



## 开篇：授人以鱼不若授人以渔 —— Redis 可以用来做什么？

Redis 是互联网技术领域使用最为广泛的存储中间件，它是「**Remote Dictionary Service**」的首字母缩写，也就是「远程字典服务」。Redis 以其超高的性能、完美的文档、简洁易懂的源码和丰富的客户端库支持在开源中间件领域广受好评。国内外很多大型互联网公司都在使用 Redis，比如 Twitter、YouPorn、暴雪娱乐、Github、StackOverflow、腾讯、阿里、京东、华为、新浪微博等等，很多中小型公司也都有应用。也可以说，对 Redis 的了解和应用实践已成为当下中高级后端开发者绕不开的必备技能。

### 由 Redis 面试想到的

在面试后端工程师 Redis 技能的时候，面试官通常问的第一个问题就是“Redis 能用来做什么？”，第一个回答往往都会是「缓存」。缓存确实是 Redis 使用最多的领域，它相比 Memcache 而言更加易于理解、使用和控制。

可是如果再进一步问“还有呢？”，大多数同学就会开始皱眉头，只有一小部分人会回答「分布式锁」。如果你就分布式锁再深入问下去，他们基本就会开始摇头：我们项目里面 Redis 的锁方法都是别人（应该是架构师）封装好的，拿过来直接使用，内部细节没有去了解过，也没有必要了解。

对类似的场景，我深有体会。因为关于 Redis 的面试题，之前准备了很多，但是真正能用上的却很少。当面试的同学频繁地回复「不知道、没用过」的时候，再继续深入追问已经毫无意义，这时候就需要切换话题了。偶尔遇上几个能持续很多回合的同学，他们总能使人眼前一亮。如果再拓展一下周边知识点，就会发现这些人往往也会有所涉猎，这时我在心中已经暗暗地对这位同学伸出了大拇指。

这样的面试经历事后也让我深刻反思：架构师的技能很高，对提升团队研发效率很有帮助，我们非常钦佩和羡慕。但是普通开发者如果习惯于在架构师封装好的



东西之上，只专注于做业务开发，那久而久之，在技术理解和成长上就会变得迟钝甚至麻木。从这个角度看，架构师也可能成为普通开发者的“敌人”，他的强大能力会让大家变成“温室的花朵”，一旦遇到环境变化就会不知所措。



其实很多业务场景，如果仅仅是会使用某项技术、框架，那是再简单不过了。但随着业务发展，系统的用户量、并发量涨上来之后，现有系统的问题就会层出不穷地暴露出来。如果不能深入地了解系统、技术和框架背后的深层原理，很多问题根本无法理解到本质，更谈不上解决，临时抱佛脚也于事无补。

所谓「授人以鱼不若授人以渔」，本小册的初衷和目标就是帮助后端开发者较为深入的理解 Redis 背后的原理和实践经验，做到知其然也知其所以然，为未来进阶成长为架构师做好准备。

## 小册的内容范围

本小册主要讲解笔者从实战中摸索总结的 Redis 最常用最核心知识点，但限于篇幅和精力，并没有涵盖 Redis 全部的内容知识点，比如 Redis 内置的 Lua 脚本引擎就完全没有提到。之所以不讲，是因为在平时的工作中确实从来没有使用过，它就好比关系数据库的存储过程，虽然功能很强大，但是确实很少使用，而且也不易维护，所以就不推荐读者使用了。

对于很多小企业来说，本小册的很多内容都是用不上的，因为系统的并发量没有到一定的量级，这些高级功能根本没必要使用。不过机会总是留给那些有准备的孩子们，如果突然有一天流量涨上来了，Redis 的这些稀有的高级功能势必能立即派上用场。



读者们肯定也注意到，小册所有的标题都有使用特定的成语来描述。这些成语不是随便写的，而是精确考量了成语的含义和技术点的相关性精心挑选出来的，相信读者在理解了每个小节的内容之后，肯定可以明白内容和成语含义的相关性。之所以要使用成语也是为了制造悬念，吸引读者探究为什么这个技术点会和这个成语相关。老钱的语文水平不高，在选择成语时，反复使用了搜索引擎。如果读者找到了更贴切的成语，一定要即时在评论区留言告知。如果被采纳，会考虑福利反馈。😊

好了，深入理解 Redis 的学习之旅正式开始。

## Redis 可以做什么？

Redis的业务应用范围非常广泛，让我们以掘金技术社区（juejin.im）的帖子模块为实例，梳理一下，Redis 可以用在哪些地方？



1. 记录帖子的点赞数、评论数和点击数 (hash)。
2. 记录用户的帖子 ID 列表 (排序)，便于快速显示用户的帖子列表 (zset)。
3. 记录帖子的标题、摘要、作者和封面信息，用于列表页展示 (hash)。
4. 记录帖子的点赞用户 ID 列表，评论 ID 列表，用于显示和去重计数 (zset)。
5. 缓存近期热帖内容 (帖子内容空间占用比较大)，减少数据库压力 (hash)。
6. 记录帖子的相关文章 ID，根据内容推荐相关帖子 (list)。
7. 如果帖子 ID 是整数自增的，可以使用 Redis 来分配帖子 ID(计数器)。
8. 收藏集和帖子之间的关系 (zset)。
9. 记录热榜帖子 ID 列表，总热榜和分类热榜 (zset)。
10. 缓存用户行为历史，进行恶意行为过滤 (zset,hash)。

当然，实际情况下需求可能也没这么多，因为在请求压力不大的情况下，很多数据都是可以直接从数据库中查询的。但请求压力一大，以前通过数据库直接存取的数据则必须要挪到缓存里来。



以上提到的只是 Redis 的基础应用，也是日常开发中最常见的应用（如果你的 Redis 基础和经验不足，可能需要阅读完下一节之后才能回过头来思考这个问题）。除了基础应用之外，还有很多其它的 Redis 高级应用，大多数同学可能从未接触过，这部分我会在后续的章节中陆续讲解。

## 小结

接下来，我们将过一遍 Redis 的基础知识，这部分内容估计本小册大多数读者都已经非常了解，所以这里也不浪费太多笔墨，只计划用一节的篇幅快速讲完。如果读者对 Redis 基础数据结构已经了然于胸，可以直接跳到下一章节阅读 Redis 的高级知识。

另，在阅读小册过程中，如果你被某些章节卡住了，一下理解不了，可以先淡定的摸着胸口告诉自己“不要慌，一切都是正常的！”，然后临时跳过并继续阅读后面的章节。但请各位读者务必坚持到最后，相信你会明显感受到技术能力的升华。大家加油! 📖

## 扩展阅读

### 1. [《天下无难事之 Redis 面试题刁难大全》](#)

这篇文章是老钱之前就 Redis 面试总结的一篇原创分享，供大家参考阅读，这篇文章也被 IT 技术圈大号「高可用架构」转载。

### 2. [《Redis 作者 Antirez 其人趣事：为什么 Redis 的默认端口是 6379? 》](#)



Redis 由意大利人 Salvatore Sanfilippo (网名 Antirez) 开发，上图是他的个人照片。Antirez 不仅帅的不像实力派，也非常有趣。他出生在非英语系国家，英语能力长期以来是一个短板，他曾经专门为自己蹩脚的英语能力写过一篇博文《英语伤痛 15 年》，用自己的成长经历来鼓励那些非英语系的技术开发者们努力攻克英语难关。



我们都知道 Redis 的默认端口是 6379，这个端口号也不是随机选的，而是由手机键盘字母「MERZ」的位置决定的。「MERZ」在 Antirez 的朋友圈语言中是「愚蠢」的代名词，它源于意大利广告女郎「Alessia Merz」在电视节目上说了一堆愚蠢的话，想到这里我不禁开始感觉到 Antirez 的朋友圈似乎有那么点猥琐。😏



Antirez 今年已经四十岁了，依旧在孜孜不倦地写代码，为 Redis 的开源事业持续贡献力量。

下面两篇文章是老钱在翻阅了 Redis 作者 Antirez 的早期博客之后，挑了几个有趣的故事做了翻译，也分享给广大读者们。

3. [《我为 Redis 找到了一个新家——VMWare》](#)
4. [《Redis 作者 Antirez 经历的「性别歧视」风波》](#)