

Broadview[®]
www.broadview.com.cn

[PACKT] open source*
PUBLISHING community experience distilled

Mastering Redis

深入理解Redis

让你的Redis技能产生质的飞跃，让开发酷炫应用从此轻而易举

[美] Jeremy Nelson 著

汪佳南 译

 中国工信出版集团

 电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

Mastering Redis

深入理解Redis

[美] Jeremy Nelson 著
汪佳南 译

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书以由浅入深、由原理到应用场景的方式介绍了 Redis 这款 NoSQL 数据库产品。书中不仅细致地讲解了 Redis 中的数据结构及流行的使用模式，还针对 Redis 键的设计和管理，以及内存管理提出了建设性的方案。同时，作者深入 Redis 源码，将其内部构造通过源代码调试的方式进行呈现。

本书适合有一定 NoSQL 经验的开发者或者架构师阅读。读者可以从书中找到许多应用场景和解决方案，例如 Docker 部署、Redis 消息队列、基于 Redis 的 ETL 应用和基于 Redis 的机器学习等。

Copyright © 2016 Packt Publishing.

First published in the English language under the title ‘Mastering Redis’.

本书简体中文版专有出版权由 Packt Publishing 授予电子工业出版社。未经许可，不得以任何方式复制或抄袭本书的任何部分。专有出版权受法律保护。

版权贸易合同登记号 图字：01-2016-7238

图书在版编目（CIP）数据

深入理解 Redis / (美) 杰里米·尼尔森 (Jeremy Nelson) 著；汪佳南译. —北京：电子工业出版社，2017.4
书名原文：Mastering Redis

ISBN 978-7-121-31201-4

I. ①深… II. ①杰… ②汪… III. ①数据库—基本知识 IV. ①TP311.138

中国版本图书馆 CIP 数据核字(2017)第 065890 号

策划编辑：张春雨

责任编辑：郑柳洁

印 刷：北京中新伟业印刷有限公司

装 订：北京中新伟业印刷有限公司

出版发行：电子工业出版社

北京市海淀区万寿路 173 信箱 邮编：100036

开 本：787×980 1/16 印张：20 字数：416 千字

版 次：2017 年 4 月第 1 版

印 次：2017 年 4 月第 1 次印刷

定 价：89.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：(010) 88254888, 88258888。

质量投诉请发邮件至 zltz@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式：010-51260888-819, faq@phei.com.cn。

译者序

对不少互联网企业来说，使用 LNMP 架构可以快速搭建起一套系统，产品可以迅速迭代并尽早投放市场。但随着访问量的上升，那些使用传统关系型数据库（例如 MySQL）的网站开始显现出性能方面的问题。此外，越来越挑剔的用户也要求网站不能仅专注于功能特性，同时也要追求极致的产品体验（即高性能和高可用）。

一方面，架构师会在数据库层面做以下一系列优化。

1. 配置主从

例如，为 MySQL 服务器配置主从。一台宕机，另一台可以顶上继续服务，以满足高可用的要求。

2. 读写分离

配置一主多从。因为一般系统 80% 的请求都是读取操作，将这些操作分发到从服务器上可以有效地提升网站响应速度。

3. 分库分表

随着数据量的增长，单库单表已经难以满足要求。一些诸如 TDDL、Cobar、MyCAT 等的 MySQL 分布式数据库中间件应运而生。

另一方面，架构师也会通过系统化的分析，以安插缓存层的方式缓解数据库的压力。对应用系统来说，缓存都应在设计之初就纳入考虑的范畴，成为系统不可或缺的一部分。我曾供职的一家公司就采用 Redis 作为缓存的实现方案。应用程序运行在两个同等的 Tomcat 容器里。Tomcat 之前则配置了一台 Nginx 用于 Load Balance。由于用户 Session 存储在 Redis 中（通过 Spring 和 Shiro 的简单配置即可实现），应用的无状态性使得系统可以很方便地进行水平扩缩。另外，我们的业务要求用户每次浏览某个项目页面时进行访问计数（counting）。

为了避免频繁的数据库读写，我们将每个项目的浏览量的计算和读取交给了 Redis。同时，为了避免意外导致的数据丢失，我们开启了 Redis 的持久化功能，并启动定时任务，每隔一段时间就将项目浏览次数同步到数据库中。

至于为何选择 Redis 而没有选择 Memcached，当时考虑的主要原因是 Redis 支持更为丰富的数据模型。像上述页面访问量计数的需求，可以直接通过 Redis 的命令进行操作，而且利用 Redis 的单线程模型，应用程序无须编写同步代码。当然，缓存只是 Redis 的其中一个用途。一些典型的场景还包括排行榜、用户访问量统计、集合运算、消息队列等。国内外的一些大型互联网企业（例如京东、新浪、Pinterest 等）都对 Redis 有不同程度的部署和应用。希望读者能够在本书中找到自己想要的答案。

每次翻译都带给我不同的体验，并为我的工作和生活带来改变。在此，我要感谢电子工业出版社的编辑张春雨和负责审校的同事，是你们的细心指导保证了本书的翻译质量。同时，感谢我的父母和我的太太，在你们的陪伴和支持下，我得以专心工作。

由于时间仓促，文中难免有所疏漏，望不吝斧正。

汪佳南

关于作者

Jeremy Nelson 是科罗拉多斯普林市的一所四年制私立文理学院科罗拉多学院的一位元数据和系统图书管理员。除了每周 8 小时的图书馆研究技术支持工作，为大学生提供信息素养指导，并监督图书馆的系统 and 编目部门这三项工作之外，Nelson 正在积极研究和开发 Catalog Pull 平台中的各种组件和开源工具，供科罗拉多大学、科罗拉多州研究图书馆联盟和国会图书馆使用。他还是语义网络初创公司 KnowledgeLinks.io 的联合创始人和 CTO。

他之前在西部州科罗拉多大学和犹他大学有过图书馆工作经验。在成为图书管理员之前，他曾在各种软件公司和金融服务机构中担任程序员和项目经理。他的第一本书 *Becoming a Lean Library* 于 2015 年出版，将精益创业和精益制造理念应用于图书馆和图书馆的运营。Nelson 从诺克斯学院获得了本科学位，并从 University of Illinois Urbana-Champaign 获得了图书馆和信息科学的科学硕士学位。

关于审校者

Emilien Kenler 在从事了一些小型 Web 项目之后,在 2008 年高中时开始专注于游戏开发。直到 2011 年,他为不同的小组工作并专门从事系统管理。

2011 年,在研究计算机科学工程的同时,他创立了一家公司销售 Minecraft 服务器。他基于像 Node.js 和 RabbitMQ 这样的新技术,创建了一个轻量级 IaaS (<https://github.com/HostYourCreeper/>)。

此后,他在 TaDaweb 担任系统管理员,构建基础架构并创建管理部署和监控的工具。

2014 年,他在东京 Wizcorp 开启了新的历程。同年,他毕业于 University of Technology of Compiègne。

Emilien 为 Packt Publishing 编写了 *MariaDB Essentials*。他还负责了 *Learning Nagios 4*、*MariaDB High Performance*、*OpenVZ Essentials*、*Vagrant Virtual Development Environment Cookbook* 和 *Getting Started with MariaDB-Second Edition* 的审校。

Saurabh Minni 拥有计算机科学专业的工程学位。他有超过 10 年的工作经验,通晓各种编程语言,包括汇编语言、C、C++、Java、Delphi、JavaScript、Android、iOS、PHP、Python、ZMQ、Redis、Mongo、Kyoto Tycoon、Cocoa、Carbon、Apache Kafka、Apache Storm 和 ElasticSearch。总之,他是一位彻头彻尾的程序员,喜欢每天学习与技术相关的新事物。

目前,他在 Near 公司(这是一家神奇的初创公司,正在搭建位置智能平台)中担任技术架构师一职。除了处理几个项目之外,他还负责部署 Apache Kafka 集群。这有助于简化大数据处理系统中的数据消费。这些系统包括 Apache Storm、Hadoop,等等。

Saurabh 同时也是 *Apache Kafka Cookbook* 一书的作者。这是一本有关 Apache Kafka 的书,由 Packt Publishing 出版。他还负责 *Learning Apache Kafka* 一书的审校。该书由 Packt Publishing 出版。

你可以在 Twitter 上通过@the100rabh 联系他，也可以在 <https://github.com/the100rabh/> 上找到他。

正是因为我的父母 Suresh 和 Sarla，以及我太太 Puja 的不断支持，才成就了本书。感谢有你们时刻陪伴在我身边。

前言

本书旨在从两方面为读者构建 Redis 的基础知识。一方面，本书提供了 Redis 及其技术背后的深层含义及理论；另一方面，拓展了 Redis 日常实用技能。本书书名中的精通（*Mastering*）二字暗示了精通 Redis 是一个持续的过程，而非最终目的地。激动人心的是 Redis 持续开放地演进成为了时下强大的数据操作和存储技术。

Redis 背后的设计哲学

在整个项目的生命周期中，Salvatore Sanfilippo 对 Redis 的发展方向与功能发表了独到的观点和见解。在 2015 年 1 月的一篇有关 Redis 对比其他数据库的基准测试的博客中，Sanfilippo 声明“我不想说服开发者们采用 Redis。我们只是尽力提供一款合适的产品。如果人们能够使用这款产品完成工作，我们会感到非常开心。这就是我的营销理念。”Sanfilippo 和他的 Redis 核心开发团队遵循着成功的开源管理模型：“仁慈独裁者”（BDL）模型。该模型中只能有单独一个人作为最终独裁者，来裁决哪些能被提交到 Redis 代码库中。BDL 模型的成功已经被诸如 Linux 内核开发和 Python 编程语言等项目证明过了。作为主要开发者和维护者的 Sanfilippo 成功地将 BDL 模型复制到了 Redis 中。

如果独裁者抛弃项目，或者更糟的是因病或者死亡而导致无法工作时，BDL 模型的失效将是灾难性的。Redis 浮现出来的另一个重大问题是当潜在的贡献者提交 Pull Request 时，针对这些提交的行动会被延迟，或者更多时候是被忽略。说句公道话，那些必须经过检查、测试并合并到主代码库上的变更的数量非常巨大，需要激情和专门的看门人。作为 Linux 内核项目的初创者及当前的 BDL，Linus Torvalds 已经看到自己的角色发生了转变，更多是在合并那些由其他开发者贡献的代码，同时相比亲自编写代码，他做的更多的是为 Linux 提供愿景和领导力。Sanfilippo 在 Redis 主要的电子邮件通信上的一篇主题帖中确认了该问题，他给出了以下两大主要理由来继续 Redis 当前的 BDL 模型：

- 项目开发与未来方向的一致性视角
- 对任何新的或者出现的更改采取问责制

在 Sanfilippo 看来，Redis 作为键值数据存储，其易于部署、内存占用少（就 Redis 本身来说，而不是指它的数据集！）、安全可靠等特性一直是 Redis 在开发者和组织中人气持续上升的关键因素。他的观点确实造成了紧张的关系，特别是当提出 Redis 的新功能，例如使哈希中具体子值（sub-values）过期，或者为可选功能提供可加载模块时，这些特性都被拒之门外。Sanfilippo 对于保持 Redis 的小巧并专注于使其成为内存数据库的渴望，推动着他的决策和开发实践。

在 2011 年的博客中，他用有关 Redis 和 Redis 开发过程的七条宣言阐明了他的观点，简述如下。

1. **用于抽象数据类型的 DSL**。Redis 是一门领域特定语言（DSL），用于抽象数据结构的表达和使用。这些数据结构不仅包括了操作（Redis 命令），还包括了使用 Redis 相关命令存储和操作这些数据结构的内存效率和时间复杂度。

2. **内存存储是第一要务**。通过将所有数据存储存储在计算机内存中，跨系统的 Redis 的性能更为一致，用于实现这些数据结构的众多算法以更为可预测的方式运行，同时诸如有序集合这样的更为复杂的数据类型在内存数据库中更容易实现。

3. **基础数据结构对应基础 API**。Redis 为基础数据结构实现了一套基础 API。这套 API 由 Redis 命令和对应的数据结构组成。它尝试清晰地反映 API 从计算机内存中读取和写入的数据结构。Redis 遵循这一设计理念，通过更简单的数据结构操作来组合实现更为复杂的操作。

4. **代码如诗**。这是所有七条宣言之中最难以渗透的。Sanfilippo 将他的审美偏好融入到代码之中以契合 Redis 这一长篇巨著。他认为 Redis 的编码风格和方法能够帮助人们书写一段叙述。因此，是否包含第三方代码部分取决于它能否很好地契合 Redis 的叙述和 Redis 的源代码。

5. **我们反对复杂**。避免代码复杂性。如果要在使用大量代码来实现小型功能与舍弃该功能之间做出选择的话，那么 Redis 会选择后者，以舍弃额外的复杂性和向代码库中添加复杂性所带来的开销。

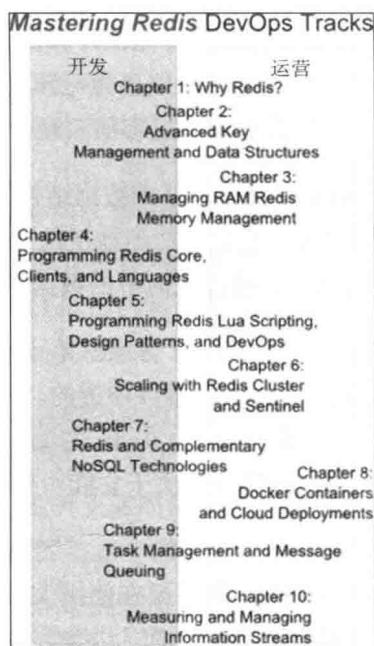
6. **两层 API**。一套是 API 的子集，以分布式的方式运行；另一套更大型、功能更为丰富，用来支持多键操作。这种分离设计支持像 Redis 主从和 Redis 集群操作模式这样重要的功能。

7. 我们以优化为乐。通常对于开发者和技术运营者来说，这是一种情感诉求，也是一段非常聪明的声明。调优技术来解决疑难杂症所带来的刺激能够激起快乐的感觉及对 Redis 未来无限可能的兴奋。

本书涵盖内容

阅读本书时你会发现其贯穿着两大主题，以平行的方式展示了流行时尚的运营和过程的开发/运营二元论，即众所周知的 DevOps。为了有助于读者更有针对性地学习章节中包含的内容，每个章节的主题会被归类为软件开发或者系统运营。由于两者之间的边界越来越模糊，对每个趋势中主题的深刻理解能够增强你和你团队的能力，以便快速高效地为项目开发 and 部署 Redis 解决方案，或者将 Redis 作为技术基础设施需求的一部分。

在下图中，我们可以看到每个章节的水平位置展示了其偏向于软件开发还是系统运营。



DevOps 轨迹

第 1 章介绍了由 Redis 之父和主要维护者 Salvatore Sanfilippo 所阐明的 Redis 开发哲学。

第 2 章通过详细阐述和解释 Redis 数据结构和键管理为读者构建起 Redis 的基础知识，其中包括了如何为应用程序构造既有意义又有表达力的键模式这一重要主题。

第 3 章讲解了 Redis 提供的用来优化应用程序内存使用的各种选项，其中包括了 Redis 支持的基于最近较少使用（LRU）的各种缓存和 Redis 中的键驱逐（evict）策略。

第 4 章是有关应用程序编程的高级主题。本章从概述 Redis 核心的 C 语言实现开始，通过对精心挑选的 C 代码片段进行深入地讲解来加深你对 Redis 的理解。然后讲到了如何使用三种不同的 Redis 客户端，并展示了分别使用 Python、Node.js 和 Haskell 的编程示例。¹



第 5 章是有关应用程序编程的高级主题。本章从概述 Redis 服务器端 Lua 脚本及如何在 Redis 中更高效地使用 Lua 开始讲起。之后，拓展讲解了一些流行的 Redis 编程设计模式，列举了几个具体实例，描述了不同的个体和公司在运营操作中是如何使用这些模式的。本章最后从软件开发者的角度描述了 Redis 是如何应用于典型的 DevOps 场景中。

第 6 章探索了两个最近添加到 Redis 中的 Redis 集群和 Redis Sentinel。Redis Sentinel 是一种特殊的高可用模式，用来监控主从服务器的健康状况，并能够在任意 Redis 主从实例故障时进行切换。之前提到的 Redis 集群现在可用于产品环境中了。对于那些太大而不能存入单台机器的大型数据来说，Redis 集群能够通过运行多个实例并将键进行分片的方式来完成存储。虽然这些主题更关注运营方面，但是在采用 Redis 的工程解决方案时至少应该知道 Redis 群集的好处和局限性。

第 7 章一开始就承认了一件事，那就是对绝大多数组织来说，它们的信息技术栈包含了不同类型的数据和处理解决方案的异构混合。Redis 是一种用来扩展其他 NoSQL 数据存

¹ 译者注：本章中没有 Haskell 示例。

储选项的理想方式。同时，在本章中，我们将看到 Redis 是如何与 MongoDB、ElasticSearch 和 Fedora Digital Repository 一起使用的。那些需要使用多种解决方案来开发并支持复杂业务需求的开发者和系统管理员会对本章的内容感兴趣。

第 8 章介绍了如何通过 Docker 容器和镜像中使用 Redis 的方式来简化管理并提升基于 Redis 的解决方案的安全性和可靠性。Docker 是一种开源容器技术，正快速地被许多企业采用。在那之后我们将研究在最为流行的计算云供应商上使用 Redis 所面临的具体挑战，这些供应商包括最大、最为成熟的 Amazon Web Services、Google 的 Compute Engine 和 Microsoft Azure，并特别提到了其他云服务供应商，例如 Rackspace 和 Digital Ocean。本章最后会研究专门用于 Redis 的云服务。这些云服务专注于托管和管理 Redis 实例。

第 9 章一开始深入探索了 Redis 的发布/订阅命令。首先通过不同的示例展示了不同进程间、不同程序间、不同的 Redis 客户端、不同的操作系统，以及远程计算机上的发布者和消费者是如何进行通信的。之后，将拓展 Redis 的发布/订阅模式，并简要介绍将 Redis 用作企业计算生态中不同层之间的消息通信队列。本章最后会通过使用 Redis 和 Celery 作为任务管理和支持发布/订阅模式的消息通信队列这一详细示例总结提到的所有概念。

第 10 章在前面章节的基础上展示了 Redis 作为实时数据聚合器是如何将组织中来自各种技术系统的不同数据流聚合起来的。之后，将研究 Redis 安全模型和最新 Redis 版本新加入的安全特性。还有一款基于 Web 的运营仪表盘将采用我们关于 Redis 客户端的知识，将输入到 Redis 中的数据进行可视化。接下来，将展示如何将机器学习算法（如朴素贝叶斯）应用到这些基于 Redis 的信息流，以提供更为丰富的概况，并加深你对组织或者部门内发生的操作的理解。

在附录的来源部分确认了章节中用到的摘录来源，并提供了链接以便读者进一步阅读。

获得精通 Redis 开放式徽章

Mozilla 基金会是赞助 Firefox 浏览器开发的开源组织，它启动了一项名为开放式徽章（Open Badges）的项目，允许组织创建并向个人颁发便携式和通用的徽章，以示其成就。



在本书网站上，你可以通过一系列在线测试获得精通 Redis 开放式徽章，并有机会向

你当前或者将来的雇主传达你对 Redis 日益增长的知识与技能。开放式徽章可以通过诸如 Facebook、Twitter 或者 LinkedIn 等流行的社交网站进行分享。

精通 Redis 开放式徽章对于购买了本书的读者来说是免费的。而对于那些没有购买的读者来说，仍然可以在本书网站上支付一些象征性的费用来获得精通 Redis 开放式徽章。你有机会与其他徽章获得者建立联系，从他们的经验中学习 Redis，同时可以分享自己的故事和知识，因此能够在阅读完本书之后的很长一段时间里鼓励你学习。我们希望本书能够马上帮助你理解 Redis，你能够通过获得开放式徽章记录这项专业成就。

需要为本书准备什么

Redis 设计用于在（诸如拥有现代 C++ 编译器）的 Linux 或者 Mac OS 等基于 POSIX 的环境下运行。当然，也有 Microsoft Windows 版本的 Redis，但是它并不受官方支持。更多信息请查看 <http://redis.io/download> 上有关 Windows 的这一节。本书中的示例使用了 Python 3.5，以及 Redis 的 Python 客户端（<https://github.com/andymccurdy/redis-py>）、Lua 和 Node.js，以及 Redis 的 Node.js 客户端（https://github.com/NodeRedis/node_redis）。

目标读者

如果你是一位 Web 开发者，对 MEAN 栈有基本的理解，有过 JavaScript 应用程序的开发经验，并且对 NoSQL 数据库也有基本的使用经验，那么本书就是为你而写的。

排版约定

你会发现本书采用了多种文本样式来区分不同种类的信息。下面是这些样式的一些示例和对应含义解释。文本中的代码、数据库表名称、文件夹名称、文件名、文件扩展名、路径名、虚拟 URL、用户输入和 Twitter handle 如下所示：“首先，在对键使用 EXPIRE 命令设置超时，只有当删除或者替换该键时超时才会被清除。”

代码块如下所示：


```
def create_tea(datastore, name, time, size):
    # Increment and save global counter
    tea_counter = datastore.incr("global/teas")
    tea_key = "tea/{}".format(tea_counter)
    datastore.hmset(tea_key,
```


```
    {"name": name,  
     "brew-time": time,  
     "box-size": size})  
    return tea_key
```

任何命令行的输入或者输出如下所示：

```
127.0.0.1:6379> LATENCY HISTORY command
```

```
1) 1) (integer) 1433877379  
    2) (integer) 1000  
2) 1) (integer) 1433877394  
    2) (integer) 250
```

[ 警告或者重要注意事项会出现在这个框中。]

[ 技巧和诀窍会出现在这里。]

读者服务

轻松注册成为博文视点社区用户（www.broadview.com.cn），您即可享受以下服务。

- **提交勘误：**您对书中内容的修改意见可在【提交勘误】处提交，若被采纳，将获赠博文视点社区积分（在您购买电子书时，积分可用来抵扣相应金额）。
- **与作者交流：**在页面下方【读者评论】处留下您的疑问或观点，与我们和其他读者一同学习交流。

页面入口：<http://www.broadview.com.cn/31201>

二维码：



目录

1 为何选择 Redis?	1
合适之选?	2
尝试使用 Redis	4
流行的使用模式	9
今非昔比	11
总结	13
2 高级键管理与数据结构	14
Redis 键	14
Redis 键模式	15
键分隔符和命名约定	17
手动创建 Redis 模式	19
解构 Redis 对象映射器	22
键过期	27
键的注意事项	27
大 O 符号	28
为自定义代码计算大 O 符号	30
回顾 Redis 数据结构的时间复杂度	32
字符串	32
哈希	33
列表	34
集合	35
有序集合	36
高级有序集合操作	39
位串和位操作	39

HyperLogLogs.....	41
总结	42
3 内存管理的建议与技巧	43
配置 Redis	43
主从复制	44
32 位 Redis	44
INFO memory 详解	46
键过期	48
LRU 键驱逐策略	53
创建内存高效的 Redis 数据结构	61
小巧的哈希、列表、集合和有序集合	61
把位、字节和 Redis 字符串用作随机访问数组	67
优化哈希，高效存储	68
硬件和网络延迟	71
操作系统建议	73
总结	74
4 Redis 编程第一部分：Redis 核心、客户端和编程语言	75
Redis 的内部结构	75
理解 redis.h 和 redis.c	82
Redis 序列化协议.....	92
Redis RDB 格式	95
使用 Redis 和 Python 创建协程	98
使用 Node.js 和 Redis 实现 Todo 列表应用.....	102
复制与公共访问.....	105
总结	105
5 Redis 编程第二部分：Lua 脚本、管理与 DevOps	106
在 Redis 中使用 Lua.....	106
使用 Redis 的 KEYS 和 ARGV	115