大数据背景下

数据挖掘及处理分析

李继光 杨 迪 著

DASHUJU BEIJING
XIA SHUJU WAJUE JI
CHULI FENXI

中国海洋大学出版社
CHINA OCEAN UNIVERSITY PRESS

大数据背景下

数据挖掘及处理分析

李继光 杨 迪 著

中国海洋大学出版社

·青岛·

图书在版编目（CIP）数据

大数据背景下数据挖掘及处理分析 / 李继光，杨迪著. — 青岛：中国海洋大学出版社，2018.9

ISBN 978-7-5670-1491-6

Ⅰ. ①大… Ⅱ. ①李… ②杨… Ⅲ. ①数据采集—研究②数据处理—研究 Ⅳ. ① TP274

中国版本图书馆 CIP 数据核字 (2018) 第 208294 号

大数据背景下数据挖掘及处理分析

出 版 人	杨立敏		
出版发行	中国海洋大学出版社有限公司		
社　　址	青岛市香港东路 23 号	邮政编码	266071
网　　址	http://www.ouc-press.com		
责任编辑	邓志科	电　　话	0532-85901040
电子邮箱	dengzhike@sohu.com		
图片统筹	河北优盛文化传播有限公司		
装帧设计	河北优盛文化传播有限公司		
印　　制	定州启航印刷有限公司		
版　　次	2019 年 1 月第 1 版		
印　　次	2019 年 1 月第 1 次印刷		
成品尺寸	170mm × 240mm	印　　张	17
字　　数	303 千	印　　数	1-1000
书　　号	ISBN 978-7-5670-1491-6	定　　价	65.00 元
订购电话	0532-82032573（传真）	18133833353	

发现印刷质量问题，请致电 18133833353 进行调换。

前　言

这是一个大数据爆发的时代。面对信息的激流，多元化数据的涌现，大数据已经为个人生活、企业经营，甚至国家与社会的发展都带来了机遇和挑战，成为IT信息产业中极具潜力的蓝海。

大数据时代已经来临，它将在众多领域掀起变革的巨浪。但我们要冷静地看到，大数据的核心在于为客户挖掘数据中蕴藏的价值，而不是软硬件的堆砌。因此，针对不同领域的大数据应用模式、商业模式研究将是大数据产业健康发展的关键。我们相信，在国家的统筹规划与支持下，通过各地方政府因地制宜制定大数据产业发展策略，通过国内外IT龙头企业以及众多创新企业的积极参与，大数据产业未来发展前景十分广阔。

本书内容首先对大数据及数据挖掘技术原理进行论述，然后对于数据的获取、可视化分析以及基于R语言的数据分析进行探索，针对数据的甄别模式和知识图谱与图数据挖掘分析，将数据挖掘与机器学习进行对比分析，结合应用实例探索数据挖掘的发展趋势。本书可使大家全面了解数据挖掘的技巧，领略大量探索和展示数据的图形功能，从而更加高效地进行分析与沟通。

由于水平有限，本书难免存在不足之处，敬请读者批评指正。

目 录

第一章　大数据时代数据挖掘

第一节　大数据概念

大约 2009 年，“大数据”才成为互联网信息技术行业的流行词汇。美国互联网数据中心指出，互联网上的数据每年将增长 50%，每两年便将翻一番，而目前世界上 90% 以上的数据是最近几年才产生的。此外，数据又并非单纯指人们在互联网上发布的信息，全世界的工业设备、汽车、电表上有着无数的数码传感器，随时测量和传递着有关位置、运动、震动、温度、湿度乃至空气中化学物质的变化，也产生了海量的数据信息。

数据充斥所带来的影响远远超出了企业界。贾斯汀·格里莫将数学与政治科学联系起来，他研究的内容涉及对博客文章、国会演讲和新闻稿进行计算机自动化分析等，希望借此洞察政治观点是如何传播的。在科学和体育、广告和公共卫生等其他许多领域中，也有着类似的情况——也就是朝着数据驱动型的发现和决策的方向发生转变。

在公共卫生、经济发展和经济预测等领域中，“大数据”的预见能力正在被开发中，而且已经崭露头角。研究者发现，曾有一次他们发现“流感症状”和“流感治疗”等词汇在谷歌上的搜索查询量增加，而在几个星期以后，到某个地区医院急诊室就诊的流感病人数量就有所增加。

大数据技术的战略意义不在于掌握庞大的数据信息，而在于对这些含有意义的数据进行专业化处理。换言之，如果把大数据比作一种产业，那么这种产业实现盈利的关键，在于提高对数据的“加工能力”，通过“加工”实现数据的“增值”。且中国物联网校企联盟认为，物联网的发展离不开大数据，依靠大数据可以提供足够有利的资源。

随着云时代的来临，大数据（Big data）也吸引了越来越多的关注。《著云台》的分析师团队认为，大数据（Big data）通常用来形容一个公司创造的大量非结构化和半结构化数据，这些数据在下载到关系数据库用于分析时会花费过

多时间和金钱。大数据分析常和云计算联系到一起，因为实时的大型数据集分析需要像分布式计算系统（Map Reduce）一样的框架来向数十、数百或甚至数千的计算机分配工作。

大数据分析相比于传统的数据仓库应用，具有数据量大、查询分析复杂等特点。

大数据最主要的作用是服务，即面向人、机、物的服务。对机器来说，需要数据有一些关联，能够从中分析出有用的信息，非结构化、半结构化、结构化等。人、机、物对数据的贡献和参与度非常高，从数据规模上，可看到人到物理世界是从小到大，从数据质量来讲，人提供的数据质量是最高的。

传统数据库 / 数据仓库是 GB/TB 级高质量、较干净、强结构化、自顶向下（Top-down）、重交易、确定解。大数据是 PB 级的，有噪声、有冗余、非结构化、自下而上（Bottom-up）、重交互、满意解。大数据出现后，非关系型数据库（NoSQL）模式变得非常流行。大数据引发了一些问题，如对数据库高并发读写要求、对海量数据的高效存储和访问需求、对数据库高可扩展性和高可用性的需求，传统结构化查询语言（SQL）主要性能没有用武之地。互联网巨头对于 NoSQL 数据模式应用非常广泛，如谷歌的分布式数据存储系统（Big Table）、脸谱网（Facebook）的开源分布式 NoSQL 数据库系统（Cassandra）、甲骨文公司（Oracle）的 NoSQL 及亚马逊的亚马逊 key-value 模式的存储平台（Dynamo）等。从大数据处理角度来看，Map Reduce 成为事实的标准。大数据的存储和处理，已有了成熟解决方案，对于在系统软件中占较大比重的操作系统来说没有太大变化，一些重要的命题还没有解决，例如，操作系统对新兴计算资源的直接抽象的调度（GPU、APU），分布式文件系统下的统一数据视图、全数据中心范围内能耗管理、大数据下的安全性等，还不成熟，需要研发。

大多数研究大数据的商业公司，都有明确的商业目的，即更好地支撑全球广域网（Web）服务，如谷歌搜索引擎服务、美国脸谱网络服务网站（Facebook SNS 网站）、新浪微博网站等。在大数据驱动下的 Web 服务特征：更加流畅的网页交互体验，更加快速的社会资讯获取，更加便捷的日常工作和生活，更加深入的人、机、物融合。

回顾一下 Web 的发展，也是国际上比较通用的说法，Web1.0 时代 Web 内容主要由网站服务商提供，Web2.0 时代用户大量参与 Web 内容的贡献，像博客和微博。到了 Web3.0 时代，特征就是人、机、物共同参与 Web 内容贡献，使 Web 形成对真实世界的全面映射。

大数据来源于人、机、物，同时服务于人、机、物，大数据时代系统软件，特别是操作系统有待进一步发展，人、机、物融合大数据将推动 Web 进入崭新 Web3.0 时代。

第二节　大数据的现状与挑战

一、大数据对社会的影响

数据用户行为反映真实需求。一切行为皆有前兆，未来的不确定性，是人类产生工具类的根源之一。简单来说，从各种各样的数据中快速获取有价值的信息能力，即为大数据。大数据时代，软件价值体现在带来的数据规模、流量与活性；公司价值在于其拥有数据的规模、活性以及收集、运用数据的能力，决定公司的核心竞争力。以国家层面看，国家数据主权体现在对数据的占有和控制。数字主权将是继边防、海防、空防之后，另一个大国博弈的空间。

（一）泛互联网化

泛互联网化是收集用户数据的唯一低成本方式，能够带来数据规模和数据活性。泛互联网化带来软件使用的三个变化：跨平台、碎片化和门户化。

1. 跨平台

应用软件深度整合网络浏览器功能，桌面、移动终端（手机，平板电脑）拥有相同的体验和协同的功能。

2. 门户化

用户无须启用其他软件即可完成绝大多数的工作和沟通需求，对于个性化的用户需求，可以直接调用第三方应用或者插件完成。

3. 碎片化

把原来大型臃肿的软件拆分成多个独立的功能组件，用户可以按需下载使用。

这三个特征的核心意义分别在于收集用户行为资源，提高客户黏性；降低软件总体拥有成本，改变商业模式。

（二）行业垂直整合

开源软件加剧了基础软件的同质化趋势，而软、硬件一体化的趋势，进一步弱化了产业链上游的发言权。大数据产业结构发展趋势有两个维度：第一维度是大数据产业链，围绕数据的采集、整理、分析和反馈。第二维度是垂直的

行业，像媒体、零售、金融服务、医疗和电信。

从这两个维度来看，大数据也有三类商业模式：第一，大数据价值链环节，专注于价值链的高附加值环节。第二，垂直产业大致数据整合，利用大数据提高垂直产业效率。第三，大数据使能者，提供大数据基础设置、技术和工具。

（三）数据成为资产

未来企业的竞争，将是拥有数据规模和活性的竞争，将是对数据解释和运用的竞争。围绕数据，可以演绎出六种新的商业模式：租售数据模式、租售信息模式、数据媒体模式、数据使用模式、数据空间运营模式、大数据技术提供商。

1. 租售数据模式

简单来说，即是租 / 售广泛收集、精心过滤、时效性强的数据。

2. 租售信息模式

一般聚集某个行业，广泛收集相关数据、深度整合萃取信息，以庞大的数据中心加上专用传播渠道，也可成为一方霸主。此处，信息指的是经过加工处理，承载一定行业特征的数据集合。

3. 数据媒体模式

全球广告市场空间为 5000 亿美元，具备培育千亿级公司的土壤和成长空间。这类公司的核心资源是获得实时、海量、有效的数据，立身之本是大数据分析技术，盈利来源于精准营销。

4. 数据使用模式

如果没有大量的数据，缺乏有效的数据分析技术，这些公司的业务其实难以开展。通过在线分析小微企业的交易数据、财务数据，甚至可以计算出应提供多少货款，多长时间可以收回等关键问题，把坏账风险降到最低。

5. 数据空间运营模式

从历史上看，传统的互联网数据中心（IDC）即为这种模式，互联网巨头都在提供此类服务。海外的多宝箱（Dropbox）、国内微盘都是此类公司的代表。这类公司的想象空间在于可以成长为数据聚合平台，盈利模式将趋于多元化。

6. 大数据技术提供商

从数据量上来看，非结构化数据是结构化数据的 5 倍以上，任何一个各类的非结构化数据处理都可以重现现有结构化数据的辉煌。语音数据处理领域、视频数据处理领域、语义识别领域、图像数据处理领域都可能出现大型的、高速成长的公司。

（四）云平台数据更加完善

企业越来越希望能将自己的各类应用程序及基础设施转移到云平台上。就像其他 IT 系统那样，大数据的分析工具和数据库也将走向云计算。

云计算能为大数据带来哪些变化呢？

首先，云计算为大数据提供了可以弹性扩展、相对便宜的存储空间和计算资源，使得中小企业也可以像亚马逊一样通过云计算来完成大数据分析。其次，云计算 IT 资源庞大、分布较为广泛，是异构系统较多的企业及时准确处理数据的有力方式，甚至是唯一的方式。

当然，大数据要走向云计算，还有赖于数据通信带宽的提高和云资源池的建设，需要确保原始数据能迁移到云环境以及资源池可以随需弹性扩展。

二、大数据的挑战、现状与展望

大数据分析相比于传统的数据仓库应用，具有数据量大、查询分析复杂等特点。为了设计适合大数据分析的数据仓库架构，列举了大数据分析平台需要具备的几个重要特性，对当前的主流实现平台——并行数据库、分布式计算系统（Map Reduce）及基于两者的混合架构进行了分析归纳，指出了各自的优势及不足，同时也对各个方向的研究现状及大数据分析方面进行介绍，并展望未来。

（一）概述

最近几年，数据仓库又成为数据管理研究的热点领域，主要原因是当前数据仓库系统面临的需求在数据源、需提供的数据服务和所处的硬件环境等方面发生了根本性的变化，这些变化是我们必须面对的。

1. 三个变化

（1）数据量。

由 TB 级升到 PB 级，并仍在持续爆炸式增长。2011 年经调查显示，最大的数据仓库中的数据量，每两年增加 3 倍（年均增长率为 173%），其增长速度远超摩尔定律增长速度。照此增长速度计算，最近几年最大数据仓库中的数据量将逼近 100PB。

（2）分析需求。

由常规分析转向深度分析（Deep Analytics）。数据分析日益成为企业利润必不可少的支撑点。根据中国商业智能网（TDWI）对大数据分析的报告，企业已经不满足于对现有数据的分析和监测，而是期望能对未来趋势有更多的分析和预测，以增强企业竞争力。这些分析操作包括诸如移动平均线分析、数据关联关系分析、回归分析、市场分析等复杂统计分析，我们称之为深度分析。

（3）硬件平台。

由高端服务器转向由中低端硬件构成的大规模机群平台。由于数据量的迅速增加，并行数据库的规模不得不随之增大，从而导致其成本的急剧上升。出于成本的考虑，越来越多的企业将应用由高端服务器转向了由中低端硬件构成的大规模机群平台。

2. 两个问题

传统的数据仓库将整个实现划分为 4 个层次，数据源中的数据首先通过 ETL 工具被抽取到数据仓库中进行集中存储和管理，再按照星形模型或雪花模型组织数据，然后由联机分析处理（OLAP）工具从数据仓库中读取数据，生成数据立方体（MOLAP）或者直接访问数据仓库进行数据分析（ROLAP）。在大数据时代，此种计算模式存在以下两个问题。

（1）数据移动代价过高。

在数据源层和分析层间引入一个存储管理层，可以提升数据质量并针对查询进行优化，但也付出了较大的数据迁移代价和执行时的连接代价。数据首先通过复杂且耗时的 ETL 过程存储到数据仓库中，在 OLAP 服务器中转化为星形模型或者雪花模型；执行分析时，又通过连接方式将数据从数据库中取出，这些代价在 TB 级时也许可以接受，但面对大数据，其执行时间至少会增长几个数量级。更为重要的是，对于大量的即时分析，这种数据移动的计算模式是不可取的。

（2）不能快速适应变化。

传统的数据仓库假设主题是较少变化的，其应对变化的方式是对数据源到前端展现的整个流程中的每个部分进行修改，然后再重新加载数据，甚至重新计算数据，导致其适应变化的周期较长。这种模式比较适合对数据质量和查询性能要求较高，而不太计较预处理代价的场合。但在大数据时代，分析处在变化的业务环境中，这种模式将难以适应新的需求。

3. 一个鸿沟

在大数据时代，巨量数据与系统的数据处理能力间将会产生一个鸿沟：一边是至少 PB 级的数据量，另一边是面向传统数据分析能力设计的数据仓库和各种商业智能（BI）工具。如果这些系统工具发展缓慢，该鸿沟将会随着数据量的持续爆炸式增长而逐步拉大。

虽然，传统数据仓库可以采用舍弃不重要数据或者建立数据集市的方式来缓解此问题，但毕竟只是权宜之策，并非系统级解决方案。而且，舍弃的数据在未来可能会重新使用，以发掘出更大的价值。

（二）期望特性

数据仓库系统需具备几个重要特性，如表 1-1 所示。

表1-1　数据仓库系统需要具备的特性

特性	说明
高度可扩展性	横向大规模可扩展，大规模并行处理
高性能	快速响应复杂查询与分析
高度容错性	对硬件平台一致性要求不高，适应能力强
支持异构环境	业务需求变化时，能快速反应
较低的分析延迟	既能方便查询，又能处理复杂分析
较低成本	较高的性价比
向下兼容性	支持传统的商务智能工具

1. 高度可扩展性

一个明显的事实是，数据库不能依靠一台或少数几台机器的升级（scale-up 纵向扩展）满足数据量的爆炸式增长，而是希望能方便地做到横向可扩展（scale-out）来实现此目标。

普遍认为无共享结构（shared-nothing）（每个节点拥有私有内存和磁盘，并且通过高速网络与其他节点互连）具备较好的扩展性。分析型操作往往涉及大规模的并行扫描、多维聚集及星形连接操作，这些操作也比较适合在无共享结构的网络环境下运行。美国天睿公司（Teradata）即采用此结构，Oracle 在其新产品 Oracle 数据库一体机（Exadata）中也采用了此结构。

2. 高性能

数据量的增长并没有降低对数据库性能的要求，反而有所提高。软件系统性能的提升可以降低企业对硬件的投入成本、节省计算资源，提高系统吞吐量。巨量数据的效率优化，并行是必由之路。1PB 数据在 50MB/S 速度下串行扫描一次，需要 230 天；而在 6000 块磁盘上，并行扫描 1PB 数据只需要一小时。

3. 高度容错性

大数据的容错性要求在查询执行过程中，一个参与节点失效时，不需要重做整个查询，而机群节点数的增加会带来节点失效概率的增加。在大规模机群

环境下，节点的失效将不再是稀有事件（根据谷歌报告，平均每个 Map Reduce 数据处理任务即有 1.2 个工作节点失效）。因此在大规模机群环境下，系统不能依赖于硬件来保证容错性，要更多地考虑软件容错。

4. 支持异构环境

建设同构系统的大规模机群难度较大，原因在于计算机硬件更新较快，一次性购置大量同构的计算机是不可取的，而且也会在未来添置异构计算资源。此外，不少企业已经积累了一些闲置的计算机资源，此种情况下，对异构环境不同节点的性能是不一样的，可能出现“木桶效应”，即最慢节点的性能决定整体处理性能。因此，异构的机群需要特别关注负载均衡、任务调度等方面的设计。

5. 较低的分析延迟

分析延迟是分析前的数据准备时间。在大数据时代，分析所处的业务环境是变化的，因此也要求系统能动态地适应业务分析需求。在分析需求发生变化时，减少数据准备时间，系统能尽可能快地做出反应，快速地进行数据分析。

6. 较低的成本

在满足需求的前提下，使技术成本越低，其生命力就越强。值得指出的是，成本是一个综合指标，不仅仅是硬件或软件的代价，还应包括日常运维成本（网络费用、电费、建筑等）和管理人员成本等。据报告，数据中心的主要成本不是硬件的购置成本，而是日常运维成本，因此，在设计系统时需要更多地关注此项内容。

7. 向下兼容性

数据仓库发展的 30 年，产生了大量面向客户业务的数据处理工具（如 Informactica、Data Stage 等）、分析软件（如 SPSS、R、MATLAB 等）和前端展现工具（如水晶报表）等。这些软件是一笔宝贵的财富，已被分析人员所熟悉，是大数据时代中小规模数据分析的必要补充。因此，新的数据仓库需考虑同传统商务智能工具的兼容性。由于这些系统往往提供标准驱动程度，如 ODBC、JDBC 等，这项需求的实际要求是对 SQL 的支持。

总而言之，以较低的成本投入、高效地进行数据分析是大数据分析的基本目标。

（三）并行数据库

并行数据库系统（Parallel Database System）是新一代高性能的数据库系统，是在 MPP 和集群并行计算环境的基础上建立的数据库系统。

并行数据库技术起源于 20 世纪 70 年代的数据库机（Database Machine）的研究，研究的内容主要集中在关系代数操作的并行化和实现关系操作的专用

硬件设计上，希望通过硬件实现关系数据库操作的某些功能，该研究以失败而告终。80 年代后期，并行数据库技术的研究方向逐步转到了通用并行机方面，研究的重点是并行数据库的物理组织、操作算法、优化和调度策略。从 90 年代至今，随着处理器、存储、网络等相关基础技术的发展，并行数据库技术的研究上升到一个新的水平，研究的重点也转移到数据操作的时间并行性和空间并行性上。

并行数据库系统的目标是高性能（High Performance）和高可用性（High Availability），通过多个处理节点并行执行数据库任务，提高整个数据库系统的性能和可用性。

性能指标关注的是并行数据库系统的处理能力，具体的表现可以统一总结为数据库系统处理事务的响应时间。并行数据库系统的高性能可以从两方面理解，一个是速度提升（Speed-Up），一个是范围提升（Scale-up）。速度提升是指，通过并行处理，可以使用更少的时间完成更多样的数据库事务。范围提升是指，通过并行处理，在相同的处理时间内，可以完成更多的数据库事务。并行数据库系统基于多处理节点的物理结构，将数据库管理技术与并行处理技术有机结合，来实现系统的高性能。

可用性指标关注的是并行数据库系统的健壮性，也就是当并行处理节点中的一个节点或多个节点部分失效或完全失效时，整个系统对外持续响应的能力。高可用性可以同时在硬件件两方面提供保障。

（1）在硬件方面，通过冗余的处理节点、存储设备、网络链路等硬件措施，可以保证当系统中某节点部分或完全失效时，其他的硬件设备可以接手其处理，对外提供持续服务。

（2）在软件方面，通过状态监控与跟踪、互相备份、日志等技术手段，可以保证当前系统中某节点部分或完全失效时，由它所进行的处理或由它所掌控的资源可以无损失或基本无损失地转移到其他节点，并由其他节点继续对外提供服务。

为了实现和保证高性能和高可用性，可扩充性也成为并行数据库系统的一个重要指标。可扩充性是指，并行数据库系统通过增加处理节点或者硬件资源（处理器、内存等），使其可以平滑地或线性地扩展其整体处理能力的特性。

随着对并行计算技术研究的深入和对称多处理（SMP）、大规模并行处理（MPP）等处理机技术的发展，并行数据库的研究也进入了一个新的领域，集群已经成为并行数据库系统中最受关注的热点。目前，并行数据库领域主要还有下列问题需要进一步研究和解决。

（1）并行体系结构及其应用，这是并行数据库系统的基础问题。为了达到并行处理的目的，参与并行处理的各个处理节点之间是否要共享资源、共享哪些资源、需要多大程度的共享，这些就需要研究并行处理的体系结构及有关实现技术。

（2）并行数据库的物理设计，主要是在并行处理的环境下，数据分布的算法的研究、数据库设计工具与管理工具的研究。

（3）处理节点间通信机制的研究。为了实现并行数据库的高性能，并行处理节点要最大限度地协同处理数据库事务，因此，节点间必不可少地存在通信问题，如何支持大量节点之间消息和数据的高效通信，也成为并行数据库系统中一个重要的研究课题。

（4）并行操作算法，为提高并行处理的效率，需要在数据分布算法研究的基础上，深入研究链接、聚集、统计、排序等具体的数据操作在多节点上的并行操作算法。

（5）并行操作的优化和同步。为获得高性能，如何将一个数据库处理事务合理地分解成相对独立的并行操作步骤，如何将这些步骤以最优的方式在多个处理节点间进行分配，如何在多个处理节点的同一个步骤和不同步骤之间进行消息和数据的同步，这些问题都值得深入研究。

并行数据库中数据的加载和再组织技术。为了保证高性能和高可用性，并行数据库系统中的处理节点可能需要进行扩充（或者调整），这就需要考虑如何对原有数据进行卸载、加载，以及如何合理地在各个节点重新组织数据。

（四）分布式计算系统（Map Reduce）

Map Reduce 的编程模型不同于以前学过的大多数编程模型，它是一种用于大规模数据集（大于 1TB）的并行运算的编程模型。其概念映射（Map）和化简（Reduce），及它们的主要思想，都是从函数式编程语言里借来的，还有从矢量编程语言里借来的特性。它极大地方便了编程人员在不会分布式并行编程的情况下，将自己的程序在分布式系统上运行。当前的软件实现是指定一个映射（Map）函数，用来把一组键值对映射成一组新的键值对；指定并发的化简（Reduce）函数，用来保证所有映射的键值对中的每一个共享相同的键组。

Map Reduce 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分节点共同完成，接着通过整合各分节点的中间结果，得到最终的结果。简单来说，Map Reduce 就是“任务的分散与结果的汇总”。

Map Reduce 处理过程被 Map Reduce 高度地抽象为两个函数：Map 和 Reduce，Map 负责把任务分解成多个任务，Reduce 负责把分解后多任务处理的结果汇总起来。至于在并行编程中的其他种种复杂问题，如分布式存储、工

作调度、负载均衡、容错处理、网络通信等，均由 Map Reduce 框架负责处理，可以不用程序员烦心。值得注意的是，用 Map Reduce 来处理的数据集（或任务）必须具备这样的特点：待处理的数据集可以分解成许多小的数据集，且每一小数据集都可以完全并行地进行处理。

计算模型的核心部分是 Map 和 Reduce 函数。这两个函数的具体功能由用户根据自己设计实现，只要能够按照用户自定义的规则，将输入的〈*key*,*value*〉对转换成另一个或一批对〈*key*,*value*〉输出即可。

在 Map 阶段，Map Reduce 框架将任务的输入数据分隔成固定大小的片段，随后将每个片段进一步分解成一批键值对〈*K*1，*V*1〉。Hadoop 为每一个片段创建一个 Map 任务（可简称为 Mapper）用于执行用户自定义的 map 函数，并将对应片段中的〈*K*1，*V*1〉对作为输入，得到计算的中间结果〈*K*2，*V*2〉。接着将中间结果按照 K2 进行排序，并将键（key）值相同的字段名（value）放在一起形成一个新列表，形成〈*K*2，*list*(*V*2)〉元组。最后再根据 key 值的范围将这些组进行分组，对应不同的 Reduce 任务（可简称为 Reducer）。

在 Reduce 阶段，Reducer 把从不同 Mapper 接收来的数据整合在一起并进行排序，然后调用用户自定义的 reduce 函数，对输入的〈*K*2，*list*(*V*2)〉对进行相应的处理，得到键值对〈*K*3，*V*3〉并输出到分布式文件系统（HDFS）上。既然 Map Reduce 框架为每个片段创建一个 Mapper，那么谁来确定 Reducer 的数目呢？其答案是用户。Mapped-site.XML 配置文件中有一个表示 Reducer 数目的属性 mapped.reduce.tasks，该属性的默认值为 1，开发人员可通过 job.setNumReduceTasks（）方法重新设置该值。

MapReduce 架构结构组成部分主要有以下类型，下面给予介绍。

1. 组成部分

（1）JobClient。

每一个 job 都会在用户端通过 JobClient 类将应用程序以及配置参数打包成 jar 文件存储在 HDFS，并把路径提交到 JobTracker，然后由 JobTracker 创建每一个 Task（即 MapTask 和 ReduceTask）并将它们分发到各个 TaskTracker 服务中去执行。

（2）JobTracker。

JobTracker 是一个 master 服务，JobTracker 负责调度 job 的每一个子任务 task 运行于 TaskTracker 上，并监控它们，如果发现有失败的 task 就重新运行它。一般应该把 JobTracker 部署在单独的机器上。

（3）TaskTracker。

TaskTracker 是运行于多个节点上的 slaver 服务。TaskTracker 则负责直接