

Dive Into Distributed Cache System
principles, frameworks and best practices

深入分布式缓存

从原理到实践

于君泽 曹洪伟 邱硕 ©等著

来自蚂蚁金服、京东、网联、新浪微博、同程旅游等公司的
10位一线架构师用心之作

深度解构分布式缓存技术原理及其在电商、社交、广告等
典型场景中的应用



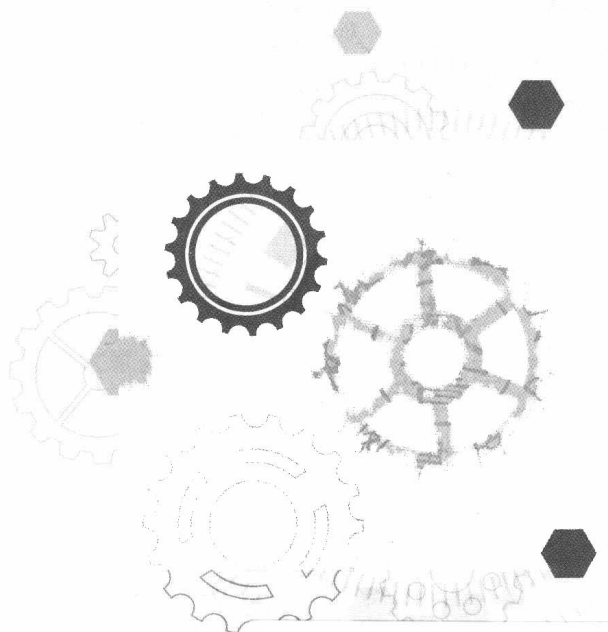
机械工业出版社
China Machine Press

Dive Into Distributed Cache System
principles, frameworks and best practices

深入分布式缓存

从原理到实践

于君泽 程超 邱硕 曹洪伟 刘璟宇 著
张开涛 何涛 宋慧庆 陈波 王晓波



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

深入分布式缓存: 从原理到实践 / 于君泽等著. —北京: 机械工业出版社, 2017.11 (2018.3重印)

ISBN 978-7-111-58519-0

I. 深… II. 于… III. 互联网络—网络服务器 IV. TP368.5

中国版本图书馆 CIP 数据核字 (2017) 第 282885 号

深入分布式缓存：从原理到实践

出版发行：机械工业出版社（北京市西城区百万庄大街 22 号 邮政编码：100037）

责任编辑：孙海亮

责任校对：殷虹

印刷：中国电影出版社印刷厂

版次：2018 年 3 月第 1 版第 4 次印刷

开本：186mm × 240mm 1/16

印张：26.25

书号：ISBN 978-7-111-58519-0

定价：99.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

客服热线：(010) 88379426 88361066

投稿热线：(010) 88379604

购书热线：(010) 68326294 88379649 68995259

读者信箱：hzit@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问：北京大成律师事务所 韩光 / 邹晓东

本书围绕分布式缓存的基础概念、开源框架、应用案例三方面进行讲解，从理论到实战，循序渐进，深入浅出。看完部分章节后，意犹未尽，欲罢不能。国内基于真实应用案例的好书太少了，君泽以及他的朋友们做到了这一点，十分期待这本书能早日上市。

——黄勇 特赞科技 CTO、《架构探险》作者

分布式缓存，是任何一个互联网公司在成长过程中都会面临的技术难题。本书作者结合理论研究和长期的互联网行业从业经验，深入浅出地介绍了分布式系统理论和分布式缓存实战，给业界以借鉴和启发。本书是作者们的用心之作。

——朱攀 德比软件架构师

缓存是软件性能优化的大杀器，分布式缓存是网站架构的必杀技，玩转缓存就玩转了网站架构的半边天。遗憾的是，目前市面上专门讲述分布式缓存的书籍不多，幸运的是本书就是一本这方面的专著。本书所有作者都是多年工作在网站架构一线的老司机，值得信赖，故将本书推荐给大家。

——李智慧 《大型网站技术架构：核心原理与案例分析》作者

从十几年前的 Ecache 到最近几年流行的 Redis，从 CDN、浏览器、API Gateway 到后端微服务，以及数据访问层的二级缓存，缓存无处不在。在体验为王、唯快不破的时代，分布式缓存是关键。本书从理论到实践，详细剖析了分布式缓存的实现原理以及应用案例，是一本接地气的好书。

——李林锋 华为 PaaS 平台架构师、公司总裁技术创新奖获得者

随着现代应用对速度的要求越来越高，对缓存机制的使用也越来越常见、越来越频繁。本书以缓存机制的基本原理为开始，逐渐过渡至缓存系统的组建以及使用上面，全书分析和讲解了多个缓存系统，并列举了缓存的各种使用场景。如果你正准备构建自己的缓

存系统，又或者你想进一步学习更多与缓存有关的知识，那么这本书将是你不容错过的一本书。

——黄健宏 《Redis 设计与实现》作者

在计算机的世界里，不论硬件层面还是软件层面，缓存都被广泛应用于解决处理响应慢的瓶颈。我们在构建微服务架构系统的时候也一样，缓存是提升性能的关键技术手段。然而，缓存在不同场景下的应用各有不同，要想用对缓存、用好缓存并不容易。本书针对不同的缓存类型、实现手段、算法策略做了非常细致的讲解，所以我推荐开发者和架构师们通过本书来对缓存做一次全面的学习，这有助于更好地使用缓存来优化我们的系统性能。

——翟永超 《Spring Cloud 微服务实战》作者

不同的存储介质，读写性能有很大的差异，价格亦是如此，性能越好的介质，价格就越高。把最常访问的数据放在读写性能最好的设备上，达到成本和性能的均衡，这便诞生了缓存。本书的几位作者都是大型分布式环境下历练出来的沙场老将，丰富的经验和翔实的案例，对于互联网行业的技术人员来说，颇有裨益。

——陈康贤 《大型分布式网站架构设计与实践》作者、阿里巴巴技术专家

如今的软件系统，已经成为分布式系统的天下。分布式的语境对系统的设计与开发提出了完全不一样的挑战，作为提升性能的关键要素——缓存，自然也不例外。在缓存的命中、更新与失效，以及数据一致性保障等诸多方面，分布式缓存应用的复杂度可能是按指数级增加的，许多问题都是我们在单机应用上无法涉猎的。倘若没有分布式系统设计的实战经验，面对分布式缓存的诸多问题，我们将束手无策。求助于网络？讲解缓存知识的文章是片鳞半爪，未成体系，故而无法窥其全部。本书的几位作者都经历过大型软件系统的深度洗礼，书中内容都来自他们的一线实战。阅读本书，读者就能站在他们的肩膀上看得更远，实在是幸运！

——张逸 民航信息技术架构师、《恰如其分的软件架构》译者

缓存是当前互联网的基石，极大弥补了关系型数据库的天然缺陷。缓存技术也是每一个架构师的必修课，从最初的 Memcached，到 Redis 及各种扩展架构，发展至今已经有些百花齐放的感觉。本书难得地进行了全面梳理，并附以在电商、社交、广告等典型场景下的大型应用。你值得拥有。

——萧田国 高效运维社区发起人、DevOpsDays 中国联合发起人

缓存是互联网架构中最关键的环节，本书作者在缓存的性能及高可用方面有丰富的经验，很高兴看到他们通过本书分享了多年的实践精华。

——杨卫华 微博研发副总经理

互联网高性能系统设计的核心之一就是缓存系统的设计。本书集合了缓存理论、开源缓存系统、大规模业务中缓存的具体实践，全方位解读了分布式缓存设计，既能够帮助读者深入理解分布式缓存系统，又提供了很好的架构设计案例供参考，细细品读，受益良多！

——李运华 阿里游戏资深技术专家、《面向对象葵花宝典》作者

推荐序 1 *Foreword*

缓存技术的大规模使用是互联网架构区别于传统 IT 技术最大的地方，对缓存的理解和使用的深度决定了是否能架构出一个高性价比、高扩展性的系统。阿里巴巴的系统使用了大量缓存技术（内部缓存的产品名字叫 Tair）。从 2007 年开始，缓存服务器的增长速度远远超过了数据库服务器的增长速度，而因为对缓存的充分使用，系统具备的吞吐量能力的增速又远远高于缓存本身规模的增速。这是一件非常划算的事情，成本下降非常明显，大概使用 1 台缓存服务器就可以完成以前 4 台服务器的工作，从而为整体集群节省 3 台服务器。

缓存代表了高性能的一项技术，同时缓存也是系统架构里非常核心的部分，一个系统缓存承担着 90% 以上的热点流量，缓存出一点点问题，系统的可用性会马上受到影响。在阿里巴巴的系统架构讨论会上，一般都会把做缓存产品的同事叫上，要求其了解系统架构规模的前提下给出流量规模、分区方案、热点节点等方面的建议。负责缓存产品的同事往往也需要对业务有所了解，否则就不能设计出更好的产品。缓存的使用也使得架构更加清晰、更容易理解，流量走向变得更简单，这使得系统持续可用性有了明显的提升。

本书比较系统地介绍了缓存在各个层面的工作原理、作用和价值。理解了缓存的相关知识，会对架构一个高性能分布式系统有很大的帮助。

——小邪（蒋江伟）阿里研究员

无处不在的缓存，空间换时间的艺术

Cache 这个词，据说来自于法语，本意是当 CPU 处理数据时，先到 Cache 中去寻找，如果数据在之前的操作已经读取而被暂存其中，就不需要从主内存（Main memory）中读取。后来 Cache 的范围有所扩大，凡是位于速度相差较大的两种硬件之间，用于协调两者数据传输速度差异的结构，都称为 Cache。

现在，我们谈的缓存，已经远远突破了原有的概念，在我看来，缓存是一种通用的设计模式：这种模式利用增加存储空间的方式，实现低速部件与高速部件之间的解耦。换言之，只要实现了解耦的地方，就有存在缓存的可能，既然解耦是计算机系统架构设计中最常用的手段，那么缓存就必将在计算机系统中无处不在。缓存用空间换时间的方式解决问题，而空间不可能无限使用，使用缓存时我们通常会考虑如何选择存储方式、如何使用多级缓存节省空间、如何有效提高缓存的命中率、如何确定有效的更新策略等问题，这些思考会有相当普遍的适用性。

本书重点解读了分布式系统下如何使用缓存。分布式系统中大规模使用缓存是从早期互联网三大法宝（MemCache、反向代理、分库分表）开始的，利用这种集中式、大规模的缓存技术，我们解决了数据库低速 IO 与高速应用之间的矛盾。我们也发现，除了缓存的通用问题之外，分布式系统缓存还需要解决更多的问题，例如如何对多种类型数据选择不同的存储方式、如何保证数据一致性、如何提高缓存自身的可用性、如何增强系统的可维护性。为了系统性解决这些问题，分布式缓存逐渐变成了分布式架构系统中的一个基础设施。

缓存的使用虽然广泛，但把缓存作为一种基础设施的发展历史并不长，大多数的商业化软件比较复杂，主要以内存数据库为主，而轻量型分布式缓存存在不同场景下有不同的关注点和使用方式。君泽集合了一批老司机，为大家展示了多种分布式系统中缓存的使用场景与方案，既有新浪微博这样重量级社交平台信息聚合、分发缓存的方法，也有海量广告

业务信息撮合的缓存模式；既有电商系统冷热数据分离的缓存策略，也有对完整分布式系统缓存的技术选型和总结。

通过分布式、微服务架构，实现业务的云化 / 数字化，建立起的应用生态是一个复杂的体系，其中方方面面的著作很多，但专注在缓存方面的还没有，本书选择了这个方向，就是希望能帮助我们形成一个完整的技术视图，我相信这是作者们做这件事的初衷。

——焦烈焱 普元信息 CTO

缓存为王

君泽人很低调也很友善，第一次和他见面还是在 2016 年的年底。2017 年的春夏之交，当我正在机场为飞机晚点心烦不已的时候，收到了君泽的一条微信。内容是说他自己正在忙着准备一本书，而且把主要的章节内容发给了我。在百无聊赖之间我就读了起来。

这本书是君泽组织国内互联网技术领域具有实战经验的部分专家，分工合作联合撰写的技术专著。以最常用、最有效果也最容易出问题的缓存为主题。从基础概念、开源框架和应用案例三个层次进行了讨论。给我的感觉是既有理论也有实践，既有广度也有深度，既有国外的开源软件也有国内的具体实践案例，是非常值得深入阅读和细心揣摩的一本工具书。

给我印象比较深刻的是开篇的“缓存为王”，因为今年年初我也专门写过一篇同名的文章，论述了从网络靠近用户端的 CDN，到内部网络缓存，到应用缓存，再到数据库缓存的一系列缓存手段、技术和方法。不敢说英雄所见略同，至少大家的关注点差不多。我还记得在当初的文章中用了一句耐人寻味的话，“解决雪崩问题的最好办法是不发生雪崩”。不论是在硅谷互联网公司里还是在国内的互联网平台上，曾多次遇到过海量规模的交易瞬间吞噬平台的悲惨故事。核心的解决方案大同小异，都是通过缓存，逐层减少流量的冲击，保护用户的体验和平台的服务可用。金融、财务行业有现金为王的说法，互联网技术行业用“缓存为王”清楚地概括了缓存的重要性。

这本书还深入浅出地讨论了 Ehcache、Memcached、Redis、Tair、EVCache 等各种常用缓存开源系统的方案及精粹。我读过不少国外有关缓存的技术专著，大多数偏原则和理论，具体实施方案和案例较少，特别是结合中国互联网技术实践的更少。所以君泽组织的这本书实际上填补了缓存技术文献在这方面的空白。尤其是结合微博、社交和电商平台的应用实践探讨，对想学习如何利用缓存技术解决实践中具体问题的读者大有裨益。

——陈斌 易宝支付 CTO

作者寄语 *Foreword*

我在这部书里，负责编写了 Tair 章节，合作编写了缓存周边、动手写缓存章节，期间阅读了大量的代码、整理了非常多的资料，希望能带给读者需要的知识。在一年多的时间里，工作之余的大部分时间，都投入到了书的编写中。在这个过程中，家人给了我极大的支持，这里特别感谢我的太太李春花。一年多的时间里，有因为大促工作量激增，有因为工作变动导致工余时间减少，感谢右军持续的鼓励，感谢一同编写本书的其他伙伴及时的帮助！

——刘璟宇 拍拍贷资深架构师

很幸运，从 2009 年新浪微博发展之初就参与微博平台系统的研发及架构工作，经历了微博从起步到当前月活用户数亿的大型互联网系统的技术演进过程。系统演进中经历了很多曲折、困难、不眠之夜，多亏了 @TimYang、@zhulei、@liudaoru 等那么多良师益友，一路上大家志同道合披荆斩棘，回头看经历的种种困难却正好是不断进步的印记。最后感谢 @右军的邀请和大力组织，感谢 @Mis 晓晓 的包容和支持，正因为有了你们，我才得以完成相关内容的编写和完善。

——陈波 新浪微博平台架构技术专家

从事数据访问层工作期间，会收到很多缓存方面的技术支持，在支持的同时，对缓存也有了更深入的了解，也促使我思考如何将这些实践经验分享给更多的人。正好右军邀请参与本书的写作，便欣然同意。写作期间，阅读了大量的源码，并和联合作者相互交流缓存的原理及实践经验，力求能够呈现实用的技术。本书涉及的分布式缓存体系非常全面，从分布式、Redis、Memcached 等原理的剖析到大量的一线实践案例，并且对分布式缓存应用当中的一些痛点、难点进行了深入的阐述。希望本书能够为想了解缓存技术的读者带来快乐和收益。

——何涛 唯品会架构师

承右军兄邀请，有幸参与本书第 11 章“Aerospike 原理及广告业务应用”的编写。本人虽然在互联网广告行业摸爬打滚十余年，从广告网络到 SSP、DSP，再到苹果应用市场的推广 (ASO,ASM)，从 PC 广告到移动原生广告都有所涉及。对于广告系统架构搭建 (存储，计算，缓存) 以及系统的高可用较熟悉之外，对于写书确实是赶鸭子上架头一回。又由于平时工作繁忙，家里琐碎事情亦多，几有放弃之念想，但此事绝非君子所为。若如此辜负右军兄及其他几位一同写书的弟兄，于心不忍，日后亦无颜面对他们。故硬着头皮，把此章写完并交付出版社。听闻此章定稿之后，感觉如释重负，精神亦有所高涨。当然，能够坚持下去的另一个重要原因是，其他各位参与编写本书的弟兄的鞭策和鼓励。在这里要特别感谢右军和老曹。由于本人水平有限，还望各位亲爱的读者海涵。

——宋慧庆 勤诚互动高级架构师

首先非常感谢右军兄邀请我加入写作团队，我主要负责第 3 章、第 4 章，同时参与合作编写了第 6 章。虽然我自己经常写一些博客文章，但是写书和写博客是完全不一样的过程，写书是要能够带领读者逐步深入学习，而不是简单地将知识点罗列出来。在这个过程中非常感谢右军和孙海亮对我进行认真的指导，让我的写作水平有了非常大的提高。当我第一篇稿子定稿的时候，我内心的成就是很难用言语来表达的。由于本书的写作时间周期比较长，在中途的过程中由于工作繁忙，我也曾经有退出的想法，右军和曹哥给了我巨大的鼓励，让我学会了坚持，同时我的妻子和我的父母也给了我巨大的支持，让我能够学会与团队协作。对于我来讲这本书对我的意义已经超过内容本身的范畴，我也希望通过这本书能够给我的孩子树立一个贵在坚持和勇于尝试的榜样。最后由于本人写作水平有限，还请各位读者朋友海涵和包容。

——程超 爱农驿站首席支付技术专家

Web 应用相对于传统的软件，在服务的吞吐量方面有更高的要求，也是其面临的主要难点之一。对于无状态的应用服务，日益增加的吞吐量最终转化成存储层的压力，使得后者既要解决数据一致性问题，又要考虑性能和吞吐量，成为 Web 应用性能的关键，因此已有的文档和书籍主要聚焦在它的优化上。

而缓存作为应用服务节点和持久化存储节点间的辅助层，分担着后者的吞吐量需求，好的缓存设计可以极大地降低存储层的容量风险。本书从缓存的原理、实现到缓存在不同场景的实施方案给出系统化的介绍，为“好的缓存设计”提供了指导。因此在受邀参与本书编写时，既感到荣幸，又感到意义重大。

好的写作依赖于热情和投入，而书籍的编写又需要长时间的毅力和坚持，这让本书的编写像是一段长跑里需要不断冲刺，也让这份“意义重大”既是之于内容本身，也是之于作者的自我提升。感谢本书主笔于君泽的坚持，也感谢其他每一位作者的投入，他们鞭策着我在此领域学习和思考。限于自己的水平，谨希望所编写的章节能为读者们带来一点帮

助。内容有欠缺之处，也希望读者批评指教，以此驱动这个领域的应用和实践持续演进。

——邱硕 蚂蚁金服技术专家

作为一个 70 后的老码农，一个半吊子全栈工匠，在中生代技术（freshman Technology）社区有幸认识了很多志同道合的技术人。因为自己在公众号（wireless_com）上的一篇关于缓存基础的随笔而结识了 @ 右军，进而参与了本书的创作。在如今用户体验至上的时代，性能成为系统设计中的一种核心约束，在性能提升的各种技术手段中，缓存为王。随着技术的演进，缓存同样与时俱进。参与编写的过程是一个对经验梳理的过程，同时也是一个提升的过程，尤其是和各位作者的切磋，获益匪浅。每一位作者都牺牲了大量的闲暇时间，为本书的出版付出了巨大的努力。我感谢自己的妻儿，感谢家人对自己的支持，希望这本书能够不负众望。

——曹洪伟 渡鸦科技 CTO

2014 年加入京东后，负责重新设计详情页架构，它是一个读服务，从前端浏览器到后端存储无处不用缓存，在实战中通过运用大量缓存技术提升性能、解决棘手问题，比如 618 期间有人来刷你的接口，简单上个缓存，会起到很大的保护作用，而且效果非常好。在运用缓存时有许多需要注意的地方，比如缓存一致性是否需要强一致；价格库存数据能否缓存，缓存多久；缓存分布算法是使用一致性哈希还是取模算法；热点数据怎么处理；缓存崩溃与快速恢复等等。要用好缓存并不是那么容易的，希望读者看完本书后能学到一招半式并应用到实战中。

——张开涛 京东架构师

缓存在整个高并发架构设计中是重中之重的关键一笔，所以用好缓存是每次架构设计的必经之路。感谢右军邀请我参与本书。我在本书中介绍了同程旅游的凤凰缓存系统（phoenix）是如何云化管理同程全部的缓存集群，并治理各应用中的缓存使用的。也讲述了一些我们为何要开发 phoenix 来解决缓存问题的坑事。回首这些坑事，历历在目。各种困难、各种痛苦伴随着凤凰缓存系统从构想到多个版本的更新上线。希望本书能够给踩到同样坑事的读者带来帮助和快乐。

——王晓波 同程旅游首席架构师

为了初心的纪念

一本历时 2 年的书即将出版难免有些激动，同时亦有些忐忑。激动在于这是一次有意义的社会实践，诸多作者参与其中。至完稿时，作者与发起计划时已有些不同。一些朋友因为各种原因退出了，所以从坚持的角度看，完成就意味着第一层面的成功。忐忑在于从初心出发，通读全稿仍有不足之处，诸位作者在具体实践中所遇到的线上问题远非本书所能尽数容纳。而最后精华的实践章节亦经过公司 PR 审核，略有删减。再一层，设计和编码很重要，运维管理同等重要。一个好用、易用、稳定的运维工具也是选择一套开源组件的重要参考因素，为了集中注意力，本书基本未覆盖运维部分的内容，且待有心人续之。写一本书，面世之后，作者或喜或忧，因为臧否之权利在于读者。

自觉而言，本书适合有一定研发经验的朋友阅读，它山之石，亦有攻玉之效。本书在逻辑上可分为三大篇章：基础概念篇、开源框架篇、应用案例篇。基础概念除了基础知识，也介绍了一些分布式方面的方法和思路；开源框架篇遴选了近年来流行的框架（比如 Redis），同时对淘宝 Tair、EVCache 也做了一些探索。在 Redis 大行其道之时，对于 Memcached 及其周边知识也做了介绍，某些公司还有大量的 Memcached 实例，比如微博、Twitter 等。工具的革新总是源自需求的不断被满足，而根据被满足的特性可以归纳其共性，比如解决单点高可用问题就是一个普适性问题，涉及主从模式、双活模式等，可用性同时又和性能、数据一致性相关。缓存为性能而生，但“缓存”设施的存在就决定了这个设施要符合分布式理论的要求。业界介绍理论和概要，或介绍设计原则的书不少，但拿出具体实践的稀有，比如新浪微博、Twitter 这样的社交 SNS 具体如何设计缓存。简约而不简单！在应用案例篇，笔者邀请了对应领域的专家为大家解读案例，可以让大家触摸到真实的设计意图。重要的是大家可以获得不同场景下不同设计策略的启发。

本书的产生要追溯到多年前。笔者一直对缓存技术抱有热情，关注开源框架的发展，亦在工作中关注所遇、所见，乃至所听的案例。从应用程序研发方面看分布式缓存，并不需要所有的程序员都具备开发一套组件的能力，但是需要具备正确使用它的能力。正如易

宝 CTO 陈斌老师所言：“解决雪崩问题的最好办法是不发生雪崩”。不论是在硅谷互联网公司里还是在国内的互联网平台上，曾多次遇到过海量规模的交易瞬间吞噬平台的悲惨故事。笔者亦了解一些缓存因为代码缺陷或者使用不当被击穿的案例，不同数量级的请求产生的结果有天壤之别，不可不慎。

两年前偶遇机械工业出版社的杨福川老师，攀谈之下就萌发了创作本书的念头。但由于工作繁忙且想呈现心中所想之提纲，故邀请一些不同场景下的专家共同完成。组团过程多有波折，特别感动的是北京的孔庆龙兄。他非常有兴趣参与合作，但时逢小孩即将出生，为此，孔兄开了一次家庭会议来讨论此事。虽然孔兄后续未决定参与，但可见其待人之真、之诚，是值得交的朋友。两年间发生了不少事情，刘璟宇 (leo)、何涛、曹洪伟和程超都换了工作。在本书项目开始时程超家的小朋友还未出生，现在都快 2 岁了。大家都很忙，大约 1 个月碰一下进度，有时候可能一点进展都没有。期间，程超和 leo 都一度要退出，终坚持了下来。还有些朋友中间退出了，同时有陈波、王晓波等朋友加入。到这时，啥时候出版已不那么心焦了，水到渠成。就是问初心，我们有没有尽自己的努力来呈现一份关于工具书的纪念？

特别感谢曹洪伟身体力行，按时按质完成了分配的章节，同时 review 了不少章节，带动了虚拟组织迈上一个新台阶。这是一本书的编写过程，是一次心灵的旅行，是一次基于互联网的跨组织协同，也是关于工作和生活的点缀。感谢妻子晓娜对于我写作的支持，她对于进度的关注甚于我自己。她花了大量的时间辅导两个小朋友的学习，虽然周末有辅导班、家务等各种事项，但我总能申请到一些时间来构思本书。本书也是送给我的两个宝贝的礼物。这个礼物是初心，想到什么有意义的事情要义无反顾去做；同时，这个礼物也代表“坚持”，如同写作文一样，观察、总结，然后不断练习。

同时还要感谢所有的合作者，我们交叉 review 了所有章节。感谢机械工业出版社的杨福川老师、孙海亮老师。孙老师的审阅非常专业，从读者视角直面问题所在。朱攀兄弟 review 了部分章节，在写作过程中和高磊兄弟、丁浪兄弟也有交流，一并表示感谢。

囿于篇幅和知识局限，错漏之处难免，后续将通过勘误和调整完善的形式持续优化。为了初心的纪念，一次心灵的旅游！祝大家阅读愉快！

于君泽（右军）



扫二维码联系小助手加群，与作者交流或提交勘误

Contents 目 录

赞誉	
推荐序 1	
推荐序 2	
推荐序 3	
作者寄语	
前言	
第 1 章 缓存为王	1
1.1 什么是缓存?	1
1.2 为什么使用缓存?	2
1.2.1 从用户体验说起	3
1.2.2 关于系统的性能	3
1.3 从网站的架构发展看缓存	4
1.4 客户端缓存	5
1.4.1 页面缓存	6
1.4.2 浏览器缓存	7
1.4.3 APP 上的缓存	8
1.5 网络中的缓存	11
1.5.1 Web 代理缓存	11
1.5.2 边缘缓存	12
1.6 服务端缓存	14
1.6.1 数据库缓存	14
1.6.2 平台级缓存	16
1.6.3 应用级缓存	18
第 2 章 分布式系统理论	24
2.1 分布式系统概论	24
2.2 分布式系统概念	26
2.2.1 进程与线程	26
2.2.2 并发	26
2.2.3 锁	26
2.2.4 并行	27
2.2.5 集群	27
2.2.6 状态特性	28
2.2.7 系统重发与幂等性	28
2.2.8 硬件异常	30
2.3 分布式系统理论	31
2.3.1 CAP 理论	32
2.3.2 CAP 理论澄清	34
2.3.3 Paxos	35
2.3.4 2PC	38
2.3.5 3PC	39
2.3.6 Raft	40
2.3.7 Lease 机制	41
2.3.8 解决“脑裂”问题	43

2.3.9	Quorum NWR	44	4.3.1	集群的方式	86
2.3.10	MVCC	45	4.3.2	如何配置集群	88
2.3.11	Gossip	46	4.4	Ehcache 的适用场景	89
2.4	分布式系统设计策略	49	4.5	Guava Cache 的使用	92
2.4.1	心跳检测	50	4.5.1	Guava Cache 的适用场景	92
2.4.2	高可用设计	50	4.5.2	Guava Cache 的创建方式	93
2.4.3	容错性	52	4.5.3	缓存数据删除	95
2.4.4	负载均衡	53	4.5.4	并发场景下的使用	95
2.5	分布式系统设计实践	54	4.6	本章小结	96
2.5.1	全局 ID 生成	54	第 5 章 从 Memcached 开始了解		
2.5.2	哈希取模	56	集中式缓存		
2.5.3	一致性哈希	57	5.1	Memcached 基本知识	98
2.5.4	路由表	58	5.1.1	Memcached 的操作命令	98
2.5.5	数据拆分	58	5.1.2	Memcached 使用场景	100
第 3 章 动手写缓存			5.1.3	Memcached 特征	100
3.1	缓存定义的规范	60	5.1.4	Memcached 的一些问题	101
3.1.1	新规范的主要内容及特性	60	5.2	Memcached 内存存储	102
3.1.2	新规范的 API 介绍	61	5.2.1	Slab Allocation 机制	102
3.2	缓存框架的实现	62	5.2.2	使用 Growth Factor 进行调优	104
3.2.1	前期准备	63	5.2.3	Item	105
3.2.2	缓存的架构介绍	63	5.3	典型问题解析	106
3.2.3	设计思路以及知识点详解	64	5.3.1	过期机制	106
3.3	缓存框架的使用示例	74	5.3.2	哈希算法	107
第 4 章 Ehcache 与 Guava Cache			5.3.3	热点问题	108
4.1	Ehcache 的主要特性	76	5.3.4	缓存与数据库的更新问题	108
4.2	Ehcache 使用介绍	77	5.3.5	别把缓存当存储	109
4.2.1	Ehcache 架构图	77	5.3.6	命名空间	110
4.2.2	缓存数据过期策略	78	5.3.7	CAS	110
4.2.3	Ehcache 缓存的基本用法	81	5.4	Memcached 客户端分析	110
4.2.4	在 Spring 中使用 Ehcache	83	5.4.1	Memcached 的 Client	111
4.3	Ehcache 集群介绍	85	5.4.2	Spymemcached 设计思想解析	111