

多位业界专家联合推荐
来自一线开发者的实战经验总结



Redis

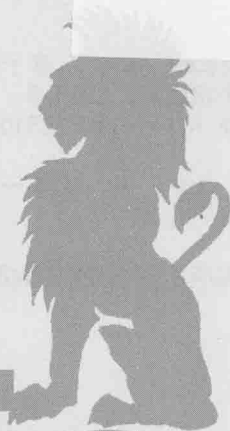
入门指南

(第2版)

李子骅 编著

真正零基础入门，深入浅出全面剖析 Redis
任务驱动式学习，轻松掌握 Redis 实战知识

 人民邮电出版社
POSTS & TELECOM PRESS



Redis

入门指南

(第2版)

李子骅 编著

人民邮电出版社

北京

图书在版编目 (C I P) 数据

Redis入门指南 / 李子骅编著. — 2版. — 北京 :
人民邮电出版社, 2015. 5
ISBN 978-7-115-38840-7

I. ①R… II. ①李… III. ①数据库—基本知识
IV. ①TP311.13

中国版本图书馆CIP数据核字(2015)第072122号

内 容 提 要

本书是一本 Redis 的入门指导书籍, 以通俗易懂的方式介绍了 Redis 基础与实践方面的知识, 包括历史与特性、在开发和生产环境中部署运行 Redis、数据类型与命令、使用 Redis 实现队列、事务、复制、管道、持久化、优化 Redis 存储空间等内容, 并采用任务驱动的方式介绍了 PHP、Ruby、Python 和 Node.js 这 4 种语言的 Redis 客户端库的使用方法。

本书的目标读者不仅包括 Redis 的新手, 还包括那些已经掌握 Redis 使用方法的人。对于新手而言, 本书的内容由浅入深且紧贴实践, 旨在让读者真正能够即学即用; 对于已经了解 Redis 的读者, 通过本书的大量实例以及细节介绍, 也能发现很多新的技巧。

-
- ◆ 编 著 李子骅
责任编辑 杨海玲
责任印制 张佳莹 焦志炜
 - ◆ 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>
三河市海波印务有限公司印刷
 - ◆ 开本: 800×1000 1/16
印张: 14
字数: 296 千字 2015 年 5 月第 2 版
印数: 6 501 - 9 500 册 2015 年 5 月河北第 1 次印刷
-

定价: 49.00 元

读者服务热线: (010)81055410 印装质量热线: (010)81055316
反盗版热线: (010)81055315

前言

Redis 如今已经成为 Web 开发社区中最火热的内存数据库之一，而它的诞生距现在不过才 4 年。随着 Web 2.0 的蓬勃发展，网站数据快速增长，对高性能读写的需求也越来越多，再加上半结构化的数据比重逐渐变大，人们对早已被铺天盖地地运用着的关系数据库能否适应现今的存储需求产生了疑问。而 Redis 的迅猛发展，为这个领域注入了全新的思维。

Redis 凭借其全面的功能得到越来越多的公司的青睐，从初创企业到新浪微博这样拥有几百台 Redis 服务器的大公司，都能看到 Redis 的身影。Redis 也是一个名副其实的多面手，无论是存储、队列还是缓存系统，都有它的用武之地。

本书将从 Redis 的历史讲起，结合基础与实践，带领读者一步步进入 Redis 的世界。

第 2 版说明

在本书第 1 版截稿的时候，加入了 Lua 脚本功能的 Redis 2.6 版刚刚发布，此时的 Redis 正在逐渐地被国内的开发者所熟知。如今整整两年过去了，Redis 也即将发布新的里程碑版本 3.0 版。在这两年中，Redis 增加了许多优秀的功能，同时也被越来越多的公司所采用与信赖。在写这段文字时，恰好 Redis 的作者 Salvatore Sanfilippo 转述了别人的一句话：“如果把 Redis 官网的‘谁在使用 Redis’页面改名为‘谁没在使用 Redis’，那么这个页面的内容一定会精简不少。”虽然是一句玩笑话，但是也从侧面体现出这两年里 Redis 的飞速发展。而继续编写《Redis 入门指南》第 2 版的最大动力也是希望将 Redis 发展的成果及时地与广大读者分享，同时也借此感谢大家对本书第 1 版的积极反馈。

目标读者

本书假定读者是 Redis 的新手，甚至可能连 Redis 是什么都没听说过。本书将会详细地介绍 Redis 是什么以及为什么要使用 Redis，旨在能让读者从零开始逐步晋升为一个优秀的 Redis 开发者。

本书还包含了很多 Redis 实践方面的知识，对于有经验的 Redis 开发者，大可以直接跳过已经掌握的内容，只阅读感兴趣的部分。每章的引言都简要介绍了本章要讲解的内容，供读者参考。

本书并不需要读者有任何 Redis 的背景知识，不过如果读者有 Web 后端开发经验或 Linux 系统使用经验，阅读本书将会更加得心应手。

组织结构

第 1 章介绍了 Redis 的历史与特性，主要回答两个初学者最关心的问题，即 Redis 是什么和为什么要使用 Redis。

第 2 章讲解了如何安装和运行 Redis。如果你身旁的计算机没有运行 Redis，那么一定不要错过这一章，因为本书后面的部分都需要读者最好能一边阅读一边实践，以提高学习效率。本章中还会介绍 Redis 命令行客户端的使用方法等基础知识，这些都是实践前需要掌握的知识。

第 3 章介绍了 Redis 的数据类型。本章讲解的不仅是每个数据类型的介绍和命令的格式，还会着重讲解每个数据类型分别在实践中如何使用。整个第 3 章会带领读者从零开始，一步步地使用 Redis 构建一个博客系统，旨在帮助读者在学习完本章的内容之后可以直接在自己的项目中上手实践 Redis。

第 4 章引入了一些 Redis 的进阶知识，比如事务和消息系统等。同样本章还会继续以博客系统为例子，以实践驱动学习。

第 5 章介绍了如何在各个编程语言中使用 Redis，这些语言包括 PHP、Ruby、Python 和 Node.js。其中讲解每种语言时最后都会以一个有趣的例子作为演示，即使你不了解某些语言，阅读这些例子也能让你收获颇丰。

第 6 章展示了 Redis 脚本的强大功能。本章会向读者讲解如何借助脚本来扩展 Redis，并且会对脚本一些需要注意的地方（如沙盒、随机结果等）进行着重介绍。

第 7 章会介绍 Redis 持久化的知识。Redis 持久化包含 RDB 和 AOF 两种方式，对持久化的支持是 Redis 之所以可以用作数据库的必要条件。

第 8 章详细说明了多个 Redis 实例的维护方法，包括使用复制实现读写分离、借助哨兵来自动完成故障恢复以及通过集群来实现数据分片。

第 9 章介绍了 Redis 安全和协议相关的内容，并向会推荐几个第三方的 Redis 管理工具。

附录 A 收录了 Redis 命令的不同属性，以及属性的特征。

附录 B 收录了 Redis 部分配置参数的章节索引。

附录 C 收录了 Redis 使用的 CRC16 实现代码。

排版约定

本书排版使用字体遵从以下约定。

- 等宽字：表示在命令行中输入的命令以及返回结果、程序代码、Redis 的命令（包括命令语句和命令定义）。
- 等宽斜体字（或夹在其中的中文楷体字）：表示命令或程序代码中由读者自行替换的参数或变量。

- 等宽粗体字：表示命令行中用户的输入内容、伪代码中的 Redis 命令。
- 命令行的输入和输出以如下格式显示：

```
$ redis-cli PING
PONG
```

- Redis 命令行客户端的输入和输出以如下格式显示：

```
redis> SET foo bar
OK
```

- 程序代码以如下格式显示：

```
var redis = require("redis");
var client = redis.createClient();

// 将两个对象 JSON 序列化后存入数据库中
client.mset(
  'user:1', JSON.stringify(bob),
  'user:2', JSON.stringify(jeff)
);
```

代码约定

本书的部分章节采用了伪代码来讲解，这种伪代码类似 Ruby 和 PHP，例如：

```
def hsetnx($key, $field, $value)
  $isExists = HEXISTS $key, $field
  if $isExists is 0
    HSET $key, $field, $value
    return 1
  else
    return 0
```

其中变量使用 \$ 符号标识，Redis 命令使用的粗体表示并省略了括号以便于阅读。在命令调用和 print 等语句中没有 \$ 符号的字符串会被当作字符串字面值。

附加文件

本书第 5 章中每一节都包含了一个完整的程序，通常来讲读者最好自己输入这些代码来加深理解，当然如果要先看到程序的运行结果再开始学习也不失为一个好办法。

这些程序代码都存放在 GitHub 上 (<https://github.com/luin/redis-book-assets>)，可以在 GitHub 上查看与下载。

致谢

在本书写作的过程中，得到了很多朋友的帮助。请允许我在这里占用少许篇幅，向他们致以诚挚的谢意。

感谢人民邮电出版社的杨海玲老师对本书的支持，没有她的悉心指导，本书就无法顺利完成。

感谢刘亚晨、李欣越、寇祖阳和余尧，他们承担了许多额外的工作，使得我可以全身心地投入到写作中。

感谢所有浏览本书初稿并提出意见和建议的人们：张沈鹏、陈硕实、刘其帅、扈焯、李其超、朱冲宇、王诗吟、黄山月、刘昕、韩重远、李申申、杨海朝、田琪等。感谢你们的支持。

另外还要感谢“宋老师”，是的，就是书中的主人公之一。几年前，我刚创业时，办公场所是和某个教育机构合租的，宋老师是该机构的一名英语老师，同时他也是国内一个知名的嘻哈乐团成员之一。他平日风趣的谈吐带给了我们很多欢乐，伴随我们走过了艰苦的创业初期，而我接触 Redis，也正是从这段时间开始的。

最后，感谢我的父母和女朋友马梦妍，你们是我生命中最重要的人，感谢你们的理解和支持。

目 录

第 1 章 简介	1	3.2.4 命令拾遗	27
1.1 历史与发展	1	3.3 散列类型	32
1.2 特性	2	3.3.1 介绍	33
1.2.1 存储结构	2	3.3.2 命令	34
1.2.2 内存存储与持久化	3	3.3.3 实践	37
1.2.3 功能丰富	3	3.3.4 命令拾遗	39
1.2.4 简单稳定	4	3.4 列表类型	40
第 2 章 准备	7	3.4.1 介绍	41
2.1 安装 Redis	7	3.4.2 命令	41
2.1.1 在 POSIX 系统中安装	7	3.4.3 实践	44
2.1.2 在 OS X 系统中安装	8	3.4.4 命令拾遗	46
2.1.3 在 Windows 中安装	9	3.5 集合类型	48
2.2 启动和停止 Redis	11	3.5.1 介绍	48
2.2.1 启动 Redis	12	3.5.2 命令	49
2.2.2 停止 Redis	14	3.5.3 实践	52
2.3 Redis 命令行客户端	14	3.5.4 命令拾遗	54
2.3.1 发送命令	14	3.6 有序集合类型	57
2.3.2 命令返回值	15	3.6.1 介绍	57
2.4 配置	17	3.6.2 命令	58
2.5 多数据库	17	3.6.3 实践	62
第 3 章 入门	19	3.6.4 命令拾遗	63
3.1 热身	19	第 4 章 进阶	67
3.2 字符串类型	21	4.1 事务	67
3.2.1 介绍	22	4.1.1 概述	68
3.2.2 命令	22	4.1.2 错误处理	69
3.2.3 实践	26	4.1.3 WATCH 命令介绍	70
		4.2 过期时间	72
		4.2.1 命令介绍	73

4.2.2	实现访问频率限制之一	75	5.3.3	简使用法	117
4.2.3	实现访问频率限制之二	76	5.3.4	实践: 在线的好友	117
4.2.4	实现缓存	77	5.4	Node.js 与 Redis	123
4.3	排序	78	5.4.1	安装	123
4.3.1	有序集合的集合操作	78	5.4.2	使用方法	123
4.3.2	 SORT 命令	79	5.4.3	简使用法	126
4.3.3	 BY 参数	81	5.4.4	实践: IP 地址查询	127
4.3.4	 GET 参数	83	第 6 章	脚本	131
4.3.5	 STORE 参数	84	6.1	概览	131
4.3.6	性能优化	85	6.1.1	脚本介绍	132
4.4	消息通知	85	6.1.2	实例: 访问频率限制	132
4.4.1	任务队列	86	6.2	Lua 语言	133
4.4.2	使用 Redis 实现任务队列	87	6.2.1	Lua 语法	134
4.4.3	优先级队列	88	6.2.2	标准库	143
4.4.4	“发布/订阅”模式	90	6.2.3	其他库	147
4.4.5	按照规则订阅	91	6.3	Redis 与 Lua	147
4.5	管道	92	6.3.1	在脚本中调用 Redis 命令	148
4.6	节省空间	93	6.3.2	从脚本中返回值	148
4.6.1	精简键名和键值	94	6.3.3	脚本相关命令	149
4.6.2	内部编码优化	94	6.3.4	应用实例	150
第 5 章	实践	103	6.4	深入脚本	153
5.1	PHP 与 Redis	103	6.4.1	 KEYS 与 ARGV 	153
5.1.1	安装	104	6.4.2	沙盒与随机数	154
5.1.2	使用方法	104	6.4.3	其他脚本相关命令	154
5.1.3	简使用法	105	6.4.4	原子性和执行时间	155
5.1.4	实践: 用户注册登录功能	107	第 7 章	持久化	157
5.2	Ruby 与 Redis	111	7.1	RDB 方式	157
5.2.1	安装	111	7.1.1	根据配置规则进行自动 快照	158
5.2.2	使用方法	111	7.1.2	用户执行 SAVE 或 BGSAVE 命令	158
5.2.3	简使用法	112	7.1.3	执行 FLUSHALL 命令	159
5.2.4	实践: 自动完成	112	7.1.4	执行复制时	159
5.3	Python 与 Redis	116			
5.3.1	安装	116			
5.3.2	使用方法	116			

7.1.5 快照原理	159	第9章 管理	193
7.2 AOF 方式	160	9.1 安全	193
7.2.1 开启 AOF	160	9.1.1 可信的环境	193
7.2.2 AOF 的实现	161	9.1.2 数据库密码	194
7.2.3 同步硬盘数据	162	9.1.3 命名命令	194
第8章 集群	165	9.2 通信协议	195
8.1 复制	165	9.2.1 简单协议	195
8.1.1 配置	166	9.2.2 统一请求协议	197
8.1.2 原理	168	9.3 管理工具	197
8.1.3 图结构	170	9.3.1 redis-cli	198
8.1.4 读写分离与一致性	171	9.3.2 phpRedisAdmin	199
8.1.5 从数据库持久化	171	9.3.3 Rdbtools	201
8.1.6 无硬盘复制	171	附录 A Redis 命令属性	203
8.1.7 增量复制	172	A.1 REDIS_CMD_WRITE	203
8.2 哨兵	173	A.2 REDIS_CMD_DENYOOM	205
8.2.1 什么是哨兵	173	A.3 REDIS_CMD_NOSCRIPT	206
8.2.2 马上上手	174	A.4 REDIS_CMD_RANDOM	207
8.2.3 实现原理	177	A.5 REDIS_CMD_SORT_FOR_	
8.2.4 哨兵的部署	180	SCRIPT	207
8.3 集群	181	A.6 REDIS_CMD_LOADING	207
8.3.1 配置集群	181	附录 B 配置参数索引	209
8.3.2 节点的增加	185	附录 C CRC16 实现参考	211
8.3.3 插槽的分配	185		
8.3.4 获取与插槽对应的节点	190		
8.3.5 故障恢复	191		

第 1 章

简介

Redis 是一个开源的、高性能的、基于键值对的缓存与存储系统，通过提供多种键值数据类型来适应不同场景下的缓存与存储需求。同时 Redis 的诸多高层级功能使其可以胜任消息队列、任务队列等不同的角色。

本章将分别介绍 Redis 的历史和特性，以使读者能够快速地对 Redis 有一个全面的了解。

1.1 历史与发展

2008 年，意大利的一家创业公司 Merzia^①推出了一款基于 MySQL 的网站实时统计系统 LLOOGG^②，然而没过多久该公司的创始人 Salvatore Sanfilippo 便开始对 MySQL 的性能感到失望，于是他决定亲自为 LLOOGG 量身定做做一个数据库，并于 2009 年开发完成，这个数据库就是 Redis。不过 Salvatore Sanfilippo 并不满足只将 Redis 用于 LLOOGG 这一款产品，而是希望让更多的人使用它，于是在同一年 Salvatore Sanfilippo 将 Redis 开源发布，并开始和 Redis 的另一名主要的代码贡献者 Pieter Noordhuis 一起继续着 Redis 的开发，直到今天。

Salvatore Sanfilippo 自己也没有想到，短短的几年时间，Redis 就拥有了庞大的用户群体。Hacker News 在 2012 年发布了一份数据库的使用情况调查^③，结果显示有近 12% 的公司在使用 Redis。国内如新浪微博、街旁和知乎，国外如 GitHub、Stack Overflow、Flickr、暴雪和 Instagram，都是 Redis 的用户。

VMware 公司从 2010 年开始赞助 Redis 的开发，Salvatore Sanfilippo 和 Pieter Noordhuis

^① <http://merzia.com>

^② <http://lloogg.com>

^③ <http://news.ycombinator.com/item?id=4833188>

也分别于同年的3月和5月加入VMware，全职开发Redis。

Redis的代码托管在GitHub上，开发十分活跃^①。2015年4月2日，Redis发布了3.0.0的正式版本。

1.2 特性

作为一款个人开发的系统，Redis究竟有什么魅力吸引了如此多的用户呢？

1.2.1 存储结构

有过脚本语言编程经验的读者对字典（或称映射、关联数组）数据结构一定很熟悉，如代码 `dict["key"] = "value"` 中 `dict` 是一个字典结构变量，字符串 `"key"` 是键名，而 `"value"` 是键值，在字典中我们可以获取或设置键名对应的键值，也可以删除一个键。

Redis 是 REremote DIctionary Server（远程字典服务器）的缩写，它以字典结构存储数据，并允许其他应用通过 TCP 协议读写字典中的内容。同大多数脚本语言中的字典一样，Redis 字典中的键值除了可以是字符串，还可以是其他数据类型。到目前为止 Redis 支持的键值数据类型如下：

- 字符串类型
- 散列类型
- 列表类型
- 集合类型
- 有序集合类型

这种字典形式的存储结构与常见的 MySQL 等关系数据库的二维表形式的存储结构有很大的差异。举个例子，如下所示，我们在程序中使用 `post` 变量存储了一篇文章的数据（包括标题、正文、阅读量和标签）：

```
post["title"] = "Hello World!"
post["content"] = "Blablabla..."
post["views"] = 0
post["tags"] = ["PHP", "Ruby", "Node.js"]
```

现在我们将这篇文章的数据存储在数据库中，并且要求可以通过标签检索出文章。如果使用关系数据库存储，一般会将其中的标题、正文和阅读量存储在一个表中，而将标签存储在另一个表中，然后使用第三个表连接文章和标签表^②。需要查询时还得将3个表进

^① <https://github.com/antirez/redis>

^② 这是一种符合第三范式的设计。事实上还可以使用其他方式来实现标签系统，参阅 (<http://tagging.pui.ch/post/37027745720/tags-database-schemas>) 以了解更多相关资料。

行连接，不是很直观。而 Redis 字典结构的存储方式和对多种键值数据类型的支持使得开发者可以将程序中的数据直接映射到 Redis 中，数据在 Redis 中的存储形式和其在程序中的存储方式非常相近。使用 Redis 的另一个优势是其对不同的数据类型提供了非常方便的操作方式，如使用集合类型存储文章标签，Redis 可以对标签进行如交集、并集这样的集合运算操作。3.5 节会专门介绍如何借助集合运算轻易地实现“找出所有同时属于 A 标签和 B 标签且不属于 C 标签”这样关系数据库实现起来性能不高且较为繁琐的操作。

1.2.2 内存存储与持久化

Redis 数据库中的所有数据都存储在内存中。由于内存的读写速度远快于硬盘，因此 Redis 在性能上对比其他基于硬盘存储的数据库有非常明显的优势，在一台普通的笔记本电脑上，Redis 可以在一秒内读写超过 10 万个键值。

将数据存储在内存中也有问题，比如程序退出后内存中的数据会丢失。不过 Redis 提供了对持久化的支持，即可以将内存中的数据异步写入到硬盘中，同时不影响继续提供服务。

1.2.3 功能丰富

Redis 虽然是作为数据库开发的，但由于其提供了丰富的功能，越来越多的人将其用作缓存、队列系统等。Redis 可谓是名副其实的多面手。

Redis 可以为每个键设置生存时间（Time To Live, TTL），生存时间到期后键会自动被删除。这一功能配合出色的性能让 Redis 可以作为缓存系统来使用，而且由于 Redis 支持持久化和丰富的数据类型，使其成为了另一个非常流行的缓存系统 Memcached 的有力竞争者。

讨论 关于 Redis 和 Memcached 优劣的讨论一直是一个热门的话题。在性能上 Redis 是单线程模型，而 Memcached 支持多线程，所以在多核服务器上后者的性能理论上相对更高一些。然而，前面已经介绍过，Redis 的性能已经足够优异，在绝大部分场合下其性能都不会成为瓶颈，所以在使用时更应该关心的是二者在功能上的区别。随着 Redis 3.0 的推出，标志着 Memcached 几乎所有功能都成为了 Redis 的子集。同时，Redis 对集群的支持使得 Memcached 原有的第三方集群工具不再成为优势。因此，在新项目中使用 Redis 代替 Memcached 将会是非常好的选择。

作为缓存系统，Redis 还可以限定数据占用的最大内存空间，在数据达到空间限制后可以按照一定的规则自动淘汰不需要的键。

除此之外，Redis 的列表类型键可以用来实现队列，并且支持阻塞式读取，可以很容易地实现一个高性能的优先级队列。同时在更高层面上，Redis 还支持“发布/订阅”的消息模式，可以基于此构建聊天室^①等系统。

1.2.4 简单稳定

即使功能再丰富，如果使用起来太复杂也很难吸引人。Redis 直观的存储结构使得通过程序与 Redis 交互十分简单。在 Redis 中使用命令来读写数据，命令语句之于 Redis 就相当于 SQL 语言之于关系数据库。例如在关系数据库中要获取 posts 表内 id 为 1 的记录的 title 字段的值可以使用如下 SQL 语句实现：

```
SELECT title FROM posts WHERE id = 1 LIMIT 1
```

相对应的，在 Redis 中要读取键名为 post:1 的散列类型键的 title 字段的值，可以使用如下命令语句实现：

```
HGET post:1 title
```

其中 HGET 就是一个命令。Redis 提供了 100 多个命令（如图 1-1 所示），听起来很多，但是常用的却只有十几个，并且每个命令都很容易记忆。读完第 3 章你就会发现 Redis 的命令比 SQL 语言要简单很多。

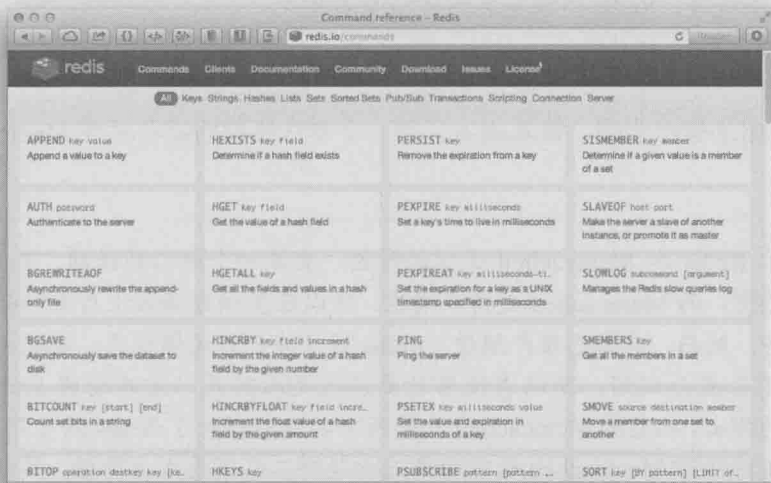


图 1-1 Redis 官网提供了详细的命令文档

^① Redis 的贡献者之一 Pieter Noordhuis 提供了一个使用该模式开发的聊天室的例子，见 <https://gist.github.com/348262>。

Redis 提供了几十种不同编程语言的客户端库，这些库都很好地封装了 Redis 的命令，使得在程序中与 Redis 进行交互变得更容易。有些库还提供了可以将编程语言中的数据类型直接以相应的形式存储到 Redis 中（如将数组直接以列表类型存入 Redis）的简单方法，使用起来非常方便。

Redis 使用 C 语言开发，代码量只有 3 万多行。这降低了用户通过修改 Redis 源代码来使之更适合自己的项目需要的门槛。对于希望“榨干”数据库性能的开发者的而言，这无疑是一个很大的吸引力。

Redis 是开源的，所以事实上 Redis 的开发者并不止 Salvatore Sanfilippo 和 Pieter Noordhuis。截至目前，有将近 100 名开发者为 Redis 贡献了代码。良好的开发氛围和严谨的版本发布机制使得 Redis 的稳定版本非常可靠，如此多的公司在项目中使用了 Redis 也可以印证这一点。

第 2 章

准备

“纸上得来终觉浅，绝知此事要躬行。”

——陆游《冬夜读书示子聿》

学习 Redis 最好的办法就是动手尝试它。在介绍 Redis 最核心的内容之前，本章先来介绍一下如何安装和运行 Redis，以及 Redis 的基础知识，使读者可以在之后的章节中一边学习一边实践。

2.1 安装 Redis

安装 Redis 是开始 Redis 学习之旅的第一步。在安装 Redis 前需要了解 Redis 的版本规则以选择最适合自己的版本，Redis 约定次版本号（即第一个小数点后的数字）为偶数的版本是稳定版（如 2.8 版、3.0 版），奇数版本是非稳定版（如 2.7 版、2.9 版），生产环境下一般需要使用稳定版本。本书的内容以 3.0 版为目标编写，同时绝大部分内容也适用于 2.6 版和 2.8 版。对于只在最新版才有的特性（如 Cluster 集群），本书会做特别说明。

2.1.1 在 POSIX 系统中安装

Redis 兼容大部分 POSIX 系统，包括 Linux、OS X 和 BSD 等，在这些系统中推荐直接下载 Redis 源代码编译安装以获得最新的稳定版本。Redis 最新稳定版本的源代码可以从地址 <http://download.redis.io/redis-stable.tar.gz> 下载。

下载安装包后解压即可使用 `make` 命令完成编译，完整的命令如下：