

NoSQL

数据库技术实战

皮雄军 编著

腾讯公司前资深后台工程师呕心沥血之作，大数据时代开发者必读佳品
全面介绍NoSQL从基本操作到高级技术和核心原理，再到项目开发的所有重点知识

- ✓ 详细讲解了NoSQL的兴起原因、CAP理论、副本之间数据一致性的解决方法和水平扩展方法，并介绍了NoSQL的分类、使用场景及其在实际开发中的技术选型
- ✓ 既对NoSQL系统的理论做了广泛而深入的分析，又重点对Hadoop/HBase、MongoDB和Redis这三种NoSQL系统给出了具体实践
- ✓ 注重实战，对三种NoSQL类型的开源系统的实现给出了详细的解释，并提供了实际可运行的程序，还对其优化做了丰富的阐述

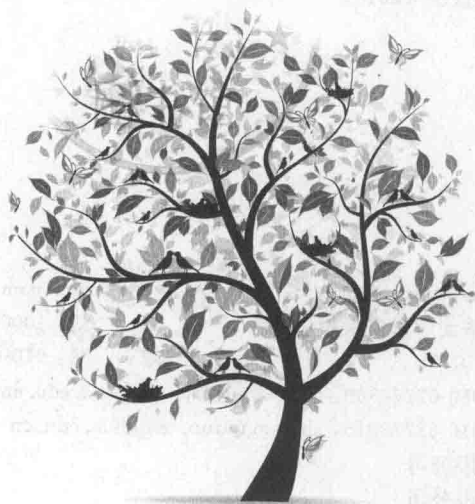


清华大学出版社

NoSQL

数据库技术实战

皮雄军 编著



内 容 简 介

本书由浅入深,全面系统地介绍了 NoSQL 系统。本书既对 NoSQL 系统的理论进行了深入浅出的分析,又介绍了每一种 NoSQL 数据库在业界广泛应用的一个具体系统,理论与实践并重。

本书共分 5 篇,12 章。涵盖的内容有:NoSQL 与大数据简介、NoSQL 的数据一致性、NoSQL 的水平扩展与其他基础知识、BigTable 与 Google 云计算原理、Google 云计算的开源版本——Hadoop、Dynamo: Amazon 的高可用键值对存储、LevelDb——出自 Google 的 Key-Value 数据库、Redis 实战、面向文档的数据库 CouchDB、MongoDB 实战、MySQL 基础、MySQL 高级特性与性能优化。

本书涉及面广,从基本操作到高级技术和核心原理,再到项目开发,几乎涉及 NoSQL 系统的所有重要知识。本书适合所有想全面学习 NoSQL 的人员阅读,也适合各种使用 NoSQL 进行开发的工程技术人员使用。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

NoSQL 数据库技术实战 / 皮雄军编著. —北京:清华大学出版社, 2014
ISBN 978-7-302-38039-9

I. ①N… II. ①皮… III. ①数据库系统 IV. ①TP311.138

中国版本图书馆 CIP 数据核字(2014)第 219814 号

责任编辑:夏兆彦
封面设计:欧振旭
责任校对:徐俊伟
责任印制:何 芊

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座

邮 编:100084

社总机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印刷者:三河市君旺印务有限公司

装订者:三河市新茂装订有限公司

经 销:全国新华书店

开 本:185mm×260mm

印 张:25.5

字 数:637 千字

版 次:2015 年 1 月第 1 版

印 次:2015 年 1 月第 1 次印刷

印 数:1~3500

定 价:69.00 元



前言

“数据是 21 世纪最有价值的资产，它比黄金和石油更有价值”。随着大数据时代的来临，传统的关系型数据库在可扩展性、数据模型和可用性方面遇到了难以克服的障碍。此时各种 NoSQL 系统出现了。它们的特点各不相同，分别应用于不同的场景并迅速取得了巨大的成功。作为一名从事后台开发多年的工作者，我对每一种新技术的出现与应用都充满了渴望与期待，其中 NoSQL 解决了我实际工作中遇到的许多问题。NoSQL 具有下面几方面的优点：

1. 灵活的可扩展性

多年以来，数据库管理员们都是通过“垂直扩展”的方式（当数据库的负载增加的时候，购买更大型的服务器来承载增加的负载）来进行扩展的，而不是通过“水平扩展”的方式（当数据库负载增加的时候，在多台主机上分配增加的负载）来进行扩展。但是，随着请求量和可用性需求的增加，数据库也正在迁移到云端或虚拟化环境中，“水平扩展”的经济优势变得更加明显了，对各大企业来说，这种“诱惑”是无法抗拒的。

要对 RDBMS（关系型数据库，比如 Oracle）做“水平扩展”，并不是很容易。但是各种新类型的 NoSQL 数据库主要是为了进行透明的扩展，来利用新节点而设计的，而且，它们通常都是为了低成本的 commodity hardware 而设计的。

2. 轻松应对海量数据

在过去的十年里，正如请求量发生了翻天覆地的增长一样，需要存储的数据量也发生了急剧的膨胀。为了满足数据量增长的需要，RDBMS 的容量也在日益增加，但是，对一些企业来说，随着请求量的增加，单一数据库能够管理的数据量也变得越来越让人无法忍受了。现在，大量的“大数据”可以通过 NoSQL 系统（如 MongoDB）来处理，它们能够处理的数据量远远超出了最大型的 RDBMS 所能处理的极限。

3. 维护简单

在过去的几年里，虽然一些 RDBMS 供应商们声称在可管理性方面做出了很多的改进，但是高端的 RDBMS 系统维护起来仍然十分昂贵，而且还需要训练有素的 DBA 们的协助。DBA 们需要亲自参与高端的 RDBMS 系统的设计、安装和调优。

NoSQL 数据库从一开始就是为了降低管理方面的要求而设计的：从理论上来说，自动修复，数据分配和简单的数据模型的确可以让管理和调优方面的要求降低很多。

4. 经济

NoSQL 数据库通常使用廉价的 Commodity Servers 集群来管理膨胀的数据和请求量，

而 RDBMS 通常需要依靠昂贵的专有服务器和存储系统来做到这一点。使用 NoSQL, 每 GB 的成本或每秒处理的请求的成本都比使用 RDBMS 的成本少很多, 这可以让企业花费更低的成本存储和处理更多的数据。

5. 灵活的数据模型

对于大型的生产性 RDBMS 来说, 变更管理是一件很令人头痛的事情。即使只对一个 RDBMS 的数据模型做出很小的改动, 也必须要十分小心的管理, 也许还需要停机或降低服务水平。NoSQL 数据库在数据模型约束方面是更加宽松的, 甚至可以说并不存在数据模型的约束。NoSQL 的 Key/Value 数据库和文档型数据库可以让应用程序在一个数据元素里存储任何结构的数据。即使是规定更加严格的基于“大表”的 NoSQL 数据库(如 HBase)通常也允许创建新列, 这并不会造成什么麻烦。

应用程序变更和数据库模式的变更并不需要作为一个复杂的变更单元来管理。从理论上来说, 这可以让应用程序迭代的更快, 但是, 很明显, 如果应用程序无法维护数据的完整性, 那么这也会带来一些不良的副作用。

本书的诞生

在当前的图书市场上, 还没有一本全面而深入介绍 NoSQL 系统的图书。要么只有理论内容并且大部分并不全面, 要么千篇一律把某个 NoSQL 系统的操作一一罗列。为了让众多的 NoSQL 学习人员能够更加全面和深入地学习 NoSQL 技术, 笔者编写了本书。本书从系统的角度出发, 既深入讲解 NoSQL 的产生原因与理论基础, 又对三种典型 NoSQL 系统引入了丰富的实战, 使读者可以全面而深入地了解各种 NoSQL, 知道各个 NoSQL 和 MySQL 系统的联系和不同, 在实际中根据自己的情况进行正确的技术选型。

本书特色

1. 内容全面、新颖

本书内容全面而新颖, 既对 NoSQL 系统的理论进行了深入浅出的分析, 又深入讲解了列式的、Key/Value 的和文档类型这三种 NoSQL 系统, 最后还讲解了 MySQL 的性能优化。

2. 讲解由浅入深, 循序渐进

本书是一本入门图书, 如果你从来没有用过 NoSQL 系统, 那么本书正好适合你。本书也是一本深入讲解 NoSQL 的图书, 它将各种 NoSQL 系统联系起来并分析各自的相同点和不同点, 读完之后你一定会对 NoSQL 系统有一个高屋建瓴的认识。

3. 理论与实践并重

本书对 NoSQL 系统的产生原因和理论基础做了广泛而深入的分析, 让读者知其然, 更知其所以然。书中尽力消除初学者学习 NoSQL 系统时容易遇到的障碍, 变抽象为具体,

变复杂为简单。而且重点对 Hadoop/HBase、MongoDB 和 Redis 这三种 NoSQL 系统都给出了具体实践。

4. 图文并茂，容易理解

本书针对 NoSQL 系统中的一些架构和较难理解的概念，提供了大量的插图，并结合具体文字来讲解，非常直观，更有利于读者的学习与理解。

5. 语言通俗易懂

本书不使用那些让人难以理解的语言来分析问题，而是采用通俗易懂的语言去讲解 NoSQL 系统的相关知识，让读者能够真正了解 NoSQL 技术，继而在开发中使用这些技术。

本书内容

第1篇 NoSQL的兴起与理论基础（第1~3章）

本篇介绍了大数据时代 NoSQL 的兴起原因及 NoSQL 的理论基础，包括 NoSQL 与大数据简介，NoSQL 系统的分类和特点，各种数据一致性模型的定义、原理、实现、举例及适用场景，NoSQL 系统水平扩展的方法，主要是复制和分区技术的难点、在实际系统中的运用及和数据一致性的关系，最后简单介绍了其他杂项知识比如五分钟法则等。这些内容都是学习后续章节所必须要掌握的基础知识，后续章节中的实际 NoSQL 系统如 Redis 可以看成是这些理论知识的运用。

第2篇 列式NoSQL系统（第4、5章）

本篇主要介绍了列式 NoSQL 系统。首先以 Google 的 BigTable 为例介绍了列式 NoSQL 系统的特点、原理和应用。然后实战演练开源的 HBase 系统。由于 HBase 与 Hadoop、Zookeeper 等有着十分密切的关系，因此对其一并进行了介绍。

第3篇 Key/Value NoSQL系统（第6~8章）

本篇主要介绍了 Key/Value NoSQL 系统。首先以 Amazon 的 Dynamo 为例介绍了 Key/Value 系统的特点、原理和应用场景，并详细描述了该系统是如何运用本书第 1 篇中的理论知识而构建的。然后介绍了另一个来自 Google 的 LevelDB 系统，最后实战演练了被广泛使用的 Redis 系统。

第4篇 文档型 NoSQL系统（第9、10章）

本篇主要介绍了文档型 NoSQL 系统。首先以 CouchDB 为例介绍了文档型 NoSQL 的特点、原理和应用场景，然后实战演练了被广泛使用的 MongoDB 系统。

第5篇 MySQL基础与性能优化（第11、12章）

本篇主要介绍了目前在互联网公司被广泛使用的 MySQL 关系型数据库。虽然在大数据时代 NoSQL 将会占据数据处理技术的主流，但是传统的 MySQL 在一些应用场景仍有着

自己的优势。所以本篇开始先介绍了一些 MySQL 的基础知识，然后详细描述了其高级特性，最后介绍了 MySQL 的性能优化、复制技术、垂直扩展、水平扩展和综合应用。

本书读者对象

- ❑ 没有基础的 NoSQL 初学者；
- ❑ 互联网公司高并发系统的后台开发人员；
- ❑ 大数据、NoSQL 开发爱好者；
- ❑ 刚入职的初中级程序员；
- ❑ 高等院校师生；
- ❑ 相关培训班的学员。

本书源程序获取方式

本书涉及的源程序与资源请读者自行到 www.tup.com.cn 上搜索到本书页面后按提示下载，也可以到 www.wanjuanchna.net 上的相关版块下载。

本书作者

本书由皮雄军主笔编写。其他参与编写的人员有吴万军、项延铁、谢邦铁、许黎民、薛在军、杨佩璐、杨习伟、于洪亮、张宝梅、张功勤、张建华、张建志、张敬东、张倩、张庆利、赵剑川、赵薇、郑强、周静、朱盛鹏、祝明慧、张晶晶。

虽然笔者花费了大量精力写作本书，并力图将疏漏减少到最少，但仍恐百密一疏。如果您在阅读本书的过程中发现有任何疏漏，或者对本书的讲解有任何疑问，发送电子邮件到 bookservice2008@163.com 以获得帮助。

编者

目 录

第 1 篇 NoSQL 的兴起与理论基础

第 1 章 NoSQL 与大数据简介	2
1.1 引子——NoSQL 在国内使用的案例	2
1.1.1 新浪微博	2
1.1.2 淘宝数据平台	3
1.1.3 视觉中国网站	4
1.1.4 优酷运营数据分析	5
1.1.5 飞信空间	6
1.1.6 豆瓣社区	7
1.2 大数据	8
1.2.1 大数据的度量单位	8
1.2.2 大数据的特点	9
1.3 大数据相关技术	10
1.3.1 大数据采集技术	10
1.3.2 大数据预处理技术	11
1.3.3 大数据存储及管理技术	11
1.3.4 大数据分析 & 挖掘技术	11
1.3.5 大数据展现与应用技术	12
1.4 NoSQL 简介	12
1.4.1 什么是 NoSQL	12
1.4.2 关系型数据库简史	13
1.4.3 数据库分类	13
1.4.4 关系型数据库的优势	14
1.4.5 不擅长的处理	15
1.4.6 NoSQL 数据库	18
1.5 NoSQL 数据库的类型	20
1.5.1 键值 (Key/Value) 存储	20
1.5.2 面向文档的数据库	21
1.5.3 面向列的数据库	22
1.6 如何使用和学习 NoSQL 数据库	23

1.6.1	始终只是一种选择	23
1.6.2	在何种程度上信赖它	23
1.7	云数据管理	24
第 2 章	NoSQL 的数据一致性	25
2.1	传统关系数据库中的 ACID	25
2.1.1	原子性	25
2.1.2	一致性	25
2.1.3	隔离性	26
2.1.4	持久性	26
2.1.5	举例	26
2.2	CAP 理论	27
2.2.1	NoSQL 系统是分布式系统	27
2.2.2	CAP 理论阐述	27
2.3	AP 的例子——DNS 系统	29
2.3.1	DNS 系统	29
2.3.2	DNS 域名解析过程	29
2.3.3	DNS 系统是最终一致性的	31
2.4	数据一致性模型与 BASE	31
2.4.1	数据一致性模型	31
2.4.2	BASE (Basically Available, Soft-state, Eventual consistency)	32
2.5	数据一致性实现方法	33
2.5.1	Quorum 系统 NRW 策略	33
2.5.2	时间戳策略	34
2.5.3	向量时钟	37
第 3 章	NoSQL 的水平扩展与其他基础知识	41
3.1	所有数据存放在一个服务器上	41
3.2	分片 (Sharding)	41
3.3	主从复制	43
3.4	对等 (Peer To Peer) 复制	44
3.5	复制和分片的同时使用	45
3.6	数据水平扩展的方法总结	46
3.7	分片对数据的划分方式	46
3.7.1	Range-Based Partitioning	47
3.7.2	Round-Robin	49
3.8	一致性 hash 算法 (Consistent Hashing)	50
3.8.1	基本场景	51
3.8.2	hash 算法和单调性	51
3.8.3	Consistent Hashing 算法的原理	51
3.8.4	虚拟节点	54

3.9	磁盘的读写特点及五分钟法则	55
3.9.1	磁道、扇区、柱面和磁头数	56
3.9.2	固态硬盘 (SSD)：随机读写速度快	57
3.9.3	内存：读写速度极快	57
3.9.4	五分钟法则	57
3.10	不要删除数据	58
 第 2 篇 列式 NoSQL 系统 		
第 4 章	BigTable 与 Google 云计算原理	60
4.1	云计算	60
4.1.1	云计算的概念	60
4.1.2	云计算发展现状	63
4.1.3	云计算实现机制	64
4.1.4	网格计算与云计算	66
4.2	Google 文件系统 GFS	68
4.2.1	系统架构	69
4.2.2	容错机制	71
4.2.3	系统管理技术	72
4.3	并行数据处理 MapReduce	73
4.3.1	产生背景	73
4.3.2	编程模型	73
4.3.3	实现机制	74
4.4	分布式锁服务 Chubby	78
4.4.1	Paxos 算法	78
4.4.2	Chubby 系统设计	79
4.4.3	Chubby 文件系统	80
4.4.4	通信协议	82
4.4.5	正确性与性能	83
4.5	分布式结构化数据表 BigTable	84
4.5.1	设计动机与目标	85
4.5.2	数据模型	85
4.5.3	系统架构	87
4.5.4	主服务器	88
4.5.5	子表服务器	89
4.5.6	性能优化	92
第 5 章	Google 云计算的开源版本——Hadoop	94
5.1	Hadoop 简介	94
5.1.1	Hadoop 发展史	94

5.1.2	Apache Hadoop 项目及体系结构	96
5.2	Hadoop 产生的原因	97
5.2.1	海量的数据	97
5.2.2	数据的存储和分析	98
5.3	Hadoop 和其他系统的比较	99
5.3.1	和关系型数据库管理系统 (RDBMS) 的比较	99
5.3.2	和网格计算的比较	100
5.4	HDFS 的架构设计	101
5.4.1	前提和设计目标	102
5.4.2	Namenode 和 Datanode	102
5.4.3	文件系统的 Namespace	103
5.4.4	数据复制	103
5.4.5	文件系统元数据的持久化	104
5.4.6	通讯协议	105
5.4.7	健壮性	105
5.4.8	数据组织	106
5.4.9	可访问性	107
5.4.10	空间的回收	107
5.5	安装 Hadoop	108
5.5.1	安装 JDK 1.7	108
5.5.2	安装 Hadoop	109
5.6	HDFS 操作	113
5.6.1	使用 FS Shell 命令操作 HDFS	113
5.6.2	编程读写 HDFS	118
5.7	Hadoop 中的 MapReduce 模型	119
5.7.1	MapReduce 计算模型	119
5.7.2	Hadoop 中的 Hello World 程序	120
5.7.3	运行 MapReduce 程序	124
5.7.4	Hadoop 中的 Hello World 程序——新的 API	125
5.7.5	MapReduce 的数据流和控制流	127
5.8	Zookeeper	129
5.8.1	Zookeeper 配置安装	129
5.8.2	Zookeeper 的数据模型	130
5.8.3	Zookeeper 的基本使用	131
5.8.4	ZooKeeper 典型的应用场景	133
5.8.5	统一命名服务 (Name Service)	133
5.8.6	共享锁 (Locks)	136
5.8.7	队列管理	137
5.8.8	Zookeeper 总结	139
5.9	HBase	139

5.9.1	简介	139
5.9.2	逻辑视图	139
5.9.3	物理存储	141
5.9.4	系统架构	144
5.9.5	关键算法/流程	146
5.10	HBase 的安装和配置	149
5.11	HBase 使用例子	151

第 3 篇 Key/Value NoSQL 系统

第 6 章	Dynamo: Amazon 的高可用键值对存储	156
6.1	简介	156
6.2	背景	157
6.2.1	系统假设和要求	158
6.2.2	服务水平协议 (SLA)	158
6.2.3	设计考虑	160
6.3	系统架构	161
6.3.1	系统接口	162
6.3.2	划分算法	162
6.3.3	复制	163
6.3.4	版本的数据	163
6.3.5	执行 get()和 put()操作	165
6.3.6	故障处理: 暗示移交 (Hinted Handoff)	166
6.3.7	处理永久性故障: 副本同步	167
6.3.8	会员和故障检测	167
6.3.9	添加/删除存储节点	169
6.4	实现	169
6.5	Amazon 使用的经验与教训	170
6.5.1	平衡性能和耐久性	171
6.5.2	确保均匀的负载分布	172
6.5.3	不同版本: 何时以及有多少	175
6.5.4	客户端驱动或服务器驱动协调	176
6.5.5	权衡后台和前台任务	177
6.5.6	讨论	177
6.6	结论	178
第 7 章	LevelDb——出自 Google 的 Key-Value 数据库	179
7.1	LevelDb 简介	179
7.2	LevelDb 的静态部分	180

7.2.1	整体架构	180
7.2.2	log 文件	182
7.2.3	SSTable 文件	183
7.2.4	MemTable 详解	186
7.3	LevelDb 的动态部分	187
7.3.1	写入与删除记录	187
7.3.2	读取记录	188
7.3.3	Compaction 操作	190
7.3.4	LevelDb 中的 Cache	192
7.3.5	Version、VersionEdit 和 VersionSet	194
第 8 章	Redis 实战	195
8.1	Redis 安装与准备	195
8.1.1	下载与安装	195
8.1.2	配置文件修改	196
8.1.3	启动 Redis	197
8.1.4	停止 Redis	198
8.2	Redis 所支持的数据结构	198
8.2.1	String	199
8.2.2	List	203
8.2.3	Set	207
8.2.4	Hash/哈希/散列	210
8.2.5	有序集合/Zset	213
8.3	Key 操作命令	218
8.3.1	概述	218
8.3.2	命令示例	220
8.4	事物	223
8.4.1	事物概述	223
8.4.2	相关命令	223
8.4.3	命令示例	224
8.4.4	WATCH 命令和基于 CAS 的乐观锁	225
8.5	Redis 的主从复制	226
8.5.1	Redis 的 Replication	226
8.5.2	Replication 的工作原理	227
8.5.3	如何配置 Replication	227
8.5.4	应用示例	228
8.6	Redis 的持久化	228
8.6.1	持久化机制	228
8.6.2	RDB 机制的优势和劣势	229
8.6.3	AOF 机制的优势和劣势	229

8.6.4 其他	230
8.7 Redis 的虚拟内存	231
8.7.1 简介	231
8.7.2 应用场景	231
8.7.3 配置	231
8.8 pipeline/管线	233
8.8.1 请求应答协议和 RTT	233
8.8.2 管线 (pipelining)	233
8.8.3 Benchmark	234
8.9 实例	234

第 4 篇 文档型 NoSQL 系统

第 9 章 面向文档的数据库 CouchDB	240
9.1 CouchDB 介绍	240
9.1.1 基本概念	240
9.1.2 扩展概念	241
9.2 CouchDB 安装与配置	241
9.3 REST API	242
9.3.1 数据库 REST API	243
9.3.2 文档 REST API	243
9.3.3 视图 REST API	243
9.3.4 附件 REST API	244
9.4 为应用建模	244
9.4.1 描述实体	244
9.4.2 描述一对一和一对多关系	245
9.4.3 描述多对多关系	246
9.5 实战开发	247
9.5.1 开发 Web 应用	247
9.5.2 使用 CouchDB jQuery 插件	248
9.5.3 示例应用建模	249
9.5.4 管理文档	250
9.5.5 视图	253
9.6 高级话题	259
9.6.1 权限控制与安全	259
9.6.2 文档更新校验	259
9.6.3 分组	259
9.6.4 键的排序	260

第 10 章 MongoDB 实战	261
10.1 为什么要使用 MongoDB	261
10.1.1 不能确定的表结构信息	261
10.1.2 序列化可以解决一切问题吗	261
10.1.3 无需定义表结构的数据库	262
10.2 MongoDB 的优势和不足	262
10.2.1 无表结构	262
10.2.2 容易扩展	263
10.2.3 丰富的功能	263
10.2.4 性能卓越	264
10.2.5 简便的管理	264
10.2.6 MongoDB 的不足	264
10.3 基本概念	264
10.4 Linux 下 MongoDB 的安装和配置、启动和停止	265
10.4.1 下载	265
10.4.2 安装	266
10.4.3 启动数据库	267
10.4.4 停止数据库	269
10.5 创建、更新及删除文档	270
10.5.1 连接数据库	270
10.5.2 插入记录	270
10.5.3 <code>_id</code> key	272
10.5.4 修改记录	272
10.5.5 删除记录	272
10.6 查询记录	273
10.6.1 普通查询	273
10.6.2 条件查询	274
10.6.3 <code>findOne()</code> 语法	274
10.6.4 通过 <code>limit</code> 限制结果集数量	274
10.7 高级查询	275
10.7.1 条件操作符	275
10.7.2 <code>\$all</code> 匹配所有	275
10.7.3 <code>\$exists</code> 判断字段是否存在	276
10.7.4 <code>Null</code> 值处理	276
10.7.5 <code>\$mod</code> 取模运算	276
10.7.6 <code>\$ne</code> 不等于	277
10.7.7 <code>\$in</code> 包含	277
10.7.8 <code>\$nin</code> 不包含	278
10.7.9 <code>\$size</code> 数组元素个数	278
10.7.10 正则表达式匹配	278

10.7.11	JavaScript 查询和\$where 查询	279
10.7.12	count 查询记录条数	279
10.7.13	skip 限制返回记录的起点	279
10.7.14	sort 排序	280
10.7.15	游标	280
10.8	MapReduce	281
10.8.1	Map	282
10.8.2	Reduce	282
10.8.3	Result	282
10.8.4	Finalize	283
10.8.5	Options	283
10.9	索引	284
10.9.1	基础索引	284
10.9.2	文档索引	285
10.9.3	组合索引	285
10.9.4	唯一索引	285
10.9.5	强制使用索引	286
10.9.6	删除索引	287
10.10	性能优化	287
10.10.1	explain 执行计划	287
10.10.2	优化器 Profile	288
10.10.3	性能优化举例	289
10.11	性能监控	290
10.11.1	mongosniff	291
10.11.2	Mongostat	292
10.11.3	db.serverStatus	292
10.11.4	db.stats	294
10.11.5	第三方工具	294
10.12	Replica Sets 复制集	294
10.12.1	部署 Replica Sets	295
10.12.2	主从操作日志 oplog	297
10.12.3	主从配置信息	298
10.12.4	管理维护 Replica Sets	299
10.12.5	增减节点	301
10.13	Sharding 分片	305
10.13.1	建立 Sharding Cluster	306
10.13.2	管理维护 Sharding	308
10.14	Replica Sets 和 Sharding 的结合	313
10.14.1	创建数据目录	314
10.14.2	配置 Replica Sets	314

10.14.3	配置 3 台 Route Process	316
10.14.4	配置 Shard Cluster	316
10.14.5	验证 Sharding 正常工作	317

第 5 篇 MySQL 基础与性能优化

第 11 章	MySQL 基础	320
11.1	CentOS 6.5 下 MySQL 的安装	320
11.2	MySQL 基本命令	322
11.3	MySQL 数据类型	325
11.3.1	整型	325
11.3.2	浮点型	326
11.3.3	定点数	326
11.3.4	字符串 (char,varchar,xxxtext)	326
11.3.5	二进制数据	327
11.3.6	日期时间类型	327
11.3.7	数据类型的属性	327
11.4	创建数据库和表	328
11.5	检索表中的数据	331
11.6	多个表的操作	334
第 12 章	MySQL 高级特性与性能优化	338
12.1	MySQL Server 系统架构	338
12.1.1	逻辑模块组成	338
12.1.2	各模块工作配合	341
12.2	存储引擎	343
12.2.1	MySQL 存储引擎概述	343
12.2.2	MyISAM 存储引擎简介	344
12.2.3	Innodb 存储引擎简介	346
12.3	MySQL 中的锁定机制	347
12.3.1	MySQL 中锁定机制概述	347
12.3.2	合理利用锁机制优化 MySQL	349
12.4	索引与优化	350
12.4.1	选择索引的数据类型	351
12.4.2	索引入门	351
12.4.3	索引的类型	352
12.4.4	高性能的索引策略	353
12.4.5	索引与加锁	358
12.5	MySQL 的 MyISAM 和 Innodb 的 Cache 优化	359