



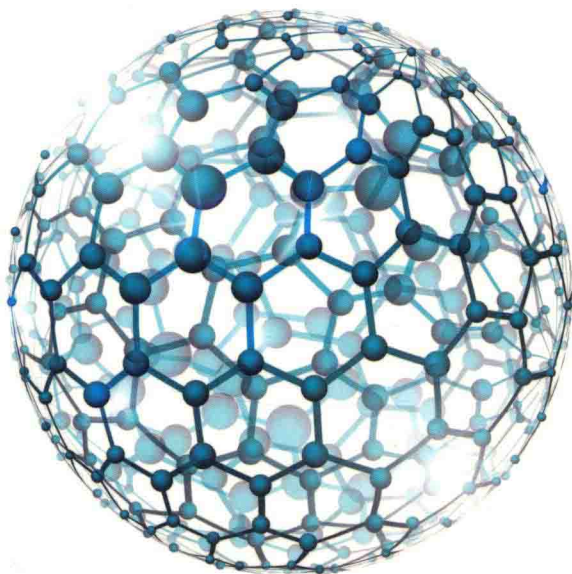
教育部高等学校计算机类专业教学指导委员会-华为ICT产学合作项目
数据科学与大数据技术专业系列规划教材

华为信息与网络
技术学院指定教材

NoSQL

数据库原理

侯宾◎编著



系统、完整的数据科学与大数据技术专业解决方案

名校名师打造大数据领域精品力作

掌握管理十亿行和十万列数据的工具

理解部署在上千节点上的数据库系统

 中国工信出版集团

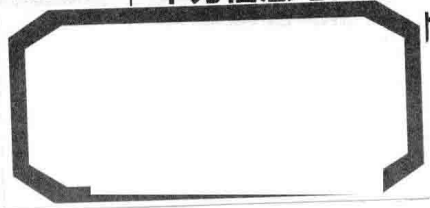
 人民邮电出版社
POSTS & TELECOM PRESS



教育部高等学校计算机类专业教学指导委员会-华为ICT产学合作项目

华为信息与网络

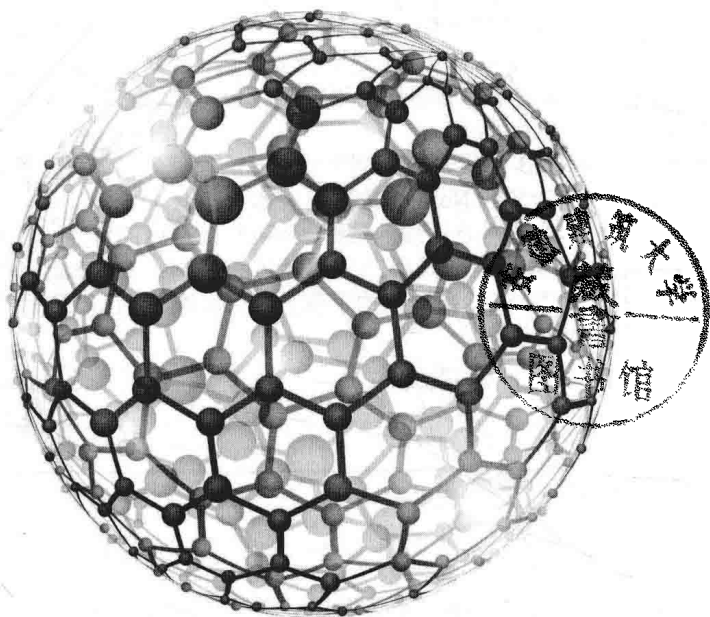
数据科学与大数据技术专业系列规划教材



NoSQL

数据库原理

侯宾 ● 编著



人民邮电出版社

北京

图书在版编目 (C I P) 数据

NoSQL数据库原理 / 侯宾编著. — 北京 : 人民邮电出版社, 2018.8
数据科学与大数据技术专业系列规划教材
ISBN 978-7-115-48306-5

I. ①N… II. ①侯… III. ①数据库系统—教材
IV. ①TP311.138

中国版本图书馆CIP数据核字(2018)第169745号

内 容 提 要

本书对统称为 NoSQL 的分布式非关系型数据库原理和使用方法进行介绍。

第 1 章, 首先介绍 NoSQL 数据库的起源背景和设计理念, 以及相关技术概念。然后介绍大数据技术体系, 以及 NoSQL 在该技术体系的地位和作用。

第 2 章, 首先回顾关系型数据库的主要机制, 然后介绍 NoSQL 数据库的常见技术原理, 以及 NoSQL 的常见存储模式。

第 3 章, 对 Hadoop 工具进行介绍, 重点介绍 HDFS 的技术原理和基本使用方法。

第 4 章, 介绍 HBase 的基本架构、基本使用方法和编程方法。

第 5 章, 介绍 HBase 中核心技术原理, 包括水平分区机制、数据写入机制、列族与合并机制等。对 HBase 中的高级管理方法、深入使用方法进行简要介绍, 并对 HBase 的第三方插件与工具进行介绍。

第 6 章, 介绍 Cassandra 的基本原理和使用方法。首先介绍 Amazon Dynamo 的相关原理, 然后介绍 Cassandra 的安装配置与 CQL 语言。

第 7 章, 介绍以 MongoDB 为代表的文档型数据库的原理和基本使用方法。

第 8 章, 介绍其他一些知名的 NoSQL 数据库技术与工具。

本书帮助读者从理论和实践两方面深入理解 NoSQL 数据库的特点。在理论上, 突出 NoSQL 数据库由于采用分布式架构和非关系型模式所产生的优势和限制; 在实践上, 给出命令行操作、Java 和 Python 语言编程等多种访问 NoSQL 数据库的示范方法。本书面向已经了解关系型数据库的原理和操作系统, 且具有一定编程基础的读者。

◆ 编 著 侯 宾

责任编辑 刘 博

责任印制 彭志环

◆ 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>

北京隆昌伟业印刷有限公司印刷

◆ 开本: 787×1092 1/16

印张: 16.25 2018 年 8 月第 1 版

字数: 421 千字 2018 年 8 月北京第 1 次印刷

定价: 49.80 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

教育部高等学校计算机类专业教学指导委员会-华为 ICT 产学合作项目
数据科学与大数据技术专业系列规划教材

编 委 会

- 主任 陈 钟 北京大学
副主任 杜小勇 中国人民大学
周傲英 华东师范大学
马殿富 北京航空航天大学
李战怀 西北工业大学
冯宝帅 华为技术有限公司
张立科 人民邮电出版社
秘书长 王 翔 华为技术有限公司
戴思俊 人民邮电出版社

委 员 (按姓名拼音排序)

- | | | | |
|-----|----------|-----|---------|
| 崔立真 | 山东大学 | 段立新 | 电子科技大学 |
| 高小鹏 | 北京航空航天大学 | 桂劲松 | 中南大学 |
| 侯 宾 | 北京邮电大学 | 黄 岚 | 吉林大学 |
| 林子雨 | 厦门大学 | 刘 博 | 人民邮电出版社 |
| 刘耀林 | 华为技术有限公司 | 乔亚男 | 西安交通大学 |
| 沈 刚 | 华中科技大学 | 石胜飞 | 哈尔滨工业大学 |
| 嵩 天 | 北京理工大学 | 唐 卓 | 湖南大学 |
| 汪 卫 | 复旦大学 | 王 伟 | 同济大学 |
| 王宏志 | 哈尔滨工业大学 | 王建民 | 清华大学 |
| 王兴伟 | 东北大学 | 薛志东 | 华中科技大学 |
| 印 鉴 | 中山大学 | 袁晓如 | 北京大学 |
| 张志峰 | 华为技术有限公司 | 赵卫东 | 复旦大学 |
| 邹北骥 | 中南大学 | 邹文波 | 人民邮电出版社 |

毫无疑问，我们正处在一个新时代。新一轮科技革命和产业变革正在加速推进，技术创新日益成为重塑经济发展模式和促进经济增长的重要驱动力量，而“大数据”无疑是第一核心推动力。

当前，发展大数据已经成为国家战略，大数据在引领经济社会发展中的新引擎作用更加突显。大数据重塑了传统产业的结构和形态，催生了众多的新产业、新业态、新模式，推动了共享经济的蓬勃发展，也给我们的衣食住行带来根本改变。同时，大数据是带动国家竞争力整体跃升和跨越式发展的巨大推动力，已成为全球科技和产业竞争的重要制高点。可以大胆预测，未来，大数据将会进一步激起全球科技和产业发展浪潮，进一步渗透到我们国计民生的各个领域，其发展扩张势不可挡。可以说，我们处在一个“大数据”时代。

大数据不仅仅是单一的技术发展领域和战略新兴产业，它还涉及科技、社会、伦理等诸多方面。发展大数据是一个复杂的系统工程，需要科技界、教育界和产业界等社会各界的广泛参与和通力合作，需要我们以更加开放的心态，以进步发展的理念，积极主动适应大数据时代所带来的深刻变革。总体而言，从全面协调可持续健康发展的角度，推动大数据发展需要注重以下五个方面的辩证统一和统筹兼顾。

一是要注重“长与短结合”。所谓“长”就是要目标长远，要注重制定大数据发展的顶层设计和中长期发展规划，明确发展方向和总体目标；所谓“短”就是要着眼当前，注重短期收益，从实处着手，快速起效，并形成效益反哺的良性循环。

二是要注重“快与慢结合”。所谓“快”就是要注重发挥新一代信息技术产业爆炸性增长的特点，发展大数据要时不我待，以实际应用需求为牵引加快推进，力争快速占领大数据技术和产业制高点；所谓“慢”就是防止急功近利，欲速而不达，要注重夯实大数据发展的基础，着重积累发展大数据基础理论与核心共性关键技术，培养行业领域发展中的大数据思维，潜心培育大数据专业人才。

三是要注重“高与低结合”。所谓“高”就是要打造大数据创新发展高地，要结合国家重大战略需求和国民经济主战场核心需求，部署高端大数据公共服务平台，组织开展国家级大数据重大示范工程，提升国民经济重点领域和标志性行业的大数据技术水平和应用能力；所谓“低”就是要坚持“润物细无声”，推进大数据在各行各业和民生领域的广泛应用，推进大数据发展的广度和深度。

四是要注重“内与外结合”。所谓“内”就是要向内深度挖掘和深入研究大数据作为一门学科领域的深刻技术内涵，构建和完善大数据发展的完整理论体系和技术支撑体系；所谓“外”就是要加强开放创新，由于大数据涉及众多学科领域和产业行业门类，也涉及国家、社会、个人等诸多问题，因此，需要推动国际国内科技界、产业界的深入合作和各级政府广泛参与，共同研究制定标准规范，推动大数据与人工智能、云计算、物联网、网络安全等信息技术领域的协同发展，促进数据科学与计算机科学、基础科学和各种应用科学的深度融合。

五是要注重“开与闭结合”。所谓“开”就是要坚持开放共享，要鼓励打破现有体制机制障碍，推动政府建立完善开放共享的大数据平台，加强科研机构、企业间技术交流合作，推动大数据资源高效利用，打破数据壁垒，普惠数据服务，缩小数据鸿沟，破除数据孤岛；所谓“闭”就是要形成价值链生态闭环，充分发挥大数据发展中技术驱动与需求牵引的双引擎作用，积极运用市场机制，形成技术创新链、产业发展链和资金服务链协同发展的态势，构建大数据产业良性发展的闭环生态圈。

总之，推动大数据的创新发展，已经成为了新时代的新诉求。刚刚闭幕的党的十九大更是明确提出要推动大数据、人工智能等信息技术产业与实体经济深度融合，培育新增长点，为建设网络强国、数字中国、智慧社会形成新动能。这一指导思想为我们未来发展大数据技术和产业指明了前进方向，提供了根本遵循。

习近平总书记多次强调“人才是创新的根基”“创新驱动实质上是人才驱动”。绘制大数据发展的宏伟蓝图迫切需要创新人才培养体制机制的支撑。因此，需要把高端人才队伍建设作为大数据技术和产业发展的重中之重，需要进一步完善大数据教育体系，加强人才储备和梯队建设，将以大数据为代表的新兴产业发展对人才的创新性、实践性需求渗透融入人才培养各个环节，加快形成我国大数据人才高地。

国家有关部门“与时俱进，因时施策”。近期，国务院办公厅正式印发《关于深化产教融合的若干意见》，推进人才和人力资源供给侧结构性改革，以适应创新驱动发展战略的新形势、新任务、新要求。教育部高等学校计算机类专业教学指导委员会、华为公司和人民邮电出版社组织编写的《教育部高等学校计算机类专业教学指导委员会-华为 ICT 产学合作项目——数据科学与大数据技术专业系列规划教材》的出版发行，就是落实国务院文件精神，深化教育供给

侧结构性改革的积极探索和实践。它是国内第一套成专业课程体系规划的数据科学与大数据技术专业系列教材，作者均来自国内一流高校，且具有丰富的大数据教学、科研、实践经验。它的出版发行，对完善大数据人才培养体系，加强人才储备和梯队建设，推进贯通大数据理论、方法、技术、产品与应用等的复合型人才培养，完善大数据领域学科布局，推动大数据领域学科建设具有重要意义。同时，本次产教融合的成功经验，对其他学科领域的人才培养也具有重要的参考价值。

我们有理由相信，在国家战略指引下，在社会各界的广泛参与和推动下，我国的大数据技术和产业发展一定会有光明的未来。

是为序。



中国科学院院士 郑志明

2018年4月16日

在 500 年前的大航海时代，哥伦布发现了新大陆，麦哲伦实现了环球航行，全球各大洲从此连接了起来，人类文明的进程得以推进。今天，在云计算、大数据、物联网、人工智能等新技术推动下，人类开启了智能时代。

面对这个以“万物感知、万物互联、万物智能”为特征的智能时代，“数字化转型”已是企业寻求突破和创新的必由之路，数字化带来的海量数据成为企业乃至整个社会最重要的核心资产。大数据已上升为国家战略，成为推动经济社会发展的新引擎，如何获取、存储、分析、应用这些大数据将是这个时代最热门的话题。

国家大数据战略和企业数字化转型成功的关键是培养多层次的大数据人才，然而，根据计世资讯的研究，2018 年中国大数据领域的人才缺口将超过 150 万人，人才短缺已成为制约产业发展的突出问题。

2018 年初，华为公司提出新的愿景与使命，即“把数字世界带入每个人、每个家庭、每个组织，构建万物互联的智能世界”，它承载了华为公司的历史使命和社会责任。华为企业 BG 将长期坚持“平台+生态”战略，协同生态伙伴，共同为行业客户打造云计算、大数据、物联网和传统 ICT 技术高度融合的数字化转型平台。

人才生态建设是支撑“平台+生态”战略的核心基石，是保持产业链活力和持续增长的根本，华为以 ICT 产业长期积累的技术、知识、经验和成功实践为基础，持续投入，构建 ICT 人才生态良性发展的使能平台，打造全球有影响力的 ICT 人才认证标准。面对未来人才的挑战，华为坚持与全球广大院校、伙伴加强合作，打造引领未来的 ICT 人才生态，助力行业数字化转型。

一套好的教材是人才培养的基础，也是教学质量的重要保障。本套教材的出版，是华为在大数据人才培养领域的重要举措，是华为集合产业与教育界的高端智力，全力奉献的结晶和成果。在此，让我对本套教材的各位作者表示由衷的感谢！此外，我们还要特别感谢教育部高等学校计算机类专业教学指导委员会副主任、北京大学陈钟教授以及秘书长、北京航空航天大学马殿富教授，没有你们的努力和推动，本套教材无法成型！

同学们、朋友们，翻过这篇序言，开启学习旅程，祝愿在大数据的海洋里，尽情展示你们的才华，实现你们的梦想！



华为公司董事、企业 BG 总裁 阎力大

2018 年 5 月

伴随着互联网的兴起，人们生产、收集数据的能力大大增强，也更加希望从数据中获得新价值。大数据业务不断出现，例如搜索引擎、推荐系统、位置服务与日志分析等。

传统的关系型数据库可能无法对这些新业务进行有效支撑。首先，关系型数据库的横向扩展能力较差，使之无法对不断增长的大数据进行有效管理；其次，关系型数据库的数据模型严格固定，使之对多源数据或复杂的数据类型的处理能力较差；最后，一些大数据业务对事务机制、一致性和完整性的要求较低，这使得关系型数据库的优势难以发挥作用，反而成为制约。

在这种背景下，多家互联网企业的技术团队不约而同地着手研发非关系型数据库，其核心理念是通过牺牲事务机制和强一致性保障机制，以获得更好的分布式部署能力和横向扩展能力；通过打破关系模型、创造新的数据模型，使其在特定大数据场景下，对特定业务数据的处理性能更佳。

随着新型数据库产品不断涌现，人们开始将一些流行的分布式、非关系型数据库统称为“NoSQL”，一般解释为“Not only SQL”。NoSQL并非一个严谨的概念，它更像一个技术革新运动的口号，代表了一个趋势：大数据时代来临，我们必须为不同的业务场景选择更适合的技术工具，此时不再强调技术工具的全面，而是强调“取舍”。此外，NoSQL并非是对关系型数据库或 SQL 语言的否定，而是对传统数据库的发展和补充。NoSQL 无法代替关系型数据库，就像关系型数据库也无法代替 NoSQL。

作为新生事物，NoSQL 在技术原理和使用方法上，都和传统数据库有所不同，而且，由于没有统一的标准，不同 NoSQL 产品之间的技术差别也非常大。这使得传统的数据库用户在产品选型和使用上无所适从，可能遇到以下问题。

第一，不理解不同 NoSQL 产品的设计理念和技术特点，因此难以进行产品选择。

第二，不理解分布式系统可能遇到的技术问题和解决方法，因此难以进行有效的部署、配置和优化。

第三，习惯使用 SQL 语句操作数据库，对新的数据库操作方法感到陌生。

针对这些问题，本书希望通过介绍 NoSQL 数据库的分布式架构、数据模型、数据管理策略等核心原理，使读者能够理解 NoSQL 数据库的特点，进而判断 NoSQL 的适用场景。通过

介绍典型 NoSQL 数据库的部署方法、数据操作方法和编程访问方法，使读者学会在实际工作中初步运用这些工具。

本书的前两章介绍 NoSQL 基本原理，主要突出 NoSQL 数据库为进行分布式部署、实现高可用性、高效率和易用性等目标所采取的设计方法和功能取舍。

本书的后续章节分类介绍具体的 NoSQL 数据库，主要选择国内外较为流行的 HBase、Cassandra、MongoDB 和 Neo4j 等数据库进行讲解。

HBase 是一种经典的、基于列族的 NoSQL 数据库，一般会与著名大数据工具 Hadoop 共同部署、配合使用。由于 Hadoop 是大数据工具的一个事实标准，很多大数据工具都会保持和 Hadoop、HBase 的兼容性，这使得 HBase 的应用领域很广，构建各类解决方案的能力很强，在互联网、电信、电力、金融等行业均能看到其使用案例。

通过解读 HBase 的核心技术原理和使用方法，读者能够对 NoSQL 产生清晰的第一印象，也能够和其他 NoSQL 数据库进行对照学习。

Cassandra 也是一种基于列族的 NoSQL 数据库，其特点在于，一方面，采用了环形拓扑结构，避免了主节点单点失效问题；另一方面，提供了类似 SQL 的 CQL 语言。通过对 Cassandra 的学习，读者可以理解列族数据模型的优势和限制，理解这些特点和底层结构相关，和操作方法无关。

MongoDB 是一种基于文档模型的分布式 NoSQL 数据库，其特点在于可以利用单表存储复杂的数据结构，例如列的嵌套，而不必理会关系型数据库所要求的列原子性。此外，MongoDB 支持非常灵活的数据查询操作，且在大数据量的情况下，查询性能也很高。

Neo4j 是一个图数据库，其分布式部署能力较弱，但数据结构很有特色，适合存储节点和关系模型，例如社交媒体上的关注关系、网页之间的链接关系等。

在内容侧重上，本书的覆盖面较大，介绍了多种软件的原理和部署、操作和编程方法。

在原理方面，本书尽可能介绍和 NoSQL 直接相关的、有特色的技术原理，并尽量用比较通俗的方式进行解释，避免对技术原理进行过于深入的分析。此外，本书给出了一些经典的技术论文，可供学习者做扩展阅读。

在部署方法上，本书主要介绍部署的核心步骤和基本方法，例如尽可能多地采用默认参数等，目的是突出分布式软件的部署需求和重点配置内容。详细步骤将通过配套实验等方式体现。

在使用方法上，本书主要介绍基本的库表操作、数据增删改查方法、数据批量操作等方法，并介绍不同软件所提供的特色功能。受篇幅所限，所介绍的方法在深度和覆盖面上均有所保留。

在编程方法上，本书提供 Java 和 Python 两种语言的示例。一般来说，通过 Java 编程可以实现更多功能和更高的效率；而 Python 语言更易使用，更容易进行功能验证，学习时可以根据需要进行取舍。为突出主题，本书并未对 Java 和 Python 的基础编程方法、开发环境以及如何导入依赖包等问题进行介绍。

考虑到篇幅和难度，本书对于进阶性内容涉及较少，这主要体现在以下几点，缺少分布式环境下的定制化部署、性能优化和运维管理等内容；缺少对多个大数据工具联合使用的方法和步骤的介绍；缺少对认证、授权等安全机制的配置方法介绍；缺少对高级编程方法的介绍等。这些问题要么涉及过多的基础、外延知识和操作步骤，要么所涉及的实践环境很难满足，或者和具体的商业化工具有关。

特别是考虑到一般情况下，实验环境可能有所限制，对于如何在分布式环境下进阶使用和维护 NoSQL 数据库等内容，本书介绍较少。

限于本人的能力和水平，以及写作时间仓促，书中必然存在错误、疏漏或欠妥之处。恳请各界读者批评指正，以便今后不断完善、改进！

编者

2018年3月

目 录 CONTENTS

第 1 章 绪论 1

1.1 数据库的相关概念 2
1.1.1 关系型数据库管理系统 2
1.1.2 关系型数据库的瓶颈 4
1.1.3 NoSQL 的特点 4
1.1.4 NewSQL 的概念 6
1.1.5 NoSQL 的典型应用场景 7
1.2 大数据的技术体系 8
1.2.1 大数据的特征 9
1.2.2 大数据的采集 10
1.2.3 大数据的存储 11
1.2.4 大数据的管理和使用 13
1.2.5 数据可视化 13
1.2.6 大数据安全与治理 15
小结 15
思考题 16

第 2 章 NoSQL 数据库的基本原理 17

2.1 关系型数据库的原理简述 18
2.1.1 关系模型 18
2.1.2 关系型数据库的完整性约束 19
2.1.3 关系型数据库的事务机制 19
2.1.4 关系型数据库的分布式部署 21
2.2 分布式数据管理的特点 23
2.2.1 数据分片 24
2.2.2 数据多副本 24
2.2.3 一次写入多次读取 26
2.2.4 分布式系统的可伸缩性 27

2.3 分布式系统的一致性问题 27
2.3.1 CAP 原理 28
2.3.2 BASE 和最终一致性 29
2.3.3 Paxos 算法简介 30
2.4 NoSQL 的常见模式 33
2.4.1 键值对存储模式 33
2.4.2 文档式存储模式 34
2.4.3 列存储模式 35
2.4.4 图存储模式 36
2.5 NoSQL 系统的其他相关技术 37
2.5.1 分布式数据处理 37
2.5.2 时间同步服务 38
2.5.3 布隆过滤器 38
小结 40
思考题 40

第 3 章 HDFS 的基本原理 41

3.1 Hadoop 概述 42
3.1.1 Hadoop 的由来 42
3.1.2 Hadoop 的架构与扩展 43
3.1.3 Hadoop 的部署需求 45
3.2 HDFS 原理 46
3.2.1 HDFS 架构 46
3.2.2 Namenode 的数据结构 47
3.2.3 数据分块和多副本机制 48
3.2.4 数据读写原理 50
3.2.5 HDFS 支持的序列化文件 51
3.3 部署和配置 HDFS 52
3.3.1 部署 HDFS 53
3.3.2 HDFS 的基本配置 55

3.3.3 集群的启动和停止	57	4.8.3 数据更新	94
3.4 使用和管理 HDFS	58	4.8.4 数据查询	96
3.4.1 管理和操作命令	58	4.8.5 删除列和行	97
3.4.2 格式化 Namenode	59	4.8.6 过滤器的使用	98
3.4.3 Namenode 的安全模式	60	4.9 通过 Python 访问 HBase	99
3.4.4 元数据恢复	61	4.9.1 基于 Thrift 框架的多语言编程	99
3.4.5 子节点添加与删除	61	4.9.2 环境准备	100
3.4.6 HDFS 文件系统操作	62	4.9.3 代码分析	101
3.4.7 以 Web 方式查看文件系统	64	小结	104
3.5 MapReduce 原理简介	66	思考题	104
3.6 Hive 分布式数据仓库	68	第 5 章 HBase 的高级原理 ... 105	
小结	69	5.1 水平分区原理	106
思考题	69	5.1.1 META 表	106
第 4 章 HBase 的基本原理与		5.1.2 数据写入和读取机制	108
 使用	70	5.1.3 预写日志	110
4.1 HBase 概述	71	5.1.4 分区拆分	111
4.2 HBase 的数据模型	72	5.2 列族与 Store	113
4.3 HBase 的拓扑结构	74	5.2.1 列族的属性	113
4.4 HBase 部署与配置	75	5.2.2 表在 HDFS 上的存储	115
4.5 子节点伸缩性管理	78	5.2.3 HFile 的结构	116
4.6 HBase 的基本操作	79	5.2.4 Storefile 合并	119
4.6.1 HBase Shell	79	5.3 数据表的基本设计原则	121
4.6.2 表和列族操作	79	5.4 HBase 集群的高可用性与伸缩性 ... 121	
4.6.3 数据更新	82	5.4.1 Zookeeper 的基本原理	122
4.6.4 数据查询	83	5.4.2 基于 Zookeeper 的高可用性	124
4.6.5 过滤查询	85	5.4.3 独立安装 Zookeeper	125
4.6.6 快照操作	89	5.4.4 集群间同步复制	126
4.7 批量导入导出	89	5.5 HBase 的扩展	128
4.7.1 批量导入数据	89	5.5.1 协处理器机制	128
4.7.2 备份和恢复	91	5.5.2 基于 HBase 的分布式处理	129
4.8 通过 Java 访问 HBase	92	5.5.3 扩展开源软件	131
4.8.1 开发环境的配置	92	5.5.4 FusionInsight HD 简介	134
4.8.2 表的连接和操作	93	小结	135

思考题	136	6.7.2 多数据中心与机架感知策略	177
第 6 章 Cassandra 的原理和 使用	137	6.7.3 Nodetool 工具	178
6.1 Cassandra 概述	138	6.7.4 常见节点管理方法	180
6.2 Cassandra 的技术原理	138	6.8 编程访问 Cassandra	181
6.2.1 Amazon Dynamo	138	6.8.1 通过 Java 访问 Cassandra	181
6.2.2 Cassandra 的数据模型	143	6.8.2 通过 Python 访问 Cassandra	182
6.2.3 Yaml 格式	145	小结	183
6.2.4 其他相关技术原理	146	思考题	183
6.3 Cassandra 的部署与配置	147	第 7 章 MongoDB 的原理和 使用	184
6.3.1 单节点部署 Cassandra	147	7.1 概述	185
6.3.2 Cassandra 的配置文件	148	7.2 MongoDB 的技术原理	185
6.3.3 Cassandra 集群部署	150	7.2.1 文档和集合	185
6.3.4 集群启动	150	7.2.2 分片机制和集群架构	186
6.4 CQL 语言与 cqlsh 环境	151	7.2.3 CouchDB 简介	188
6.4.1 cqlsh 环境简介	152	7.3 安装配置 MongoDB	189
6.4.2 键空间管理	153	7.3.1 单机环境部署	189
6.4.3 数据表管理	156	7.3.2 MongoDB 的配置文件	191
6.4.4 CQL 的数据类型	158	7.4 基本命令行操作	191
6.5 CQL 数据查询	161	7.4.1 Shell 环境	191
6.5.1 基本数据查询	161	7.4.2 数据库和集合操作	193
6.5.2 条件查询	162	7.4.3 基本增删改查操作	194
6.5.3 索引机制	164	7.4.4 聚合和管道	196
6.5.4 使用标量函数	166	7.4.5 索引操作	198
6.6 CQL 数据更新	166	7.4.6 Gridfs 的原理和操作	201
6.6.1 插入、更新和删除	166	7.5 批量操作和数据备份	203
6.6.2 读写一致性	169	7.6 MongoDB 集群化部署	204
6.6.3 集合列操作	171	7.6.1 单机多实例	204
6.6.4 计数器列的操作	174	7.6.2 部署复制集	205
6.6.5 日期时间列的操作	175	7.6.3 部署分片集	209
6.6.6 批量导入/导出数据	176	7.7 通过 Java 访问 MongoDB	213
6.7 基本集群维护方法	177	7.7.1 表和数据操作	213
6.7.1 编程接口简介	177	7.7.2 Gridfs 操作	216

7.8 通过 Python 访问 MongoDB	218	8.4 搜索引擎系统	232
小结	220	小结	236
思考题	220	思考题	236
第 8 章 其他 NoSQL 数据库		附录	237
简介	221	附录 1 在 CentOS 7 上实现 SSH	
8.1 图数据库简介	222	无密码访问	237
8.2 Neo4j 的安装与使用	223	附录 2 在 CentOS 7 上部署 NTP	
8.2.1 在 Windows 中安装 Neo4j	223	服务端与客户端	239
8.2.2 在 CentOS 7 中安装 Neo4j	224	附录 3 在 CentOS 7 上安装	
8.2.3 Neo4j 的 Web 操作界面	225	Python 3	240
8.2.4 Cypher 语言简介	226	附录 4 在 CentOS 7 上安装 Thrift	
8.2.5 通过 Java 访问 Neo4j	229	编译器	241
8.2.6 通过 Python 访问 Neo4j	230	附录 5 《NoSQL 数据库原理》配套	
8.3 Redis 和内存数据库	231	实验课程方案简介	242

人类社会的进步离不开对信息数据的管理和使用。早期人们利用人工方式或文件方式来管理信息数据。20世纪60年代,数据库以及数据库管理系统等概念相继出现,随后,关系型数据库出现并逐渐成熟,成为被最广泛应用的数据库管理系统。如今提到“数据库”这个名词时,很多情况就是默认指关系型数据库。

随着互联网、大数据等概念的兴起,关系型数据库也暴露出问题,例如,难以应对日益增多的海量数据,横向的分布式扩展能力比较弱等。因此,有人通过打破关系型数据库的模式,构建出非关系型数据库,其目的是为了构建一种结构简单、分布式、易扩展、效率高且使用方便的新型数据库系统,这就是所谓的NoSQL。如今NoSQL数据库在互联网、电信、金融等行业已经得到了广泛应用,和关系型数据库形成了一种技术上的互补关系。

为了更好地理解NoSQL的出现原因、基本特点和适用场景,本章首先从数据库和数据业务的相关概念进行介绍,进而对大数据的概念、现状和技术体系等方面进行讲解。

1.1 数据库的相关概念

数据库 (Database, DB) 一般指数据信息的集合, 也可以看作按照数据结构来存储和组织信息数据的软件容器或仓库。数据库及其管理软件构成了数据库管理系统 (Database Management System, DBMS), 实现数据的管理和使用等功能。数据库管理系统及其运行的软硬件环境、操作人员乃至手册文档等内容, 构成一个完整体系, 称为数据库系统 (Database System, DBS)。

在更多的场景中, “数据库系统”的概念, 可以看作数据库管理系统的简称。而“数据库”的概念, 既可代指数据库管理系统, 也可以代指数据库管理系统下的信息数据集, 甚至可以代指对数据库相关技术和研究领域的统称。

本书提到“关系型数据库”和“非关系型数据库”(NoSQL), 实际都是数据库管理系统的的形式, 用来管理不同特点的数据, 以及用来支撑不同的业务逻辑。而本书讨论的重点问题, 就是非关系型数据库管理系统的原理和使用。

利用数据库可以构建各类应用——数据库应用系统 (Database Application System, DBAS) 就是指在数据库管理系统的支撑下建立的计算机应用系统。根据需求的不同, 人们对数据的利用和处理方式可能各有不同, 例如有些偏重数据查询, 有些偏重数据处理, 有些偏重通过探寻数据的规律对未来进行预测等, 本节主要对数据库和数据应用中的基本概念和常见名词进行介绍。

1.1.1 关系型数据库管理系统

一般来说, 数据库管理系统一般会提供数据的定义、操作、组织、存储和管理, 以及数据库的通信、管理、控制和保护等功能, 具体包括以下功能。

(1) 数据定义: 提供数据定义语言 (Data Definition Language, DDL), 用于建立、修改数据库的库、表结构或模式, 将结构或模式信息存储在数据字典 (Data Dictionary) 之中。

(2) 数据操作: 提供数据操作语言 (Data Manipulation Language, DML), 用于增加 (Create)、查询 (Retrieve)、更新 (Update) 和删除 (Delete) 数据 (合称 CRUD 操作)。

(3) 数据的持久存储、组织和维护、管理: 能够分类组织、存储和管理各种数据, 可以实现数据的加载、转换、重构等工作; 能够将大量数据进行持久化存储, 能够监控数据库的性能。

(4) 保护和控制: 可以支持多用户对数据并发控制, 支持数据库的完整性、安全性, 支持从故障和错误中恢复数据。

(5) 通信与交互接口: 可以实现高效存取数据 (例如查询和修改数据), 可以实现数据库与其他软件、数据库的通信和互操作等功能。

历史上第一批商用的数据库管理系统诞生于 20 世纪 60 年代, 这些系统一般基于层次模型或网状模型: 例如 IBM 公司研发的、基于网状模型的信息管理系统 (Information Management System, IMS) 等。这些数据库系统的出现, 改变了过去采用人工方式或文件方式管理数据的落后方式, 提高了数据管理的效率, 实现了更强大的数据访问方法, 以及提供了细粒度的数据定义和操作方法, 但这些数据模型需要使用者 (通常是程序员) 对数据格式具有深入的了解, 且不支持高级语言, 这使得数据操作的难度很大。

1970 年, IBM 公司圣何塞实验室的埃德加·弗兰克·科德 (Edgar Frank Codd, 1923-2003) 发