

# Redis 设计与实现

黄健宏 著

---

The Design and Implementation of Redis

---

- 系统而全面地描述了 Redis 内部运行机制。
- 图示丰富，描述清晰，并给出大量参考信息，是NoSQL数据库开发人员案头必备。
- 包括大部分Redis单机特征，以及所有多机特性。



数据库技术丛书

## Redis设计与实现

黄健宏 著

ISBN: 978-7-111-46474-7

本书纸版由机械工业出版社于2014年出版，电子版由华章分社（北京华章图文信息有限公司）全球范围内制作与发行。

版权所有，侵权必究

客服热线：+ 86-10-68995265

客服信箱：service@bbbvip.com

官方网址：www.hzmedia.com.cn

新浪微博 @研发书局

腾讯微博 @yanfabook

# 目录

[前言](#)

[致谢](#)

[第1章 引言](#)

[1.1 Redis版本说明](#)

[1.2 章节编排](#)

[1.3 推荐的阅读方法](#)

[1.4 行文规则](#)

[1.5 配套网站](#)

[第一部分 数据结构与对象](#)

[第2章 简单动态字符串](#)

[2.1 SDS的定义](#)

[2.2 SDS与C字符串的区别](#)

[2.3 SDS API](#)

[2.4 重点回顾](#)

[2.5 参考资料](#)

[第3章 链表](#)

[3.1 链表和链表节点的实现](#)

[3.2 链表和链表节点的API](#)

[3.3 重点回顾](#)

[第4章 字典](#)

[4.1 字典的实现](#)

[4.2 哈希算法](#)

[4.3 解决键冲突](#)

[4.4 rehash](#)

[4.5 渐进式rehash](#)

[4.6 字典API](#)

[4.7 重点回顾](#)

[第5章 跳跃表](#)

[5.1 跳跃表的实现](#)

[5.2 跳跃表API](#)

[5.3 重点回顾](#)

[第6章 整数集合](#)

[6.1 整数集合的实现](#)

[6.2 升级](#)

[6.3 升级的好处](#)

## [6.4 降级](#)

## [6.5 整数集合API](#)

## [6.6 重点回顾](#)

## [第7章 压缩列表](#)

### [7.1 压缩列表的构成](#)

### [7.2 压缩列表节点的构成](#)

### [7.3 连锁更新](#)

### [7.4 压缩列表API](#)

### [7.5 重点回顾](#)

## [第8章 对象](#)

### [8.1 对象的类型与编码](#)

### [8.2 字符串对象](#)

### [8.3 列表对象](#)

### [8.4 哈希对象](#)

### [8.5 集合对象](#)

### [8.6 有序集合对象](#)

### [8.7 类型检查与命令多态](#)

### [8.8 内存回收](#)

### [8.9 对象共享](#)

### [8.10 对象的空转时长](#)

### [8.11 重点回顾](#)

## [第二部分 单机数据库的实现](#)

## [第9章 数据库](#)

### [9.1 服务器中的数据库](#)

### [9.2 切换数据库](#)

### [9.3 数据库键空间](#)

### [9.4 设置键的生存时间或过期时间](#)

### [9.5 过期键删除策略](#)

### [9.6 Redis的过期键删除策略](#)

### [9.7 AOF、RDB和复制功能对过期键的处理](#)

### [9.8 数据库通知](#)

### [9.9 重点回顾](#)

## [第10章 RDB持久化](#)

### [10.1 RDB文件的创建与载入](#)

### [10.2 自动间隔性保存](#)

### [10.3 RDB文件结构](#)

### [10.4 分析RDB文件](#)

[10.5 重点回顾](#)

[10.6 参考资料](#)

[第11章 AOF持久化](#)

[11.1 AOF持久化的实现](#)

[11.2 AOF文件的载入与数据还原](#)

[11.3 AOF重写](#)

[11.4 重点回顾](#)

[第12章 事件](#)

[12.1 文件事件](#)

[12.2 时间事件](#)

[12.3 事件的调度与执行](#)

[12.4 重点回顾](#)

[12.5 参考资料](#)

[第13章 客户端](#)

[13.1 客户端属性](#)

[13.2 客户端的创建与关闭](#)

[13.3 重点回顾](#)

[第14章 服务器](#)

[14.1 命令请求的执行过程](#)

[14.2 serverCron函数](#)

[14.3 初始化服务器](#)

[14.4 重点回顾](#)

[第三部分 多机数据库的实现](#)

[第15章 复制](#)

[15.1 旧版复制功能的实现](#)

[15.2 旧版复制功能的缺陷](#)

[15.3 新版复制功能的实现](#)

[15.4 部分重同步的实现](#)

[15.5 PSYNC命令的实现](#)

[15.6 复制的实现](#)

[15.7 心跳检测](#)

[15.8 重点回顾](#)

[第16章 Sentinel](#)

[16.1 启动并初始化Sentinel](#)

[16.2 获取主服务器信息](#)

[16.3 获取从服务器信息](#)

[16.4 向主服务器和从服务器发送信息](#)

[16.5 接收来自主服务器和从服务器的频道信息](#)

[16.6 检测主观下线状态](#)

[16.7 检查客观下线状态](#)

[16.8 选举领头Sentinel](#)

[16.9 故障转移](#)

[16.10 重点回顾](#)

[16.11 参考资料](#)

[第17章 集群](#)

[17.1 节点](#)

[17.2 槽指派](#)

[17.3 在集群中执行命令](#)

[17.4 重新分片](#)

[17.5 ASK错误](#)

[17.6 复制与故障转移](#)

[17.7 消息](#)

[17.8 重点回顾](#)

[第四部分 独立功能的实现](#)

[第18章 发布与订阅](#)

[18.1 频道的订阅与退订](#)

[18.2 模式的订阅与退订](#)

[18.3 发送消息](#)

[18.4 查看订阅信息](#)

[18.5 重点回顾](#)

[18.6 参考资料](#)

[第19章 事务](#)

[19.1 事务的实现](#)

[19.2 WATCH命令的实现](#)

[19.3 事务的ACID性质](#)

[19.4 重点回顾](#)

[19.5 参考资料](#)

[第20章 Lua脚本](#)

[20.1 创建并修改Lua环境](#)

[20.2 Lua环境协作组件](#)

[20.3 EVAL命令的实现](#)

[20.4 EVALSHA命令的实现](#)

[20.5 脚本管理命令的实现](#)

[20.6 脚本复制](#)

[20.7 重点回顾](#)

[20.8 参考资料](#)

[第21章 排序](#)

[21.1 SORT命令的实现](#)

[21.2 ALPHA选项的实现](#)

[21.3 ASC选项和DESC选项的实现](#)

[21.4 BY选项的实现](#)

[21.5 带有ALPHA选项的BY选项的实现](#)

[21.6 LIMIT选项的实现](#)

[21.7 GET选项的实现](#)

[21.8 STORE选项的实现](#)

[21.9 多个选项的执行顺序](#)

[21.10 重点回顾](#)

[第22章 二进制位数组](#)

[22.1 位数组的表示](#)

[22.2 GETBIT命令的实现](#)

[22.3 SETBIT命令的实现](#)

[22.4 BITCOUNT命令的实现](#)

[22.5 BITOP命令的实现](#)

[22.6 重点回顾](#)

[22.7 参考资料](#)

[第23章 慢查询日志](#)

[23.1 慢查询记录的保存](#)

[23.2 慢查询日志的阅览和删除](#)

[23.3 添加新日志](#)

[23.4 重点回顾](#)

[第24章 监视器](#)

[24.1 成为监视器](#)

[24.2 向监视器发送命令信息](#)

[24.3 重点回顾](#)

# 前言

时间回到2011年4月，当时我正在编写一个用户关系模块，这个模块需要实现一个“共同关注”功能，用于计算出两个用户关注了哪些相同的用户。

举个例子，假设huangz关注了peter、tom、jack三个用户，而john关注了peter、tom、bob、david四个用户，那么当huangz访问john的页面时，共同关注功能就会计算并打印出类似“你跟john都关注了peter和tom”这样的信息。

从集合计算的角度来看，共同关注功能本质上就是计算两个用户关注集合的交集，因为交集这个概念是如此的常见，所以我很自然地认为共同关注这个功能可以很容易地实现，但现实却给了我当头一棒：我所使用的关系数据库并不直接支持交集计算操作，要计算两个集合的交集，除了需要对两个数据表执行合并（`join`）操作之外，还需要对合并的结果执行去重复（`distinct`）操作，最终导致交集操作的实现变得异常复杂。

是否存在直接支持集合操作的数据库呢？带着这个疑问，我在搜索引擎上面进行查找，并最终发现了Redis。在我看来，Redis正是我想要找的那种数据库——它内置了集合数据类型，并支持对集合执行交集、并集、差集等集合计算操作，其中的交集计算操作可以直接用于实现我想要的共同关注功能。

得益于Redis本身的简单性，以及Redis手册的详尽和完善，我很快学会了怎样使用Redis的集合数据类型，并用它重新实现了整个用户关系模块：重写之后的关系模块不仅代码量更少，速度更快，更重要的是，之前需要使用一段甚至一大段SQL查询才能实现的功能，现在只需要调用一两个Redis命令就能够实现了，整个模块的可读性得到了极大的提高。

自此之后，我开始在越来越多的项目里面使用Redis，与此同时，我对Redis的内部实现也越来越感兴趣，一些问题开始频繁地出现在我的脑海中，比如：



·Redis的五种数据类型分别是由什么数据结构实现的？

·Redis的字符串数据类型既可以存储字符串（比如“hello world”），又可以存储整数和浮点数（比如10086和3.14），甚至是二进制位（使用SETBIT等命令），Redis在内部是怎样存储这些值的？

·Redis的一部分命令只能对特定数据类型执行（比如APPEND只能对字符串执行，HSET只能对哈希表执行），而另一部分命令却可以对所有数据类型执行（比如DEL、TYPE和EXPIRE），不同的命令在执行时是如何进行类型检查的？Redis在内部是否实现了一个类型系统？

·Redis的数据库是怎样存储各种不同数据类型的键值对的？数据库里面的过期键又是怎样实现自动删除的？

·除了数据库之外，Redis还拥有发布与订阅、脚本、事务等特性，这些特性又是如何实现的？

·Redis使用什么模型或者模式来处理客户端的命令请求？一条命令请求从发送到返回需要经过什么步骤？

为了找到这些问题的答案，我再次在搜索引擎上面进行查找，可惜的是这次搜索并没有多少收获：Redis还是一个非常年轻的软件，对它的最好介绍就是官方网站上面的文档，但是这些文档主要关注的是怎样使用Redis，而不是介绍Redis的内部实现。另外，网上虽然有一些博客文章对Redis的内部实现进行了介绍，但这些文章要么不齐全（只介绍了Redis中的少数几个特性），要么就写得过于简单（只是一些概述性的文章），要么关注的就是旧版本（比如2.0、2.2或者2.4，而当时的最新版已经是2.6了）。

综合来看，详细而且完整地介绍Redis内部实现的资料，无论是外文还是中文都不存在。意识到这一点之后，我决定自己动手注释Redis的源代码，从中寻找问题的答案，并通过写博客的方式与其他Redis用户分享我的发现。在积累了七八篇Redis源代码注释文章之后，我想如果能将这些博文汇集成书的话，那一定会非常有趣，并且我自己也会从中学到很多知识。于是我在2012年年末开始创作《Redis设计与实现》，并最终于2013年3月8日在互联网发布了本书的第一版。

尽管《Redis设计与实现》第一版顺利发布了，但在我的心目中，

这个第一版还是有很大不完美的地方：

·比如说，因为第一版是我边注释Redis源代码边写的，如果有足够时间让我先完整地注释一遍Redis的源代码，然后再进行写作的话，那么书本在内容方面应该会更为全面。

·又比如说，第一版只介绍了Redis的内部机制和单机特性，但并没有介绍Redis多机特性，而我认为只有将关于多机特性的介绍也包含进来，这本《Redis设计与实现》才算是真正的完成了。

就在我考虑应该何时编写新版来修复这些缺陷的时候，机械工业出版社的吴怡编辑来信询问我是否有兴趣正式地出版《Redis设计与实现》，能够正式地出版自己写的书一直是我梦寐以求的事情，我找不到任何拒绝这一邀请的理由，就这样，在《Redis设计与实现》第一版发布几天之后，新版《Redis设计与实现》的写作也马不停蹄地开始了。

从2013年3月到2014年1月这11个月间，我重新注释了Redis在unstable分支的源代码（也即是现在的Redis 3.0源代码），重写了《Redis设计与实现》第一版已有的所有章节，并向书中添加了关于二进制位操作（bitop）、排序、复制、Sentinel和集群等主题的新章节，最终完成了这本新版的《Redis设计与实现》。本书不仅介绍了Redis的内部机制（比如数据库实现、类型系统、事件模型），而且还介绍了大部分Redis单机特性（比如事务、持久化、Lua脚本、排序、二进制位操作），以及所有Redis多机特性（如复制、Sentinel和集群）。

虽然作者创作本书的初衷只是为了满足自己的好奇心，但了解Redis内部实现的好处并不仅仅在于满足好奇心：通过了解Redis的内部实现，理解每一个特性和命令背后的运作机制，可以帮助我们更高效地使用Redis，避开那些可能会引起性能问题的陷阱。我衷心希望这本新版《Redis设计与实现》能够帮助读者更好地了解Redis，并成为更优秀的Redis使用者。

本书的第一版获得了很多热心读者的反馈，这本新版的很多改进也来源于读者们的意见和建议，因此我将继续在[www.RedisBook.com](http://www.RedisBook.com)设置disqus论坛（可以不注册直接发帖），欢迎读者随时就这本新版《Redis设计与实现》发表提问、意见、建议、批评、勘误，等等，我会努力地采纳大家的意见，争取在将来写出更好的《Redis设计与实现》，以此来回报大家对本书的支持。

欢迎访问：电子书学习和下载网站 (<https://www.shgis.com>)

文档名称：《Redis设计与实现（数据库技术丛书）》黄健宏 著.pdf

请登录 <https://shgis.com/post/4091.html> 下载完整文档。

手机端请扫码查看：

