

大数据应用与技术丛书

The Definitive Guide to MongoDB, Second Edition

MongoDB

大数据处理权威指南 (第2版)

David Hows

Eelco Plugge

[美]

Peter Membrey

著

Tim Hawkins

王肖峰

译

apress®



清华大学出版社

大数据应用与技术丛书

MongoDB 大数据处理

权威指南(第2版)

David Hows
[美] Eelco Plugge 著
Peter Membrey
Tim Hawkins
王肖峰 译

清华大学出版社

北 京

David Hows, Eelco Plugge, Peter Membrey, Tim Hawkins

The Definitive Guide to MongoDB: A Complete Guide to Dealing with Big Data Using MongoDB, Second Edition
EISBN: 978-1-4302-5821-6

Original English language edition published by Apress Media. Copyright © 2013 by Apress Media.
Simplified Chinese-Language edition copyright © 2015 by Tsinghua University Press. All rights reserved.

本书中文简体字版由 Apress 出版公司授权清华大学出版社出版。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。

北京市版权局著作权合同登记号 图字：01-2014-7360

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

MongoDB 大数据处理权威指南(第2版)/(美)豪斯(Hows, D.)等著;王肖峰译.

—北京:清华大学出版社, 2015

(大数据应用与技术丛书)

书名原文: The Definitive Guide to MongoDB: A Complete Guide to Dealing with Big Data Using MongoDB, Second Edition

ISBN 978-7-302-38366-6

I. M… II. ①豪… ②王… III. ①关系数据库系统—指南 IV. ①TP311.138-62

中国版本图书馆 CIP 数据核字(2014)第 247391 号

责任编辑:王 军 李维杰

装帧设计:牛艳敏

责任校对:邱晓玉

责任印制:刘海龙

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

印装者:北京密云胶印厂

经 销:全国新华书店

开 本:185mm×260mm

印 张:17.75

字 数:477 千字

版 次:2015 年 1 月第 1 版

印 次:2015 年 1 月第 1 次印刷

印 数:1~3000

定 价:49.80 元

产品编号:060831-01



译者序

随着大数据的不断发展，非关系型数据库已经变得越来越重要，相关的产品也都得到了飞速发展。而其中 MongoDB 更是佼佼者，作为高性能开源文档数据库，MongoDB 以敏捷、可扩展和对企业应用友好而著称。使用 MongoDB 的公司包括 Foursquare、Craiglist、迪士尼、SAP、Intuit、EA 等。

本人从 2013 年开始有幸能够接触到 MongoDB，并在实际工作中使用它。但是在学习使用过程中始终存在着一些困难，MongoDB 相关的书籍太少，尤其是中文方面的，并且对 MongoDB 的讲解也不够详尽。因此，如果想彻底了解和学习 MongoDB，就必须硬啃它的官方文档，这对于英文能力不佳的读者肯定也是难点。

所以这次清华大学出版社准备引进出版此书时，我感到非常高兴，也非常荣幸能够负责本书的翻译工作。能够为 MongoDB 的推广做出一点贡献，能帮助大家深入了解 MongoDB 的强大特性，我就感到非常满足了。

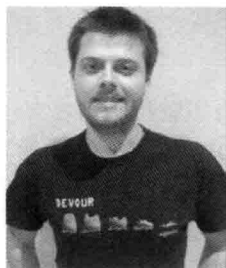
《MongoDB 大数据处理权威指南(第 2 版)》由浅入深地介绍 MongoDB 的基础知识、使用和高级功能，并且提供了丰富的样例，保证读者在读完本书之后能独立搭建起自己的开发环境。即便是对 NoSQL 毫不了解的读者，也可以顺利完成本书的阅读。对于有一定开发经验的读者，相信本书关于数据库优化、复制和分片方面的讲解也会让你感到耳目一新。总而言之，本书是学习 MongoDB 数据库的一本不可多得的精品之作。

另外，感谢清华大学出版社的编辑们为本书付出的心血。同样感谢妻子对我翻译工作的支持和鼓励。没有你们的支持和鼓励，本书就不可能顺利出版。

对于这本经典之作，译者对本书进行了严格审校，对其中一些具有争议的地方也进行了反复考证，但个人精力有限，难免有疏漏之处，敬请各位读者谅解。如有任何意见或建议，请不吝指正。本书全部章节由王肖峰翻译，参与翻译活动的还有杜欣、高国一、孙其淳、孙绍辰、徐保科、尤大鹏、袁强强、张宏、张立红、邓伟、王蕊。

最后，希望各位读者能够早日掌握强大的 MongoDB 特性，轻松地处理工作中遇到的问题，如原书作者所描述的：轻松工作，按时下班！

作者简介



David Hows 以优异的成绩毕业于澳大利亚新南威尔士州的卧龙岗大学。他第一次接触计算机，是在尝试不花钱的情况下改进家庭 PC 的性能。这促使他加入 IT 行业，David 曾经担任过系统管理员、性能工程师、软件开发者、解决方案架构师和数据库工程师等职务。David 也曾徒劳地尝试过多年足球，并且他的咖啡杯上写着“Grumble Bum”。



Peter Membrey 是一位特许 IT 研究员，他拥有 15 年使用 Linux 和开源解决方案解决现实中问题的经验。从 17 岁起他就是一位红帽认证工程师，并且他也有幸在 Red Hat 工作过，并编写了几本开源解决方案相关的书籍。他拥有利物浦大学的信息安全硕士学位，目前是香港理工大学的博士生，他的研究方向包括云计算、大数据、科学和安全。他与自己出色的妻子 Sara 和儿子 Kaydyn 一起居住在香港。尽管他的世界语一直不错，但广东话却在不断退步。



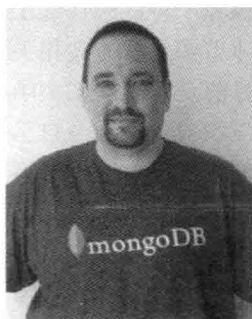
Eelco Plugge 是一位年轻的 IET/BSC 专家，他对 IT 安全领域非常感兴趣。在 21 岁时他就成为一位加密专家，目前从事移动设备管理行业，偶尔也会编写一些书籍。Eelco 最近拿到了利物浦大学的计算机安全硕士学位，并持有多项专业证书。他对 Linux、网络安全和加密技术都抱有极大的热情。Eelco 与他的年轻家庭一起居住在荷兰。他喜欢寿司，讨厌过于复杂的事情。

Tim Hawkins 曾在 1993 年创建了世界上第一个在线分类广告门户网站 loot.com，之后负责管理雅虎欧盟的许多非媒体属性产品，例如搜索、本地搜索、邮件、消息和社交网络。他目前正在管理美国主要电子零售商的一个大型离岸团队，负责开发和部署下一代电子商务应用。他喜欢帽子，讨厌复杂性。

技术编辑简介



Alexandre Beaulne 来自于加拿大的北部，他获得了蒙特利尔麦吉尔大学的神经系统学士学位，随后获得了蒙特利尔 HEC 的金融工程硕士学位。不过，他的研究在他参加了纽约的黑客学校之后达到了巅峰。他一直不断地在声明，表示他并未“明白”Haskell 的 monads¹。



Stephen Stenecker(又名 Stennie)是一位经验丰富的全能软件开发者、咨询师和讲师。Stephen 曾长期工作于澳大利亚的技术公司，是雅虎澳大利亚和新西兰、HomeScreen Entertainment 和 Grox 的技术负责人。他获得了英属哥伦比亚大学的学士学位(计算机科学)。

他目前的职位是 MongoDB Inc.的技术服务工程师，为 MongoDB 提供支持、咨询和培训。他经常在用户组和会议上发言，并且他也是悉尼 MongoDB 用户组的创始人和管理者(<http://www.meetup.com/SydneyMUG/>)。

在 Twitter、StackOverflow 或 Github 上可以找到他，他的 ID 为@stennie。

1. 译者注：Haskell 是一种编程语言，monads 则是该语言下的一种抽象数据类型。

致 谢

感谢 MongoDB 团队曾经和现有的所有成员。没有他们的帮助我们就无法完成本书的编写，人们对数据存储的观点也会出现极大的不同。我要特别感谢悉尼 MongoDB 团队的同事们，他们为本书的编写提供了极大的帮助。

——David Hows

编写书籍是一个团队性的工作。即使有时只有一个作者，在幕后也会有许多人帮助将所有东西整合在一起。因此我非常感谢 MongoDB 社区和 Apress 的所有人，感谢他们的辛勤工作、耐心和支持。特别要感谢 Dave 和 Eelco，他们帮助完成了本书的第 2 版。

我还想感谢 Chuck Smith，一位让我迷上世界语并让我保持有很高兴趣的朋友(维基百科世界语版本的创始人)。这正好与本书第 2 版的编写产生了交集，我对学习世界语的极大兴趣延伸到了本书的编写上。另外还必须感谢 L.L.Zamenhof 博士，他创建了世界语。一个远超前时代的人，我希望能带着这个梦想前进。

最后，感谢妻子 Sarah 的耐心和长久的支持。

——Peter Membrey

感谢 9gag 社区，没有他们本书就无法在数月前完成。

——Eelco Plugge

我想感谢 mongodb-user 和 mongodb-dev 邮件列表中的所有成员，感谢他们能够忍耐我提出的无休止的问题。

——Tim Hawkins

前言

我接触数据库的时间相对较晚，从 2006 年才开始使用 MySQL。在学完所有计算机本科都提供的逻辑课程之后，我开始使用 MySQL 构建出一个完整的 LAMP 架构，其中用到了一些基本表。此时，我并未对 SQL 表管理的复杂性进行深入思考。不过，随着时间的流逝，我看到了存储越来越多异构数据的需要，并且随着时间的推移，简单的模式可以如何在它的生命周期中成长和演变。

我第一次接触 MongoDB 是在 2011 年，当时 Peter Membrey 建议我不要使用包含 30 个键行以及 30 个值行的上下文表，而是应该使用 MongoDB 实例来存储数据。就像所有开发者在面对一项新技术时的感觉一样，我对之嗤之以鼻并且坚持我原来的计划。直到我使用糟糕的设计已经完成一半代码时，Peter 仍然坚持我应该尝试使用 MongoDB，此时我才接受意见。如同所有来自 SQL 阵营的开发者一样，MongoDB 能够接受任何类型数据并且可以根据任何搜索条件返回这些数据的能力让我感到震惊。直到现在我也仍然大呼过瘾。

——David Hows

本书的组织方式

在本书中，Peter、Eelco Plugge、Tim Hawkins 和我都希望能够完全地展示出我们在学习 MongoDB 时的经验：在保持设计简单和清晰的同时，教会你如何使用 MongoDB。每章内容都将展示一个单独的样例数据库，因此你可以按照模块或线性的方式阅读本书；这完全取决于你自己。这意味着如果愿意，你可以略过某些特定的章节，而不会影响你对其他内容的学习。

本书的样例命令将显示在它们的输出中。它们将以等宽的“代码”字体出现，并且会以加粗的方式显示，以便与其他结果输出加以区分。在大多数章节中，你都会遇到提示、警告和注意，它们包含有用的有时甚至极其重要的信息。

——David Hows

目 录

第 I 部分 MongoDB 基础

第 1 章 MongoDB 简介	3
1.1 了解 MongoDB 哲学	3
1.1.1 使用正确的工具处理正确的工作	3
1.1.2 天生缺少对事务的支持	5
1.1.3 JSON 和 MongoDB	5
1.1.4 采用非关系的方式	7
1.1.5 选择性能还是特性	8
1.1.6 在任何地方均可运行数据库	8
1.2 将所有组合在一起	9
1.2.1 生成或创建键	9
1.2.2 使用键和值	9
1.2.3 实现集合	10
1.2.4 了解数据库	10
1.3 了解特性列表	10
1.3.1 使用面向文档存储(BSON)	11
1.3.2 支持动态查询	11
1.3.3 为文档创建索引	12
1.3.4 使用地理空间索引	13
1.3.5 分析查询	13
1.3.6 就地更新信息	13
1.3.7 存储二进制数据	14
1.3.8 复制数据	14
1.3.9 实施分片	14
1.3.10 使用映射和归约函数	15
1.3.11 全新的聚合框架	15
1.4 获取帮助	15
1.4.1 访问网站	15
1.4.2 与 MongoDB 开发者沟通	16
1.4.3 剪切和粘贴 MongoDB 代码	16
1.4.4 在 Google 小组中寻找解决方案	16
1.4.5 利用 JIRA 跟踪系统	16
1.5 小结	16
第 2 章 安装 MongoDB	17
2.1 选择版本	17
2.2 在系统中安装 MongoDB	18

2.2.1 在 Linux 中安装 MongoDB	18
2.2.2 在 Windows 中安装 MongoDB	19
2.3 运行 MongoDB	20
2.3.1 先决条件	20
2.3.2 研究安装目录布局	20
2.3.3 使用 MongoDB shell	21
2.4 添加额外的驱动	22
2.4.1 安装 PHP 驱动	22
2.4.2 确认 PHP 安装正确	25
2.4.3 安装 Python 驱动	26
2.4.4 确认 PyMongo 安装正确	28
2.5 小结	29
第 3 章 数据模型	31
3.1 设计数据库	31
3.1.1 集合的更多细节	32
3.1.2 使用文档	33
3.1.3 在文档中内嵌或引用信息	34
3.1.4 创建_id 字段	35
3.2 构建索引	36
3.3 使用地理空间索引	37
3.4 在真实世界中使用 MongoDB	42
3.5 小结	42
第 4 章 使用数据	43
4.1 浏览数据库	43
4.2 在集合中插入数据	44
4.3 查询数据	45
4.3.1 使用点号	47
4.3.2 使用函数 sort、limit 和 skip	47
4.3.3 使用固定集合、自然顺序和 \$natural	48
4.3.4 获取单个文档	50
4.3.5 使用聚集命令	50
4.3.6 使用条件操作符	52
4.3.7 使用正则表达式	59
4.4 更新数据	59
4.4.1 使用 update() 更新	60

4.4.2	使用 save()命令实现 upsert	60	6.3	连接和断开连接	91
4.4.3	自动更新信息	60	6.4	插入数据	92
4.4.4	指定匹配数组的位置	64	6.5	查询数据	94
4.4.5	原子操作	65	6.5.1	返回单个文档	94
4.4.6	以原子的方式修改和返回文档	66	6.5.2	列出所有文档	95
4.5	重命名集合	67	6.5.3	使用查询操作符	96
4.6	删除数据	67	6.5.4	查询特定信息	96
4.7	引用数据库	68	6.5.5	排序、限制和忽略数据项	97
4.7.1	手动引用数据	68	6.5.6	统计匹配结果的数目	98
4.7.2	使用 DBRef 引用数据	70	6.5.7	使用聚集框架对数组分组	98
4.8	使用索引相关的函数	71	6.5.8	使用 hint()函数指定索引	99
4.8.1	学习索引相关的命令	72	6.5.9	使用条件操作符重新定义查询	100
4.8.2	强制使用某个索引查询数据	72	6.5.10	正则表达式	106
4.8.3	限制查询匹配	73	6.6	使用 PHP 修改数据	106
4.9	小结	74	6.6.1	使用 update()函数更新数据	106
第 5 章	GridFS	75	6.6.2	节省更新操作的时间	108
5.1	背景	75	6.6.3	使用 \$inc 增加特定键的值	108
5.2	使用 GridFS	76	6.6.4	使用 \$set 修改键值	109
5.3	开始使用命令行工具	76	6.6.5	使用 \$unset 删除字段	109
5.3.1	使用 _id 键	77	6.6.6	使用 \$rename 重命名字段	110
5.3.2	使用文件名	77	6.6.7	在更新/插入期间使用 \$setOnInsert 修改键值	110
5.3.3	决定文件长度	78	6.6.8	使用 \$push 向指定字段中添加值	111
5.3.4	使用块大小	78	6.6.9	使用 \$push 和 \$each 向某个键中 添加多个值	111
5.3.5	处理上传日期	79	6.6.10	使用 \$addToSet 将数据添加到 数组中	112
5.3.6	生成文件的哈希值	79	6.6.11	使用 \$pop 从数组中删除元素	113
5.4	查看 MongoDB 中的数据	79	6.6.12	使用 \$pull 删除所有指定值	113
5.4.1	使用搜索命令	81	6.6.13	同时删除多个元素	113
5.4.2	删除	81	6.6.14	使用 save()函数更新/插入数据	114
5.4.3	从 MongoDB 中获取文件	82	6.6.15	以原子的方式修改文档	115
5.4.4	mongofiles 命令小结	82	6.7	删除数据	116
5.5	使用 Python	82	6.8	DBRef	118
5.5.1	连接数据库	83	6.9	GridFS 和 PHP 驱动	120
5.5.2	访问单词	83	6.9.1	存储文件	120
5.6	在 MongoDB 中添加文件	84	6.9.2	在已存储的文件中添加元数据	121
5.7	从 GridFS 中读取文件	84	6.9.3	获取文件	121
5.8	删除文件	85	6.9.4	删除数据	122
5.9	小结	85	6.10	小结	122
第 II 部分	在开发中使用 MongoDB		第 7 章	Python 和 MongoDB	123
第 6 章	PHP 和 MongoDB	89	7.1	在 Python 中使用文档	123
6.1	比较 MongoDB 和 PHP 中的 文档	89	7.2	使用 PyMongo 模块	124
6.2	MongoDB 类	90			

7.3	连接和断开	124	8.3.5	调试 MapReduce	174
7.4	插入数据	125	8.4	小结	175
7.5	搜索数据	126	第III部分 使用MongoDB处理大数据		
7.5.1	搜索单个文档	126	第9章 数据库管理	179	
7.5.2	搜索多个文档	127	9.1	使用管理工具	179
7.5.3	使用点操作符	128	9.1.1	mongo——MongoDB 控制台	180
7.5.4	返回字段	128	9.1.2	使用第三方管理工具	180
7.5.5	使用 sort()、limit()和 skip()简化查询	129	9.2	备份 MongoDB 服务器	180
7.5.6	聚集查询	130	9.2.1	创建第一个备份	180
7.5.7	使用 hint()指定索引	132	9.2.2	备份单个数据库	182
7.5.8	使用条件操作符重定义查询	133	9.2.3	备份单个集合	182
7.5.9	使用正则表达式执行搜索	137	9.3	深入学习备份	183
7.6	修改数据	138	9.4	恢复单个数据库或集合	184
7.6.1	更新数据	139	9.4.1	恢复单个数据库	185
7.6.2	修改操作符	140	9.4.2	恢复单个集合	185
7.6.3	使用 save()函数快速保存文档	144	9.5	自动化备份	185
7.6.4	以原子的方式修改文档	145	9.5.1	使用本地数据存储	185
7.6.5	使用参数	145	9.5.2	使用远端数据存储(基于云)	187
7.7	删除数据	146	9.6	备份大数据库	188
7.8	在两个文档之间创建链接	146	9.6.1	使用隐藏的辅助服务器备份数据	189
7.9	小结	149	9.6.2	使用日志文件系统创建快照	189
第8章 高级查询	151		9.6.3	使用卷管理器时的磁盘布局	191
8.1	文本搜索	151	9.7	将数据导入 MongoDB	191
8.1.1	文本搜索的代价和限制	152	9.8	从 MongoDB 导出数据	192
8.1.2	启用文本索引	152	9.9	通过限制对 MongoDB 服务器的访问保护数据安全	194
8.1.3	使用文本索引	153	9.10	使用认证保护服务器	194
8.1.4	其他语言中的文本索引	158	9.10.1	添加 admin 用户	195
8.1.5	文本索引的复合索引	159	9.10.2	启用认证	195
8.2	聚集框架	161	9.10.3	在 mongo 控制台中执行认证	195
8.2.1	\$group	162	9.10.4	MongoDB 用户角色	196
8.2.2	\$sum	162	9.10.5	代理凭证	197
8.2.3	\$limit	163	9.10.6	修改用户凭证	197
8.2.4	\$match	164	9.10.7	添加只读用户	198
8.2.5	\$sort	165	9.10.8	删除用户	198
8.2.6	\$unwind	166	9.10.9	在 PHP 应用中认证连接	198
8.2.7	\$project	167	9.11	管理服务器	199
8.2.8	\$skip	168	9.11.1	启动服务器	199
8.3	MapReduce	170	9.11.2	重新配置服务器	200
8.3.1	MapReduce 的工作方式	170	9.11.3	获得服务器版本	201
8.3.2	创建测试文档	170	9.11.4	获得服务器状态	201
8.3.3	使用 map 函数	170			
8.3.4	高级 MapReduce	173			

9.11.5	关闭服务器	203	10.7	优化小对象的存储	229
9.12	使用 MongoDB 日志文件	203	10.8	小结	230
9.13	验证和修复数据	204	第 11 章	复制	231
9.13.1	修复服务器	204	11.1	MongoDB 复制的目标	231
9.13.2	验证单个集合	206	11.1.1	改善可扩展性	231
9.13.3	修复集合验证错误	207	11.1.2	改善持久性/可靠性	232
9.14	升级 MongoDB	209	11.1.3	提供隔离性	232
9.15	MongoDB 的滚动升级	209	11.2	复制基础	232
9.16	监控 MongoDB	209	11.2.1	主服务器的定义	233
9.17	使用 MongoDB 管理服务	211	11.2.2	辅助服务器的定义	233
9.18	小结	211	11.2.3	仲裁服务器的定义	234
第 10 章	优化	213	11.3	深入学习 oplog	234
10.1	优化服务器硬件	213	11.4	实现复制集	235
10.1.1	了解 MongoDB 使用内存的方式	213	11.4.1	创建复制集	236
10.1.2	了解工作集大小	214	11.4.2	启动服务器成员	237
10.1.3	选择正确的数据库服务器硬件	214	11.4.3	向复制集中添加服务器	238
10.2	评估查询性能	214	11.4.4	添加仲裁服务器	241
10.2.1	MongoDB 分析器	215	11.4.5	复制集链	242
10.2.2	使用 explain() 分析特定的查询	217	11.4.6	管理复制集	242
10.2.3	使用分析器和 explain() 优化查询	218	11.4.7	为复制集成员设置选项	246
10.3	管理索引	222	11.4.8	从应用连接到复制集	248
10.3.1	显示索引	222	11.5	小结	251
10.3.2	创建简单的索引	223	第 12 章	分片	253
10.3.3	创建复合索引	223	12.1	了解分片的需求	253
10.4	指定索引选项	224	12.2	对数据进行水平和垂直分区	254
10.4.1	使用 {background: true} 在后台创建索引	224	12.2.1	对数据进行垂直分区	254
10.4.2	使用 {unique: true} 创建唯一键索引	225	12.2.2	对数据进行水平分区	254
10.4.3	使用 {dropdups: true} 自动去重	225	12.3	分析一个简单的分片场景	255
10.4.4	使用 {sparse: true} 创建稀疏索引	226	12.4	使用 MongoDB 实现分片	256
10.4.5	TTL 索引	226	12.4.1	创建分片设置	258
10.4.6	文本索引	227	12.4.2	在集群中添加新的分片	260
10.4.7	删除索引	227	12.4.3	从集群中移除分片服务器	262
10.4.8	重建集合索引	227	12.4.4	确定连接的方式	263
10.5	MongoDB 选择索引的方式	228	12.4.5	列出分片服务器的状态	263
10.6	使用 hint() 强制使用特定的索引	229	12.4.6	使用复制集实现分片	264
			12.5	均衡器	264
			12.6	哈希片键	266
			12.7	标签分片	267
			12.8	小结	269

第 I 部分



MongoDB 基础

第 1 章 MongoDB 简介

第 2 章 安装 MongoDB

第 3 章 数据模型

第 4 章 使用数据

第 5 章 GridFS

第 1 章



MongoDB 简介

想象一下这样的世界：数据库使用是如此的简单，以至于你忘记了正在使用它。再想象一下这样的世界：不需要任何复杂配置或设置，数据库仍然能够快速运行，并且具有良好的扩展性。想一下，如何可以只关注 `gg` 于手上的任务，完成它，并可以按时下班。这听起来有点神奇，但是 MongoDB 承诺帮助你完成所有这些事情(甚至更多)。

MongoDB(源自单词 *humongous*)是一种相对较新的数据库，它没有表、模式、SQL 或行的概念。它没有事务、ACID 兼容性、连接、外键或其他许多容易在凌晨引起问题的特性。简单地说，MongoDB 是一个非常特别的数据库，它不同于你之前所使用的数据库，特别是关系数据库管理系统(Relational DataBase Management System, RDBMS)。事实上，当你知道 MongoDB 缺少对这些所谓的“标准”特性的支持时，你甚至可能已经在摇头了。

不要担心！在接下来的内容中，你将学习 MongoDB 的背景和指导原则，以及 MongoDB 团队这样做的原因。我们也将浏览 MongoDB 的特性列表，并提供足够的细节，从而保证你会完全沉迷于本书其余部分的话题中。

首先我们将了解创建 MongoDB 背后的哲学和理念，以及一些有趣的和有争议的设计决策。我们将学习面向文档数据库的概念、文档的组织方式以及它们的优缺点。我们还将学习 JSON 以及如何在 MongoDB 中应用 JSON。最后，我们将学习一些 MongoDB 最引人注目的特性。

1.1 了解 MongoDB 哲学

如同所有的项目一样，MongoDB 有一套自己的设计哲学用于帮助指导开发。本节内容将介绍一些 MongoDB 数据库的基本原则。

1.1.1 使用正确的工具处理正确的工作

MongoDB 中最重要的哲学概念是：一鞋难合众人脚。在过去的许多年中，传统的关系(SQL)数据库(MongoDB 是面向文档的数据库)一直被用于存储所有类型的数据。无论该数据是否符合关系模型(被用在所有的 RDBMS 数据库中，例如 MySQL、PostgreSQL、SQLite、Oracle、SQL Server 等)都无所谓；无论如何，数据都将被填充到数据库中。一般来说，部分原因是因为读取和修改数据库相比操作文件系统更加简单(和安全)。如果选择一本 PHP 方面的书籍，例如 *PHP for Absolute Beginners*, by Jason Lengstorf (Apress, 2009)，它将会教你使用数据库存储信息，而不是文件系统。这样做只是因为它更简单。在使用数据库存储信息的时候，开发者必须一直遵守它的工作流程。很明显，我们并未按照数据库原有的意图使用它。任何尝试在数据库中存储复杂数据、创建 5 张表，然后将它们组合在一起的开发人员，都会明白我们在讲什么。

MongoDB 团队决定他们不会创建另一个试图为所有人做所有事情的数据库。相反，该团队希望创建一个只用于处理文档的数据库，而不是行，并且它的速度要快，还要具有强大的扩展性和易用性。为了实现这个目标，他们不得不忽略一些特性，这意味着 MongoDB 在某些特定的情况下并不是最理想的选择。例如，它缺少事务支持，意味着无法使用 MongoDB 编写财务应用。也就是说，MongoDB 可能对于之前提到的部分应用(例如存储复杂数据)是非常合适的。不过这不是个问题，因为你完全可以在财务模块中使用传统的 RDBMS，而使用 MongoDB 存储文档。这样的混合解决方案非常常见，并且一些产品级应用(例如 New York Times 网站)已经在这样做了。

一旦适应 MongoDB 可能无法解决所有问题的理念之后，你会发现对于某些问题，MongoDB 可以完美地解决它们，例如分析(例如网站中使用的实时 Google Analytics)和复杂数据结构(例如博客文章和评论)。如果你仍然无法接受 MongoDB 是一个正式的数据库工具这个观点，那么请提前跳到 1.3 节“了解特性列表”，该部分内容将展示 MongoDB 的一些强大特性。

注意：

缺少事务和其他传统数据库特性并不意味着 MongoDB 不稳定，或者不能用于管理重要数据。

MongoDB 设计背后的另一个关键概念是：数据库应该一直具有多个副本。如果单个数据库实例出现问题，那么它可以轻松地通过另一个服务器恢复到正常状态。因为 MongoDB 的目标是尽可能地快，所以它采取了一些捷径，导致它难以从系统崩溃中恢复。开发者认为最严重的系统崩溃可能就是从服务中移除一台计算机；这意味着即使数据库完全恢复了，也无法正常使用。记住：MongoDB 不会尝试为所有人完成所有事情。但对于许多目的(例如构建 Web 应用)，MongoDB 是一个能够实现解决方案的完美工具。

现在你应该已经明白了 MongoDB 的起源。它不会尝试在所有方面都表现完美，也乐于承认它不会适用于所有人。不过，对于选择使用它的开发者，MongoDB 提供了一个功能丰富的面向文档数据库，并且对运行速度和扩展性做了优化。它也几乎可以运行在任何目标上。MongoDB 的网站上包含了可运行在 Linux、Mac OS、Windows 和 Solaris 中的安装文件。

MongoDB 成功实现了这些目标，因此使用 MongoDB 有点像梦幻一样(至少对于我们来说)。不必担心如何将数据压缩到一张表中，只需要将数据组合在一起，然后将数据传递给 MongoDB。考虑一个真实的例子。本书的合作者 Peter Membrey 最近开发了一个应用，用于存储一组 eBay 搜索结果。搜索结果的数量是不固定的(最多 100 个)，因此他需要一种简单的方式在数据库中将用户和搜索结果关联起来。

Peter 曾尝试使用 MySQL，他不得不设计出一张表用于存储数据，并编写相应代码存储他的结果，然后再编写代码将结果组合在一起。这是一个相当常见的场景，大多数开发者在开发中经常会遇到。通常，我们不得不这样做；不过，对于该项目，他决定使用 MongoDB，因此事情就变得有点不同了。

具体地说，他添加了下面这样两行代码：

```
request['ebay_results'] = ebay_results_array  
collection.save(request)
```

在本例中，request 是 Peter 的文档，ebay_result 是键，而 ebay_result_array 包含来自 eBay 的搜索结果。第二行保存了修改后的数据。将来当他访问该文档时，他将获得与之前格式完全相同的数据。他不需要任何 SQL；也不需要执行任何会话；更不需要创建任何新表或编写任何特殊代码——MongoDB 就可以完成工作。他最终轻松地完成了工作，并按时回家。

1.1.2 天生缺少对事务的支持

MongoDB 开发者做出的另一个重要设计决定是：该数据库将不会包含事务的语义(用于保证数据一致性和存储的元素)。这是根据 MongoDB 的目标——简单、快速和可扩展做出的权衡之计。一旦移除这些重量级特性，数据库就可以更轻松地实现水平扩展。

通常在使用传统的 RDBMS 时，如果需要提高性能，就要购买更大、更快的机器。这是一种垂直扩展的方式，但只能做到这种程度。如果使用了水平扩展，就不需要使用一台大型机器，相反可以使用许多稍弱一些小机器。从历史上讲，这样的服务器集群非常有利于负载均衡网站，但因为数据库的内部设计限制，这种方式一直存在着问题。

你可能会认为缺少事务的支持将构成致命缺陷；不过，许多人都忘了一种在 MySQL 中最流行的表类型(MYISAM——这也恰好是默认的表类型)也不支持事务。但这个事实并未阻止 MySQL 在之前的 10 年中变成并保持着主流开源数据库的地位。与开发解决方案时的大多数选择一样，是否使用 MongoDB 取决于个人的偏好，以及它是否符合你的项目。

注意：

MongoDB 在同时使用至少两台服务器时可以提供持久性，这也是为生产环境部署时推荐使用的基本配置。主服务器可以在复制服务器确认接收到数据后，再确认数据已被接受。

尽管目前单服务器的持久性是无法保证的，但在将来可能会改变，在目前这也是一个热门领域。

1.1.3 JSON 和 MongoDB

JSON(JavaScript Object Notation, JavaScript 对象记号)不止是一种交换数据的方式，它也是一种存储数据的良好方式。RDBMS 是高度结构化的，包含了多个文件(表)用于存储不同的数据。而 MongoDB 将在单个文档中存储所有的数据。MongoDB 如同 JSON 一样，该模型为数据存储提供了丰富和高效的方式。而且，JSON 可以高效地描述指定文档中的所有内容，所以不需要提前指定文档的结构。JSON 是没有模式的(不需要模式)，因为文档可以单独更新，或者独立于其他文档进行修改。另外，JSON 还通过将相关数据存储在同一位置的方式提供了出色的性能。

实际上，MongoDB 并未使用 JSON 存储数据，而使用由 MongoDB 团队开发的一种称为 BSON(二进制 JSON 的英文简称)的开放数据格式。大多数情况下，使用 BSON 取代 JSON 并不会改变处理数据的方式。BSON 通过使计算机更容易处理和搜索文档的方式，使 MongoDB 处理速度变得更快。BSON 还添加了一些标准 JSON 不支持的特性，包括为处理二进制数据添加类型的能力。在本章的 1.3.1 节“使用面向文档存储(BSON)”中将深入讲解 BSON。

RFC 4627 中描述了 JSON 的初始规范，它由 Douglas Crockford 制订。JSON 允许在简单的、可读文本格式中展现复杂的数据结构，使用它通常比阅读和理解 XML 要简单得多。如同 XML 一样，JSON 被设计为一种在 Web 客户端(例如浏览器)和 Web 应用之间交换数据的方式。由于它描述对象的丰富方式和简单性，大多数开发者都选择它作为交换数据的格式。

你可能会好奇复杂数据结构到底意味着什么。历史上，CSV(Comma-Separated Value, 逗号分隔值)曾用于交换数据(确实，这种方式在今天也仍然非常常见)。CSV 是一种简单的文本格式，使用换行符分隔不同行，使用逗号分隔不同字段。例如下面的 CSV 文件：