512K
thread_cache_size = 8
max_connections = 300
```

如果代码块的特定部分需要特别关注，相应行或内容会加粗显示。

```
# * Fine Tuning
#
key_buffer = 16M
key_buffer_size = 32M
max_allowed_packet = 16M
thread_stack = 512K
thread_cache_size = 8
max_connections = 300
```

命令行的输入或输出会以如下形式显示。

```
cd /ProgramData/Propeople
rm -r Drush
git clone --branch master http://git.drupal.org/project/drush.git
```

新词或者重要的词会以黑体字显示。以菜单或对话框为例，读者在屏幕上看到的内容如下所示“在**Select Destination Location**（选择目的地址）页面，点击**Next**（下一步）按钮接受默认输出地址。”



警告或重要提示会出现在这样的方框里。



小窍门和技巧会以这样的形式出现。

读者反馈

我们非常欢迎来自读者的反馈。将你对本书的看法告诉我们，哪些内容是你喜欢的或哪些是你不喜欢的。读者反馈可以帮助我们不断提高书稿质量，这是非常重要的。

你只需向我们发送一封电子邮件，在邮件主题中提及书名即可反馈你的意见。我们的邮箱是：feedback@packtpub.com。

如果你擅长某个领域并对写作或者创作有兴趣，请参阅www.packtpub.com/authors上的作者指南。

客户支持

既然你购买了Packt的图书，我们将为你提供多种服务，使你本次购买物超所值。

下载示例代码

你可以通过<http://www.packtpub.com>网站的账号下载所购买Packt书籍的示例代码文件。或者登录网站<http://www.packtpub.com/support>进行注册，随后我们将通过电子邮件向你发送本书配套的示例代码文件。

勘误表

虽然作者尽最大努力保证本书内容的准确性，然而错误在所难免。如果你在Packt出版的书中发现了文字错误或是代码错误，衷心希望你能及时向我们反馈，并预致谢意。你的意见可以帮助我们改进图书的后续版本，并能帮助其他读者避免错误。请通过以下步骤提交你发现的错误之处：访问网站<http://www.packtpub.com/submit-errata>，选择相应图书，点击提交勘误表，登记你的勘误表详情。勘误表通过验证之后将上传到Packt网站，或添加到已有的勘误列表中。

关于盗版

互联网上的盗版问题由来已久，而且各种形式的媒介都存在这个问题。Packt出版社非常重视版权保护问题。如果你在互联网上看到了Packt图书的任何非法拷贝，无论它以何种形式存在，请立即向我们提供其位置或网站名称以便及时补救。

请通过邮箱copyright@packtpub.com与我们联系，并附上涉嫌盗版材料的链接。

你的帮助保障了Packt图书作者的权益，也让我们能持续提供高品质图书。

问题

如果你有关于本书任何方面的问题，可以通过questions@packtpub.com联系我们。我们将会尽力解决。

目 录

第 1 章 绪论	1	2.13.1 使用 EMR 的其他方式	41
1.1 大数据处理	1	2.13.2 AWS 生态系统	42
1.1.1 数据的价值	2	2.14 本地 Hadoop 与 EMR Hadoop 的对比	42
1.1.2 受众较少	2	2.15 小结	43
1.1.3 一种不同的方法	4	第 3 章 理解 MapReduce	44
1.1.4 Hadoop	7	3.1 键值对	44
1.2 基于 Amazon Web Services 的云计算	12	3.1.1 具体含义	44
1.2.1 云太多了	12	3.1.2 为什么采用键/值数据	45
1.2.2 第三种方法	12	3.1.3 MapReduce 作为一系列键/值 变换	46
1.2.3 不同类型的成本	12	3.2 MapReduce 的 Hadoop Java API	47
1.2.4 AWS: Amazon 的弹性架构	13	3.3 编写 MapReduce 程序	50
1.2.5 本书内容	14	3.4 实践环节: 设置 classpath	50
1.3 小结	15	3.5 实践环节: 实现 WordCount	51
第 2 章 安装并运行 Hadoop	16	3.6 实践环节: 构建 JAR 文件	53
2.1 基于本地 Ubuntu 主机的 Hadoop 系统	16	3.7 实践环节: 在本地 Hadoop 集群运行 WordCount	54
2.2 实践环节: 检查是否已安装 JDK	17	3.8 实践环节: 在 EMR 上运行 WordCount	54
2.3 实践环节: 下载 Hadoop	18	3.8.1 0.20 之前版本的 Java MapReduce API	56
2.4 实践环节: 安装 SSH	19	3.8.2 Hadoop 提供的 mapper 和 reducer 实现	57
2.5 实践环节: 使用 Hadoop 计算圆周率	20	3.9 实践环节: WordCount 的简易方法	58
2.6 实践环节: 配置伪分布式模式	22	3.10 查看 WordCount 的运行全貌	59
2.7 实践环节: 修改 HDFS 的根目录	24	3.10.1 启动	59
2.8 实践环节: 格式化 NameNode	25	3.10.2 将输入分块	59
2.9 实践环节: 启动 Hadoop	26	3.10.3 任务分配	60
2.10 实践环节: 使用 HDFS	27	3.10.4 任务启动	60
2.11 实践环节: MapReduce 的经典入门 程序——字数统计	28	3.10.5 不断监视 JobTracker	60
2.12 使用弹性 MapReduce	33	3.10.6 mapper 的输入	61
2.13 实践环节: 使用管理控制台在 EMR 运行 WordCount	34		

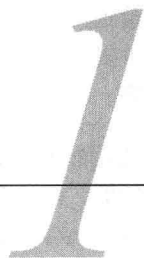
3.10.7 mapper 的执行	61	4.4 实践环节: 统计汇总 UFO 数据	80
3.10.8 mapper 的输出和 reducer 的 输入	61	4.5 实践环节: 统计形状数据	82
3.10.9 分块	62	4.6 实践环节: 找出目击事件的持续时间 与 UFO 形状的关系	84
3.10.10 可选分块函数	62	4.7 实践环节: 在命令行中执行形状/时间 分析	87
3.10.11 reducer 类的输入	62	4.8 实践环节: 使用 ChainMapper 进行字 段验证/分析	88
3.10.12 reducer 类的执行	63	4.9 实践环节: 使用 Distributed Cache 改 进地点输出	93
3.10.13 reducer 类的输出	63	4.10 计数器、状态和其他输出	96
3.10.14 关机	63	4.11 实践环节: 创建计数器、任务状态 和写入日志	96
3.10.15 这就是 MapReduce 的全部	64	4.12 小结	102
3.10.16 也许缺了 combiner	64		
3.11 实践环节: 使用 combiner 编写 Word Count	64	第 5 章 高级 MapReduce 技术	103
3.12 实践环节: 更正使用 combiner 的 WordCount	65	5.1 初级、高级还是中级	103
3.13 Hadoop 专有数据类型	67	5.2 多数据源联结	103
3.13.1 Writable 和 Writable- Comparable 接口	67	5.2.1 不适合执行联结操作的情况	104
3.13.2 wrapper 类介绍	68	5.2.2 map 端联结与 reduce 端联结的 对比	104
3.14 实践环节: 使用 Writable 包装类	69	5.2.3 匹配账户与销售信息	105
3.15 输入/输出	71	5.3 实践环节: 使用 MultipleInputs 实现 reduce 端联结	105
3.15.1 文件、split 和记录	71	5.3.1 实现 map 端联结	109
3.15.2 InputFormat 和 RecordReader	71	5.3.2 是否进行联结	112
3.15.3 Hadoop 提供的 InputFormat	72	5.4 图算法	112
3.15.4 Hadoop 提供的 RecordReader	73	5.4.1 Graph 101	112
3.15.5 OutputFormat 和 Record-Writer	73	5.4.2 图和 MapReduce	112
3.15.6 Hadoop 提供的 OutputFormat	73	5.4.3 图的表示方法	113
3.15.7 别忘了 Sequence files	74	5.5 实践环节: 图的表示	114
3.16 小结	74	5.6 实践环节: 创建源代码	115
第 4 章 开发 MapReduce 程序	75	5.7 实践环节: 第一次运行作业	119
4.1 使用非 Java 语言操作 Hadoop	75	5.8 实践环节: 第二次运行作业	120
4.1.1 Hadoop Streaming 工作原理	76	5.9 实践环节: 第三次运行作业	121
4.1.2 使用 Hadoop Streaming 的 原因	76	5.10 实践环节: 第四次也是最后一次运行 作业	122
4.2 实践环节: 使用 Streaming 实现 Word- Count	76	5.10.1 运行多个作业	124
4.3 分析大数据集	79	5.10.2 关于图的终极思考	124
4.3.1 获取 UFO 目击事件数据集	79	5.11 使用语言无关的数据结构	124
4.3.2 了解数据集	80		

5.11.1 候选技术	124	7.3.2 默认存储位置	172
5.11.2 Avro 简介	125	7.3.3 设置 Hadoop 属性的几种方式	173
5.12 实践环节: 获取并安装 Avro	125	7.4 集群设置	174
5.13 实践环节: 定义模式	126	7.4.1 为集群配备多少台主机	174
5.14 实践环节: 使用 Ruby 创建 Avro 源 数据	127	7.4.2 特殊节点的需求	176
5.15 实践环节: 使用 Java 语言编程操作 Avro 数据	128	7.4.3 不同类型的存储系统	177
5.16 实践环节: 在 MapReduce 中统计 UFO 形状	130	7.4.4 Hadoop 的网络配置	178
5.17 实践环节: 使用 Ruby 检查输出 数据	134	7.5 实践环节: 查看默认的机柜配置	180
5.18 实践环节: 使用 Java 检查输出 数据	135	7.6 实践环节: 报告每台主机所在机柜	180
5.19 小结	137	7.7 集群访问控制	183
第 6 章 故障处理	138	7.8 实践环节: 展示 Hadoop 的默认安全 机制	183
6.1 故障	138	7.9 管理 NameNode	187
6.1.1 拥抱故障	138	7.10 实践环节: 为 fsimage 文件新增一 个存储路径	188
6.1.2 至少不怕出现故障	139	7.11 实践环节: 迁移到新的 NameNode 主机	190
6.1.3 严禁模仿	139	7.12 管理 HDFS	192
6.1.4 故障类型	139	7.12.1 数据写入位置	192
6.1.5 Hadoop 节点故障	139	7.12.2 使用平衡器	193
6.2 实践环节: 杀死 DataNode 进程	141	7.13 MapReduce 管理	193
6.3 实践环节: 复制因子的作用	144	7.13.1 通过命令行管理作业	193
6.4 实践环节: 故意造成数据块丢失	146	7.13.2 作业优先级和作业调度	194
6.5 实践环节: 杀死 TaskTracker 进程	149	7.14 实践环节: 修改作业优先级并结束 作业运行	194
6.6 实践环节: 杀死 JobTracker	153	7.15 扩展集群规模	197
6.7 实践环节: 杀死 NameNode 进程	154	7.15.1 提升本地 Hadoop 集群的计 算能力	197
6.8 实践环节: 引发任务故障	160	7.15.2 提升 EMR 作业流的计算能 力	198
6.9 数据原因造成的任务故障	163	7.16 小结	198
6.10 实践环节: 使用 skip 模式处理异常 数据	164	第 8 章 Hive: 数据的关系视图	200
6.11 小结	169	8.1 Hive 概述	200
第 7 章 系统运行与维护	170	8.1.1 为什么使用 Hive	200
7.1 关于 EMR 的说明	170	8.1.2 感谢 Facebook	201
7.2 Hadoop 配置属性	171	8.2 设置 Hive	201
7.3 实践环节: 浏览默认属性	171	8.2.1 准备工作	201
7.3.1 附加的属性元素	172	8.2.2 下载 Hive	202
		8.3 实践环节: 安装 Hive	202

8.4	使用 Hive	203	9.6.1	使用 MySQL 工具手工导入	246
8.5	实践环节：创建 UFO 数据表	204	9.6.2	在 mapper 中访问数据库	246
8.6	实践环节：在表中插入数据	206	9.6.3	更好的方法：使用 Sqoop	247
8.7	实践环节：验证表	208	9.7	实践环节：下载并配置 Sqoop	247
8.8	实践环节：用正确的列分隔符重定义表	210	9.8	实践环节：把 MySQL 的数据导入 HDFS	249
8.9	实践环节：基于现有文件创建表	212	9.9	实践环节：把 MySQL 数据导出到 Hive	253
8.10	实践环节：执行联结操作	214	9.10	实践环节：有选择性的导入数据	255
8.11	实践环节：使用视图	216	9.11	实践环节：使用数据类型映射	257
8.12	实践环节：导出查询结果	219	9.12	实践环节：通过原始查询导入数据	258
8.13	实践环节：制作 UFO 目击事件分区表	221	9.13	从 Hadoop 导出数据	261
8.13.1	分桶、归并和排序	224	9.13.1	在 reducer 中把数据写入关系数据库	261
8.13.2	用户自定义函数	225	9.13.2	利用 reducer 输出 SQL 数据文件	262
8.14	实践环节：新增用户自定义函数	225	9.13.3	仍是最好的方法	262
8.14.1	是否进行预处理	228	9.14	实践环节：把 Hadoop 数据导入 MySQL	262
8.14.2	Hive 和 Pig 的对比	229	9.15	实践环节：把 Hive 数据导入 MySQL	265
8.14.3	未提到的内容	229	9.16	实践环节：改进 mapper 并重新运行数据导出命令	267
8.15	基于 Amazon Web Services 的 Hive	230	9.17	在 AWS 上使用 Sqoop	269
8.16	实践环节：在 EMR 上分析 UFO 数据	230	9.18	小结	270
8.16.1	在开发过程中使用交互式作业流	235			
8.16.2	与其他 AWS 产品的集成	236			
8.17	小结	236			
第 9 章	与关系数据库协同工作	238	第 10 章	使用 Flume 收集数据	271
9.1	常见数据路径	238	10.1	关于 AWS 的说明	271
9.1.1	Hadoop 用于存储档案	238	10.2	无处不在的数据	271
9.1.2	使用 Hadoop 进行数据预处理	239	10.2.1	数据类别	272
9.1.3	使用 Hadoop 作为数据输入工具	239	10.2.2	把网络流量导入 Hadoop	272
9.1.4	数据循环	240	10.3	实践环节：把网络服务器数据导入 Hadoop	272
9.2	配置 MySQL	240	10.3.1	把文件导入 Hadoop	273
9.3	实践环节：安装并设置 MySQL	240	10.3.2	潜在的问题	273
9.4	实践环节：配置 MySQL 允许远程连接	243	10.4	Apache Flume 简介	274
9.5	实践环节：建立员工数据库	245	10.5	实践环节：安装并配置 Flume	275
9.6	把数据导入 Hadoop	246	10.6	实践环节：把网络流量存入日志文件	277
			10.7	实践环节：把日志输出到控制台	279

10.8	实践环节：把命令的执行结果写入平面文件	281	11.2	即将到来的 Hadoop 变革	300
10.9	实践环节：把远程文件数据写入本地平面文件	283	11.3	其他版本的 Hadoop 软件包	300
10.9.1	信源、信宿和信道	284	11.4	其他 Apache 项目	303
10.9.2	Flume 配置文件	286	11.4.1	HBase	303
10.9.3	一切都以事件为核心	287	11.4.2	Oozie	303
10.10	实践环节：把网络数据写入 HDFS	287	11.4.3	Whir	304
10.11	实践环节：加入时间戳	289	11.4.4	Mahout	304
10.12	实践环节：多层 Flume 网络	292	11.4.5	MRUnit	305
10.13	实践环节：把事件写入多个信宿	294	11.5	其他程序设计模式	305
10.13.1	选择器的类型	295	11.5.1	Pig	305
10.13.2	信宿故障处理	295	11.5.2	Cascading	305
10.13.3	使用简单元件搭建复杂系统	296	11.6	AWS 资源	306
10.14	更高的视角	297	11.6.1	在 EMR 上使用 HBase	306
10.14.1	数据的生命周期	297	11.6.2	SimpleDB	306
10.14.2	集结数据	297	11.6.3	DynamoDB	306
10.14.3	调度	297	11.7	获取信息的渠道	307
10.15	小结	298	11.7.1	源代码	307
第 11 章	展望未来	299	11.7.2	邮件列表和论坛	307
11.1	全书回顾	299	11.7.3	LinkedIn 群组	307
			11.7.4	Hadoop 用户群	307
			11.7.5	会议	308
			11.8	小结	308
			随堂测验答案	309	

绪 论



本书介绍一种大规模数据处理的开源框架——Hadoop。在深入探讨它的技术细节和应用之前，很有必要花点时间来了解Hadoop及其取得巨大成功的历史背景。

Hadoop并不是凭空想象出来的，它的出现源于人们创建和使用的数据量的爆炸性增长。在此背景下，不仅是庞大的跨国公司面临着海量数据处理的困难，小型创业公司同样如此。与此同时，其他历史变革改变了软件和系统的部署方式，除了传统的基础设施，人们开始使用甚至偏好于云资源。

本章将要探讨Hadoop出现的背景，并详细讲解Hadoop想要解决的问题和决定其最终设计的内在驱动因素。

本章包括以下内容：

- 概述大数据革命；
- 讲解Hadoop是什么以及如何从数据中获取有价值信息；
- 探秘云计算并了解AWS（Amazon Web Services，亚马逊网络服务）的功能；
- 了解大数据处理技术与云计算相结合带来的巨大威力；
- 概述本书其余章节内容。

现在 we 开始吧！

1.1 大数据处理

审视现有技术，不难发现，所有技术都以数据为核心。作为用户，我们对富媒体的欲望与日俱增，比如观看的电影和创建并上传到网络的照片和视频。我们也常常在日常生活中，不经意地在网上留下一串数据。

不仅是数据总量在迅速增加,同时数据生成速率也在不断增加。从电子邮件到Facebook留言,从网上购物记录到网站链接,到处都是不断增长的大数据集。在此背景下,最大的挑战在于,如何从这些数据中提取最有价值的信息。有时是提取特定的数据元素,有时是分析数据间的关系或是判断一种趋势。

微妙的变化悄然发生,数据的使用方式变得越来越有意义。一段时间以来,大型公司已经意识到了数据的价值,并用它来提升服务质量。例如,Google在用户正在访问的网页上显示内容相关的广告,Amazon或Netflix向用户推荐合乎其口味和兴趣的新产品或新电影。

1.1.1 数据的价值

如果不会带来有价值的回报或者明显的竞争优势,这些大型企业是不会投资发展大数据处理技术的。应当从以下几个方面来认识大数据。

- 只有在数据集足够大的时候,某些问题才变得有意义。例如,在其他影响因素缺失的情况下,基于一个第三人的喜好推荐电影是不可能非常精准的。然而,当参考样本增加到100时,推荐成功的几率略有上升。而使用1000万人的观看记录,可以大幅提升获得推荐模型的可能性。
- 与之前的解决方案相比,大数据处理工具通常能够以较低的成本处理更大规模的数据。因此,能够在可接受的成本范围内执行以往花费极高的数据处理任务。
- 大规模数据处理的成本不仅体现在财务费用上,等待时间也是一个重要因素。如果一个系统能够处理所有到达数据,但是其平均处理时间以周为计量单位,那么该系统也是不可用的。大数据处理工具通过将增加的数据量分配到附加的硬件,来保证即使在数据量增加的情况下,处理时间也不会失控。
- 为了满足数据库中最大数据的需要,可能需要重新审视之前关于数据库形式或者其中数据结构的假设。
- 综合上述几点内容,足够大的数据集以及灵活的工具可以使之前无法想象的问题得到解答。

1.1.2 受众较少

前面讨论的从大数据中提取有价值信息用于改进服务质量的例子,往往属于大型搜索引擎和在线公司的创新模式。这是因为在早期发展过程中,大数据处理不仅成本高而且实现困难,超出了中小企业的力量范围。

同样,比大数据处理技术应用更为广泛的数据挖掘方法已经存在了很长一段时间,但是在大型企业和政府部门之外却从来没有真正得到推广使用。

这种情况的出现可能令人感到遗憾。但在过去，对于大多数小公司来讲却无关紧要，因为它们的数据量很少，并不需要投入大量的资金来处理这些数据。

然而，现如今数据量的增加不再局限于大型组织，许多中小型公司甚至一些个人收集到的数据也越来越多。他们也意识到这些数据中可能包含着正待发掘的价值。

在理解如何实现这一目标之前，很有必要了解奠定Hadoop系统基础的背景情况。

1. 经典的数据处理系统

造成大数据挖掘系统稀有且昂贵的根本原因是，将现有小型计算系统扩展为大数据处理系统是非常困难的。正如我们所见，一直以来，数据处理系统的处理能力一直受限于单台计算机的极限运算能力。

随着数据规模的增长，出现了两种常用的扩展系统的方法，通常称之为“向上扩展”和“向外扩展”。

● 向上扩展

在大多数企业，数据处理任务通常由相当昂贵的大型机来执行。随着数据规模的增长，向上扩展的方法就是将数据处理任务迁移到更大的服务器或者存储矩阵。即便以今天的视角来看，这种架构确实有效，但其硬件成本会很容易达到几十万甚至几百万美元。本章的后续内容对此有所涉及。

简单的向上扩展系统的优点是，系统的架构并不会随着数据量的增大而发生显著变化。尽管采用了更大型的部件，但部件之间的基本关系（例如数据服务器和存储矩阵）却依然保持一致。对于像商业数据库引擎这样的应用而言，可以通过软件来处理使用硬件带来的复杂性。但是从理论上讲，数据处理能力是通过把相同的软件迁移到规模越来越大的服务器上来实现的。需要注意的是，把软件迁移到越来越多的处理器上也存在一定困难。同时，单台计算机的处理能力也受到现实条件的约束。因此，向上扩展的系统达到其极限后，无法进一步扩展数据处理的规模。

单一架构的数据处理器的规模不可能无限扩大。从概念上来讲，通过向上扩展系统的方式设计一个处理能力为1 TB、100 TB到1 PB数据的系统，可能仅需使用更大规模的相同部件。但随着数据规模的增长，这些定制硬件之间连接的复杂性可能与之前的廉价部件有所不同。

● 早期向外扩展的方法

向外扩展的方法并不通过升级系统的硬件来获得更强的处理能力，而是将数据处理任务分发给越来越多的机器。如果数据集的规模翻倍了，那就使用两台机器来处理，而不是一台有着两倍处理能力的机器。如果数据规模继续翻倍，那就使用四台机器来处理。

与向上扩展的方法相比，向外扩展系统的明显优势在于采购成本大大低于前者。大型机的硬