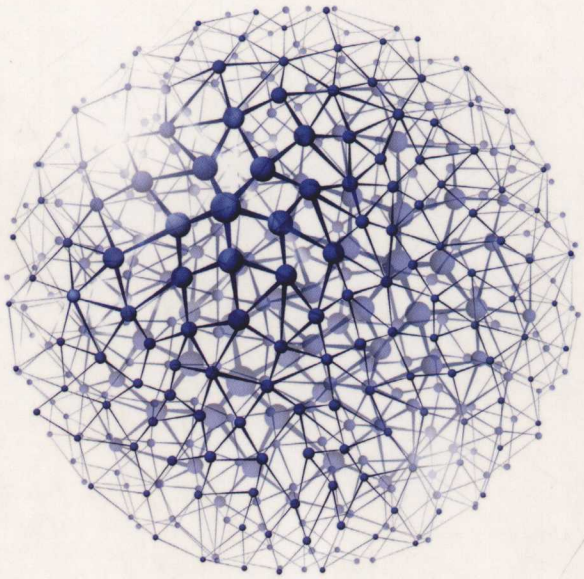
数据中国“百校工程”项目系列教材
数据科学与大数据技术专业系列规划教材

瑞翼教育

Hadoop
大数据技术与应用

杨治明 许桂秋 ◉ 主编
李海涛 杨馥如 杨汉波 高广银 丁勇 刘前 ◉ 副主编



BIG DATA
Technology

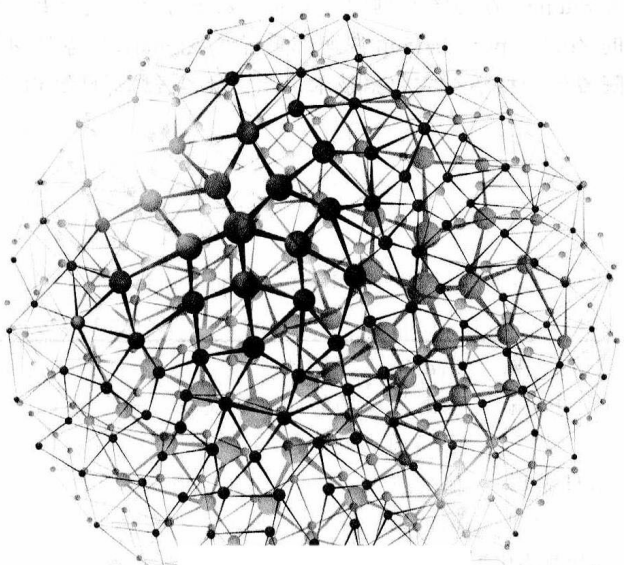
中国工信出版集团
人民邮电出版社
POSTS & TELECOM PRESS

数据中国“百校工程”项目系列教材
数据科学与大数据技术专业系列规划教材

瑞翼教育

Hadoop
大数据技术与应用

杨治明 许桂秋◉主编
李海涛 杨馥如 杨汉波 高广银 丁勇 刘前◉副主编



BIG DATA
Technology

人 民 邮 电 出 版 社
北 京

图书在版编目（CIP）数据

Hadoop大数据技术与应用 / 杨治明，许桂秋主编
. -- 北京 : 人民邮电出版社，2019.3
数据科学与大数据技术专业系列规划教材
ISBN 978-7-115-50353-4

Ⅰ. ①H… Ⅱ. ①杨… ②许… Ⅲ. ①数据处理软件－教材 Ⅳ. ①TP274

中国版本图书馆CIP数据核字(2019)第024411号

内 容 提 要

本书采用理论与实践相结合的方式，全面介绍了 Hadoop 大数据技术。主要内容包括初识 Hadoop 大数据技术，Hadoop 环境设置，分布式文件系统 HDFS，资源调度框架 YARN，分布式并行编程模型 MapReduce，分布式的列式数据库 HBase，数据仓库 Hive，数据查询与分析平台 Pig，分布式的海量日志采集、聚合和传输系统 Flume，在传统数据库与分布式数据库之间进行数据传递的工具 Sqoop，提供分布式协调一致性服务的 ZooKeeper，Hadoop 快速部署工具 Ambari，机器学习领域经典算法库 Mahout。

本书可以作为高等院校数据科学与大数据技术、计算机、信息管理等相关专业的大数据入门教材。

◆ 主　　编　杨治明　许桂秋
副 主 编　李海涛　杨馥如　杨汉波　高广银
　　丁　勇　刘　前
责任编辑　邹文波
责任印制　陈　犇

◆ 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
固安县铭成印刷有限公司印刷

◆ 开本：787×1092　1/16
印张：18.5　2019 年 3 月第 1 版
字数：486 千字　2019 年 3 月河北第 1 次印刷

定价：55.00 元

读者服务热线：(010)81055256　印装质量热线：(010)81055316
反盗版热线：(010)81055315
广告经营许可证：京东工商广登字 20170147 号

前　言

放眼全球，信息技术已经改变了世界的面貌。信息技术的高速发展，引发了近几年的大数据和人工智能浪潮。目前，整个社会都在关注大数据技术的发展。然而，多数人还是只闻其声，不知其实。信息技术人员作为时代的“弄潮儿”，在对这些波澜壮阔的景象感到兴奋的同时，又深刻感受到技术的飞速变化所带来的巨大压力。

大数据技术是信息技术几十年发展和积累催生的产物，它的技术体系也是在信息技术的积淀上发展而来的。

Hadoop 是当前热门的大数据处理与分析平台。本书作为 Hadoop 的入门教材，采用理论与实践相结合的方式全面介绍 Hadoop 技术。

全书共 11 章，主要介绍 Hadoop 技术的核心 HDFS 和 MapReduce，以及生态圈里的其他组件。

第 1 章介绍 Hadoop 产生的背景、Hadoop 生态圈的组成，让读者初步了解 Hadoop。

第 2 章详细介绍在个人计算机上搭建 Hadoop 环境的步骤，确保读者能够正确设置 Hadoop 环境，为后面的学习做好准备。

第 3 章着重阐述 Hadoop 的核心技术之一，整个生态圈的基石——HDFS，以及 HDFS 的 3 种访问方式。

第 4 章详细介绍 Hadoop 生态圈的资源调度框架——YARN。

第 5 章深入讲解离线计算框架，也是 Hadoop 的核心技术之一——MapReduce，通过多个案例让读者深入理解并掌握 MapReduce 的编程模型。

第 6 章主要讲解数据存储工具——分布式的列式数据库 HBase、数据仓库 Hive，以及数据查询与分析平台 Pig。

第 7 章介绍日志采集、聚合和传输系统 Flume。

第 8 章介绍用于在分布式数据库与传统数据库之间进行数据传递的工具 Sqoop。

第 9 章介绍提供分布式协调一致性服务的 ZooKeeper。

第 10 章介绍 Hadoop 快速部署工具 Ambari。

第 11 章介绍机器学习领域经典算法库 Mahout。

本书可以作为高等院校数据科学与大数据技术、计算机、信息管理等相关专业的大数据入门教材。建议安排 64 课时，教师可根据学生的接受能力以及学校的培养方案选择教学内容。

由于编者水平有限，编写时间仓促，书中难免存在一些疏漏和不足之处，敬请广大读者批评指正。

特别提示：由于各种软件在不断升级，读者打开的软件下载界面以及看到的可下载软件的版本可能与本书中的相关内容不一致，但下载与安装方法是类似的。

本书资源下载网址：www.ryjiaoyu.com。

编　者

2019 年 1 月

目 录

第 1 章 初识 Hadoop 大数据技术

本章主要介绍大数据产生的时代背景，给出了大数据的概念、特征，还介绍了大数据相关问题的解决方案、Hadoop 大数据技术以及 Hadoop 的应用案例。

本章的主要内容如下。

（1）大数据技术概述。

（2）Google 的三篇论文及其思想。

（3）Hadoop 概述。

（4）Hadoop 生态圈。

（5）Hadoop 的典型应用场景和应用架构。

1.1 大数据技术概述

1.1.1 大数据产生的背景

1946 年，计算机诞生，当时的数据与应用紧密捆绑在文件中，彼此不分。19 世纪 60 年代，IT 系统规模和复杂度变大，数据与应用分离的需求开始产生，数据库技术开始萌芽并蓬勃发展，并在 1990 年后逐步统一到以关系型数据库为主导，具体发展阶段如图 1-1 所示。

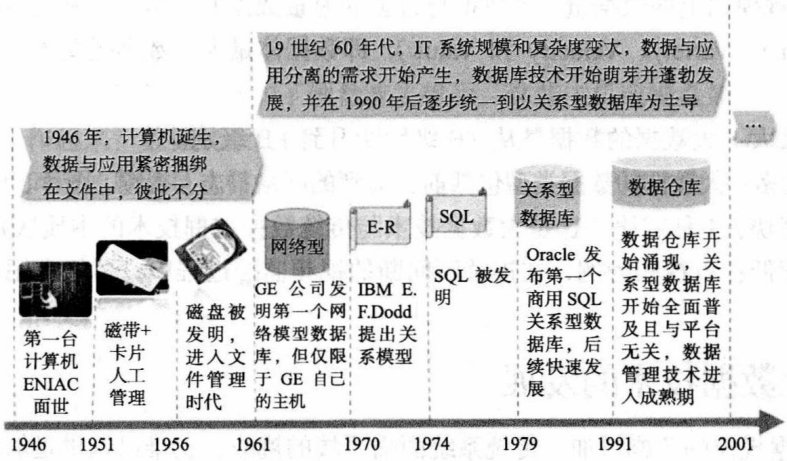


图 1-1 数据管理技术在 2001 年前的两个发展阶段

2001 年后，互联网迅速发展，数据量成倍递增。据统计，目前，超过 150 亿个设备连接到互联网，全球每秒钟发送 290 万封电子邮件，每天有 2.88 万小时视频上传到 YouTube 网站，Facebook 网站每日评论达 32 亿条，每天上传照片近 3 亿张，每月处理数据总量约 130 万 TB。2016 年全球产生数据量 16.1ZB，预计 2020 年将增长到 35ZB（1ZB = 1 百万 PB，PB = 10 亿 TB），如图 1-2 所示。

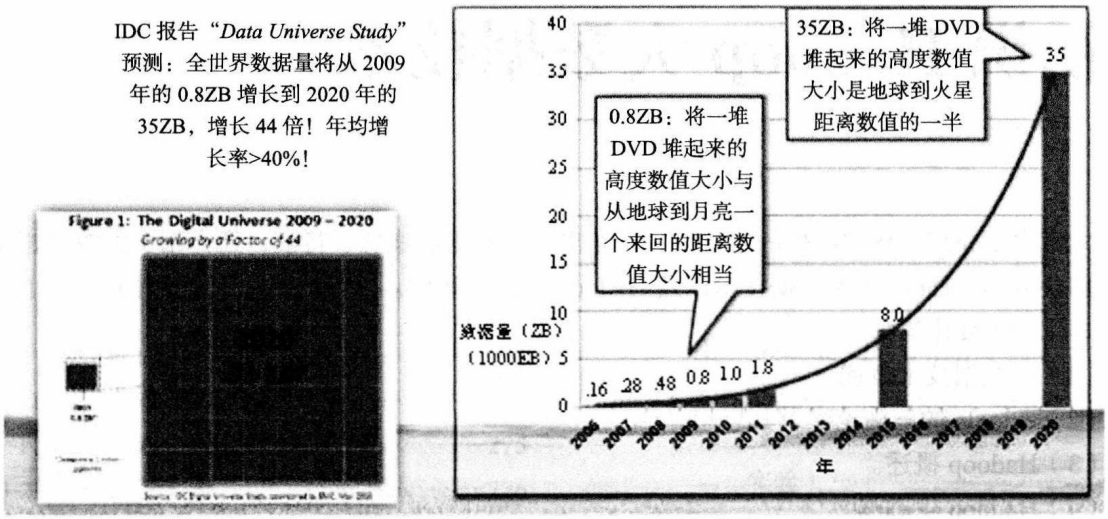


图 1-2　IDC 数据量增长预测报告

2011 年 5 月，EMC World 2011 大会主题是“云计算相遇大数据”，会议除了聚焦 EMC 公司一直倡导的云计算概念外，还抛出了“大数据”（BigData）的概念。2011 年 6 月底，IBM、麦肯锡等众多国外机构发布“大数据”相关研究报告，并予以积极的跟进。

1.1.2　大数据的定义

“大数据”是一个涵盖多种技术的概念，简单地说，是指无法在一定时间内用常规软件工具对其内容进行抓取、管理和处理的数据集合。IBM 公司将“大数据”理念定义为 4 个 V，即大量化（Volume）、多样化（Variety）、快速化（Velocity）及由此产生的价值（Value）。

要理解大数据这一概念，首先要从“大”入手。“大”是指数据规模，大数据一般指在 10TB（1TB=1024GB）规模以上的数据量。大数据与过去的海量数据有所区别，其基本特征可以用 4 个 V 来总结（Volume、Variety、Velocity 和 Value），即数据体量大、数据类型多、处理速度快、价值密度低。

- 数据体量大：大数据的数据量从 TB 级别跃升到 PB 级别。
- 数据类型多：大数据的数据类型包括前文提到的网络日志、视频、图片、地理位置信息等。
- 处理速度快：1 秒定律。这是大数据技术与传统数据挖掘技术的本质区别。
- 价值密度低：以视频为例，在连续不间断的视频监控过程中，可能有用的数据仅仅有一两秒。

1.1.3　大数据技术的发展

随着应用数据规模的急剧增加，传统系统面临严峻的挑战，它难以提供足够的存储和计算资源进行处理。大数据技术是从各种类型的海量数据中快速获得有价值信息的技术。大数据技术要

面对的基本问题，也是最核心的问题，就是海量数据如何可靠存储和如何高效计算的问题。

围绕大数据的核心问题，下面列出了大数据相关技术的发展历程。

2003 年，Google 公司发表了论文“*The Google File System*”，介绍 GFS 分布式文件系统，主要讲解海量数据的可靠存储方法。

2004 年，Google 公司发表了论文“*MapReduce: Simplified Data Processing on Large Clusters*”，介绍并行计算模型 MapReduce，主要讲解海量数据的高效计算方法。

2006 年，Google 公司发表了“*Bigtable: A Distributed Storage System for Structured Data*”，介绍 Google 大表（BigTable）的设计。BigTable 是 Google 公司的分布式数据存储系统，是用来处理海量数据的一种非关系型数据库。

Google 公司根据 GFS、MapReduce 论文思想先后实现了 Hadoop 的 HDFS 分布式文件系统、MapReduce 分布式计算模型并开源。2008 年，Hadoop 成为 Apache 基金会顶级项目。

2010 年，Google 公司根据 BigTable 论文思想，开发出 Hadoop 的 HBase 并开源。开源组织 GNU 发布 MongoDB，VMware 公司提供开源产品 Redis。

2011 年，Twitter 公司提供开源产品 Storm，它是开源的分布式实时计算系统。

2014 年，Spark 成为 Apache 基金会的顶级项目，它是专为大规模数据处理而设计的快速通用的计算引擎。

1.2　Google 的“三驾马车”

Google 公司的三篇论文：GFS、MapReduce、BigTable，奠定了大数据技术的基石，具有划时代的意义，被称为 Google 公司的“三驾马车”。下面分别介绍这三篇论文的思想。

1.2.1　GFS 的思想

论文“*The Google File System*”描述了一个分布式文件系统的设计思路。从交互实体上划分，分布式文件系统有两个基本组成部分，一个是客户端（Client），一个是服务端（Server）。

先考虑第一个问题，如果客户端把文件上传到服务端，但是服务端的硬盘不够大，怎么办？显然，我们可以多加硬盘，或多增加主机。另一个问题，则是数据的存储可靠性怎么保证？如果把文件存在硬盘上，一旦硬盘坏了，数据岂不是丢失了？对于这个问题，可以采用数据冗余存储的方式解决，即同一文件多保存几份。

而事实上事情没那么简单。多增加了硬盘或主机后，这些主机或硬盘如何被管理起来，或它们怎样才能有效运作起来？数据冗余是对每个上传的文件在各台主机都单独存放一份吗？

GFS 解决这些问题的思路是这样的，增加一个管理节点，去管理这些存放数据的主机。存放数据的主机称为数据节点。而上传的文件会按固定的大小进行分块。数据节点上保存的是数据块，而非独立的文件。数据块冗余度默认是 3。上传文件时，客户端先连接管理节点，管理节点生成数据块的信息，包括文件名、文件大小、上传时间、数据块的位置信息等。这些信息称为文件的元信息，它会保存在管理节点。客户端获取这些元信息之后，就开始把数据块一个个上传。客户端把数据块先上传到第一个数据节点，然后，在管理节点的管理下，通过水平复制，复制几份数据块到其他节点，最终达到冗余度的要求。水平复制需要考虑两个要求：可靠性、可用性。分布式文件系统如图 1-3 所示。

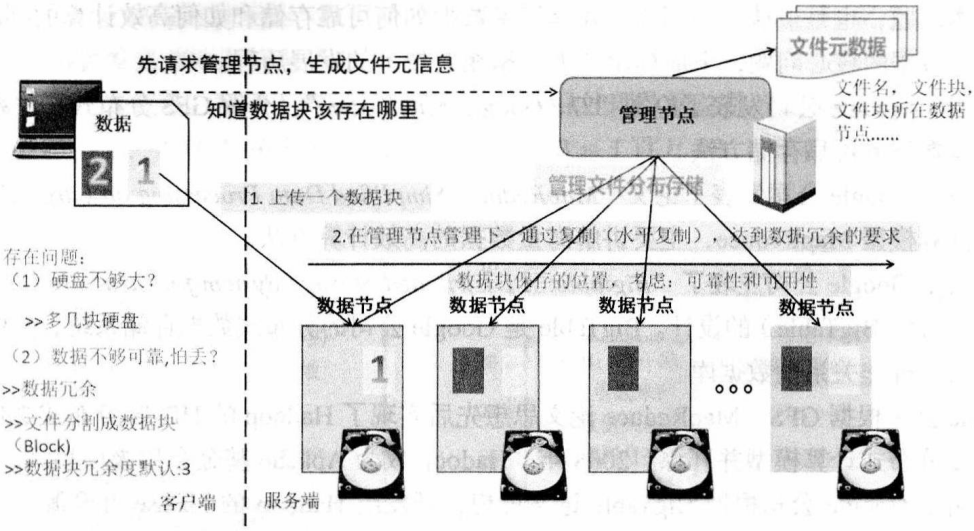


图 1-3　分布式文件系统

论文“*The Google File System*”描述的 GFS 架构如图 1-4 所示。

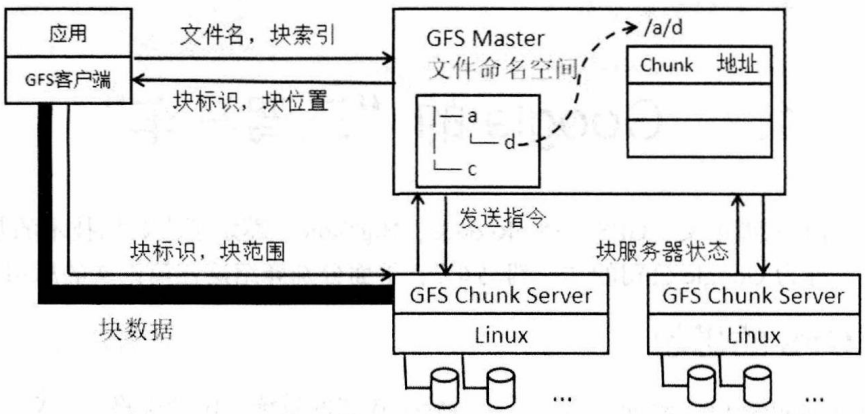


图 1-4　GFS 的架构

对于 GFS 架构，论文提到如下几个要点。

（1）GFS Master 节点管理所有的文件系统元数据，包括命名空间、访问控制信息、文件和块的映射信息以及当前块的位置信息。

（2）GFS 存储的文件都被分割成固定大小的块，每个块都会被复制到多个块服务器上（可靠性）。块的冗余度默认为 3。

（3）GFS Master 还管理着系统范围内的活动，比如块服务器之间的数据迁移等。

（4）GFS Master 与每个块服务器通信（发送心跳包），发送指令，获取状态。

论文也提到“副本的位置”的要求，即块副本位置选择的策略要满足两大目标：最大化数据可靠性和可用性。

1.2.2　MapReduce 的思想

在讨论 MapReduce 之前，我们先讨论一个与“PageRank”相关的问题。PageRank，即网页排名，又称网页级别。如果现在有 1~4 四个网页，网页 1 的内容有链接到网页 2、网页 3、网页 4，

网页 2 的内容有链接到网页 3、网页 4，网页 3 没有链接到其他页面，网页 4 有内容链接到网页 3，如图 1-5 所示。

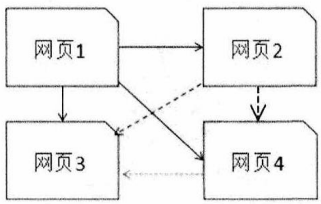


图 1-5　网页链接关系

用一个矩阵向量表来表达这几个网页的关联关系，如图 1-6 所示。例如，网页 1 有内容链接到网页 2，则在第二行第三列标“1”，否则标“0”。

	网页1	网页2	网页3	网页4
网页1	0	1	1	1
网页2	0	0	1	1
网页3	0	0	0	0
网页4	0	0	1	0

图 1-6　用矩阵向量表表示网页链接关系

计算这个 4×4 的矩阵，计算机丝毫没有问题。但如果网页非常多，比如计算 1 亿×1 亿的矩阵呢？则这个矩阵就非常大，一台计算机则计算不了，该怎么办呢？

有一个方法，就是把这个矩阵进行细分，分成很多小的矩阵。对每个小矩阵计算后，获得一个中间结果，再把中间结果合并起来，得到最终的结果。这其实就是“MapReduce”，如图 1-7 所示。

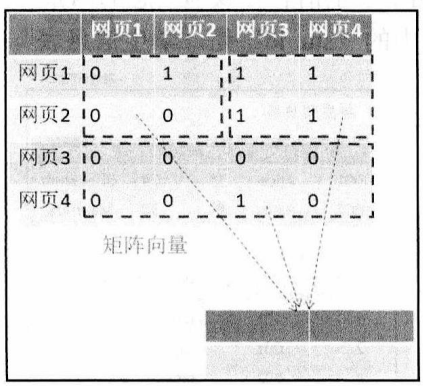
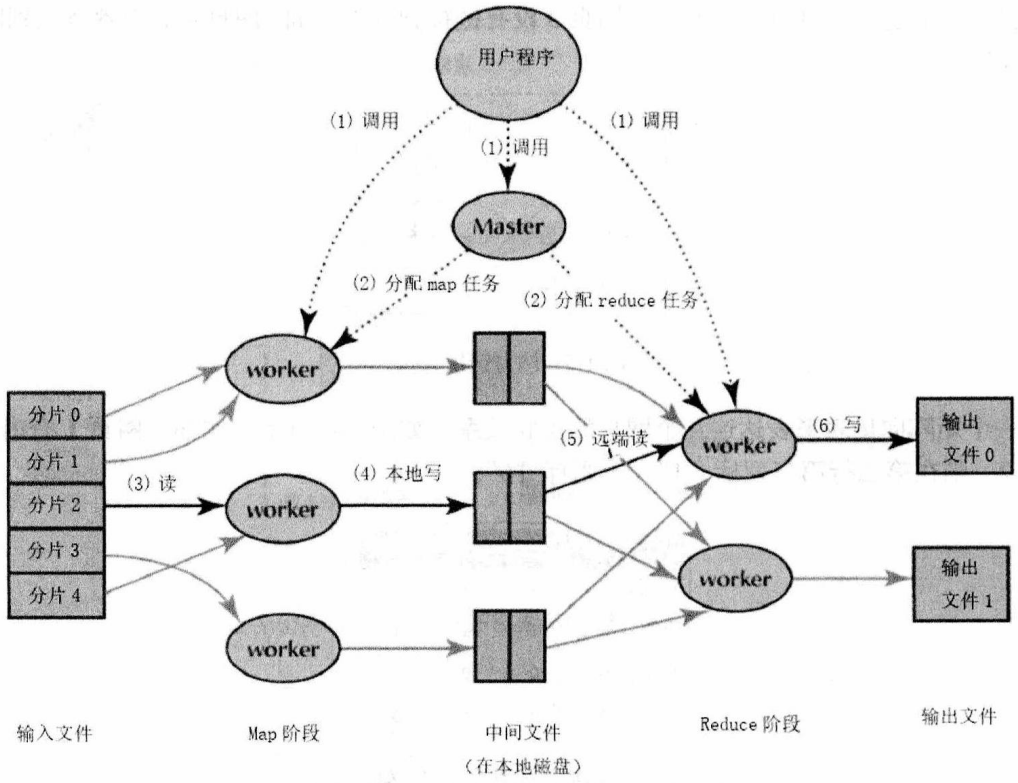


图 1-7　分成小块再计算

图 1-8 所示是论文“*MapReduce*: *Simplified Data Processing on Large Clusters*”描述的 MapReduce 的原理图。MapReduce 采用“分而治之”的思想，把对大规模数据集的操作，分发给一个主节点管理下的各个子节点共同完成，然后整合各个子节点的中间结果，得到最终的计算结果。简而言之，MapReduce 就是“分散任务，汇总结果”。



（Master 为主节点，worker 为工作节点）

图 1-8　MapReduce 的分而治之

1.2.3　BigTable 的思想

假设有一个学生信息系统记录了学生信息和成绩信息。该系统采用关系数据库保存数据。图 1-9 所示是关系数据库中存储了学生信息和成绩信息的两张表。可以看出，两张表各自有相应的字段的定义。“学生成绩表”中的“stu_id”是“学生信息表”中的外键。

学生信息表

stu_id	name	sex	age	address
001	Alice	女	24	ShenZhen
002	Rain	男	30	Maoming

学生成绩表

score_id	stu_id	course	score
1	001	chinese	89
2	001	math	90
3	002	chinese	91

图 1-9　学生信息系统中的两张表

而采用 BigTable 存储这两张表的数据，存储模型如图 1-10 所示。它把数据分成两个列族（Column Family）存放，分别是：info、score。每个列族下存放的数据都有一个行键（RowKey），它相当于关系型数据库的主键。行键（RowKey）可以重复，但不能为空。相同的行键的数据都属于同一行记录。每个列族下有多个列（Column）。列族在创建表时就固定下来，但列族下面的列可以随意定义。

RowKey	info(列族)				score(列族)	
	name	sex	age	address	chinese	math
001	Alice					
001		女				
001			24			
001				ShenZhen		
001					89	
001						90
002	Rain					

图 1-10　BigTable 存储模型示意

可以用图 1-11 所示来表达 BigTable 的数据模型。可以看出，表格（Table）由行键（RowKey）和列族（Column Family）组成，每个列族下又分了多个列（Column），每个列下包含了时间戳（Timestamp）和值（Value）。这里的 Timestamp 可以理解为是 Value 的版本号（Version），所以同一个列下的数据可能会存在多个版本。

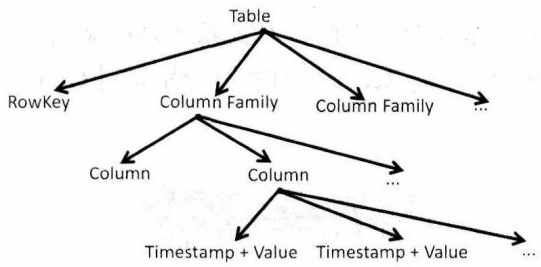


图 1-11　BigTable 存储的数据模型

表中的行用分区来管理，每个分区叫作一个“Tablet”，如图 1-12 所示。

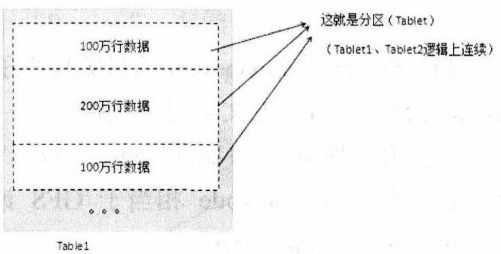


图 1-12　分区（Tablet）

Tablet Server 存储多个 Tablet，如图 1-13 所示。

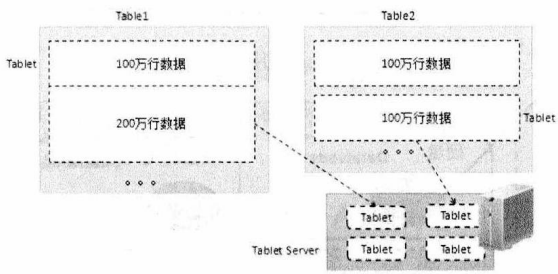


图 1-13　Tablet Server

BigTable 的架构如图 1-14 所示。Master 用来管理多个 Tablet Server。

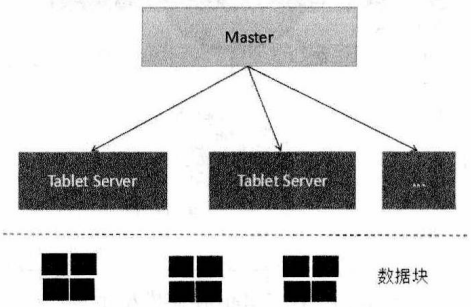


图 1-14　BigTable 架构图

1.3　Hadoop 概述

Hadoop 是一个由 Apache 基金会开发的分布式系统基础架构。Apache Hadoop 的 Logo 如图 1-15 所示。Hadoop 的 HDFS、MapReduce、HBase 分别是对 Google 公司的 GFS、MapReduce、BigTable 思想的开源实现。

图 1-15　Apache Hadoop 的 Logo

1.3.1　Hadoop 对 Google 公司三篇论文思想的实现

1. HDFS

HDFS（Hadoop Distributed File System）是 Hadoop 项目的核心子项目，是分布式计算中数据存储管理的基础，它是对 Google 公司的 GFS 论文思想的实现。

HDFS 架构如图 1-16 所示。它由名称节点（NameNode）、数据节点（DataNode）、第二名称节点组成（SecondaryNameNode）组成。NameNode 相当于 GFS 论文提到的 GFS Master，而 DataNode 相当于 GFS 论文提到的 GFS Chunk Server。

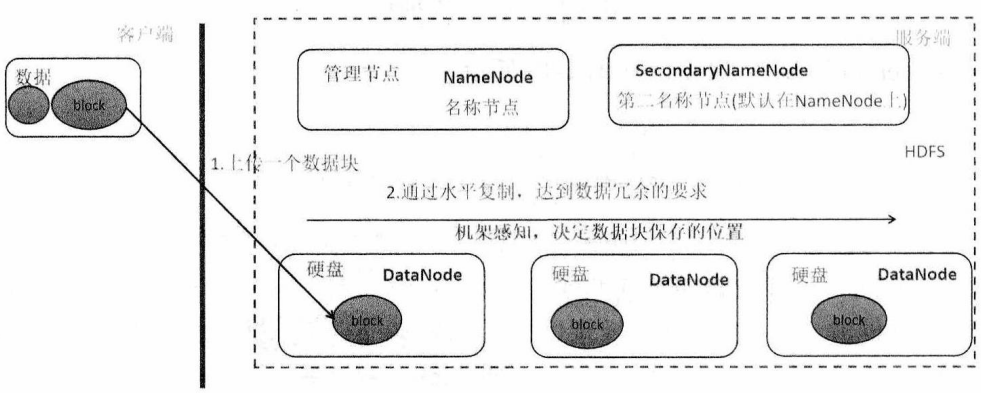


图 1-16　HDFS 的组成部分