

Spring Cloud Alibaba

微服务原理与实战

谭锋 (Mic) ©著



系统化的深度整理
高效的技术学习模型

7



Spring Cloud Alibaba

微服务原理与实战

谭锋 (Mic) ©著



電子工業出版社

Publishing House of Electronics Industry

北京•BEIJING

内 容 简 介

本书针对 Spring Cloud Alibaba 生态下的技术组件从应用到原理进行全面的分析，涉及的技术组件包括分布式服务治理 Dubbo、服务配置和服务注册中心 Nacos、分布式限流与熔断 Sentinel、分布式消息通信 RocketMQ、分布式事务 Seata 及微服务网关 Spring Cloud Gateway。由于 Spring Cloud 中所有的技术组件都是基于 Spring Boot 微服务框架来集成的，所以对于 Spring Boot 的核心原理也做了比较详细的分析。

本书中涉及的所有技术组件，笔者都采用“场景→需求→解决方案→应用→原理”高效技术学习模型进行设计，以便让读者知其然且知其所以然。在“原理”部分，笔者采用大量的源码及图形的方式来进行分析，帮助读者达到对技术组件深度学习和理解的目标。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有，侵权必究。

图书在版编目（CIP）数据

Spring Cloud Alibaba 微服务原理与实战 / 谭锋著. —北京：电子工业出版社，2020.4
（咕泡学院 Java 架构师成长丛书）
ISBN 978-7-121-38824-8

I. ①S… II. ①谭… III. ①JAVA 语言—程序设计 IV. ①TP312.8

中国版本图书馆 CIP 数据核字（2020）第 047380 号

责任编辑：董 英

印 刷：三河市华成印务有限公司

装 订：三河市华成印务有限公司

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：25.5

字数：588 千字

版 次：2020 年 4 月第 1 版

印 次：2020 年 4 月第 1 次印刷

印 数：6100 册 定价：106.00 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。

作者简介



谭锋 (Mic)

咕泡学院联合创始人，拥有12年Java开发及架构经验，主要关注微服务、高并发及容器化等技术领域，纯粹的技术爱好者。

曾就职于平安支付、挖财，担任业务架构师。2013年参与并主导Dubbo服务化落地项目。2015年负责公司内部Spring Boot集成Dubbo实现微服务化落地。2017年成立咕泡学院，目前主要负责咕泡学院Java架构进阶课程的研发和设计。



打造技术人的终身职业交流圈

咕泡学院是极具影响力的中高端IT技术在线教育品牌。聚集国内多家一线互联网企业技术大牛，精心研发了课程体系。内容紧跟IT行业潮流，理论通俗易懂，并结合实际应用场景讲授经典设计思想，让技术人从基础知识到实战能力全面提升。凭借超高口碑和品牌影响力，咕泡学院曾荣获“2019腾讯课堂年度卓越贡献奖”“腾讯回响中国年度影响力在线教育品牌”“腾讯课堂·创造101火箭机构奖”“腾讯课堂·金课堂奖”等奖项。



让每个人的职业生涯不留遗憾

推荐序 1

近几年来，我们大家都体会到了移动互联网带来的便利，企业在应对外部环境的快速变化时需要快速创新。无论是传统企业还是互联网企业，都需要不断地对 IT 架构进行升级和改进，从而支持企业的数字化转型。

说到微服务架构，想必大家都不陌生，尤其在互联网应用中谈到企业应用架构时，微服务架构是当前必聊的话题。微服务架构是从单体架构、垂直架构和 SOA 架构逐渐演变而来的。微服务架构这么火热是因为相比之前的应用架构它有许多优点，例如更快速、灵活，更能适应现在需求变更快速的大环境。

Spring Cloud Alibaba 是一套完整的微服务架构解决方案，它为开发人员提供了一些工具来快速构建分布式系统，这对于中小型互联网公司来说是一种“福音”，它能帮助企业在应对业务发展的同时，大大减少开发的成本。

Spring Cloud Alibaba 于 2019 年 8 月开源以来，受到国内诸多企业和开发者的青睐，但在市面上缺少对其系统性的介绍和使用方法的相关内容。本书作者结合自己多年对互联网技术及微服务架构的理解和实践经验，从入门示例、原理剖析和源码分析等各个方面整理了每个模块的知识。无论你是刚入门的 Java 开发者，还是从事开发工作多年的资深码农，都会受益匪浅。

我是阿里巴巴淘系技术部的一名技术小二，很高兴能参与其中推广公司的开源技术。希望本书能够帮助大家快速理解和掌握微服务架构，通过技术的力量帮助企业快速成长。当然，我也希望更多的同学能够加入阿里巴巴。

汤陈

阿里巴巴高级开发工程师

推荐序 2

近年来，随着互联网技术的蓬勃发展，微服务理念逐渐深入人心。围绕微服务的讨论，一刻也未曾停止。而关于“Dubbo VS Spring Cloud”的技术选型，是其中一个“长盛不衰”的热门话题。大家普遍认为，Dubbo 是一款具备一定服务治理能力的 RPC 框架，而 Spring Cloud 是一整套微服务生态体系，双方的比较并不在对等的维度上。

Dubbo 由阿里巴巴于 2011 年开源，很快发展成为国内最火爆的 RPC 框架，其经典程度毋庸置疑。Spring Cloud 的出现，打开了微服务技术的新局面，全家桶式的解决方案，给广大开发者带来了极大的便利。Spring Cloud 制定了集成标准，允许开源社区来提供组件的实现。Netflix 公司在其中扮演了重要的角色，贡献了 REST RPC 框架、服务注册中心、服务网关、断路器、客户端负载均衡等多个服务治理组件。

2017 年，阿里巴巴“重启”Dubbo 的维护，不久就推出了支持 Spring Boot 的版本。而随后 Spring Cloud Alibaba 的问世，不仅宣告了拥抱 Spring Cloud 标准，也与 Netflix 系的解决方案站在了对等竞争的位置。对于很多偏好 Dubbo 的国内开发者来说，面对补齐的微服务生态，无疑又多了一个坚定选择它的理由。Spring Cloud Alibaba 的优势绝不仅仅在于“更服国内的水土”，其脱胎于内部中间件、在多年海量业务场景下打磨出的微服务生态组件，技术含量与工业成熟度在全球范围内也是极具竞争力的。

本书作者谭锋在微服务架构方面具备丰富的实战经验，并不断将自己所学所想分享给职场小伙伴们，本书是他的实战经验总结，推荐给大家。另外，也欢迎大家来“PerfMa 应用性能技术社区”交流。

李嘉鹏（你假笨/寒泉子）

PerfMa CEO

推荐序 3

我曾带领团队从 2016 年开始调研 Spring Cloud 的整体生态，在了解到其完整的一体化解决方案后，决定将其引入公司做成一套全新的研发框架。由于我当时所在的是一家已经成立 15 年以上的老牌互联网公司，其内部的研究架构、运维体系等都向这次转型提出了“挑战”。不同于从 0 开始搭建，我们对 Spring Cloud 的组件进行了大量的二次开发工作，目的是能更平滑地完成这次转变。比如，在跨系统交互时，我们依然保留着一些通过 api-gateway 进行通信的方式；在微服务的运维部署上，公司内并存着容器化（既有 Mesos 也有 Kubernetes）和虚拟机部署的方式等。在进行这些工作时，给我感触最深的就是，单纯地使用 Spring Cloud 整个生态的技术，已经需要一定的技术素养和研究成本。而在其基础上进行二次开发，就会面临参考资料不足、文档和书籍匮乏等问题，不得不花上大量的精力去研究源码。

用一句话形容程序员的进阶，那就是“从说 IT 术语，变成说人话”，这里当然有夸张的成分。我与 Mic 老师既是老同事，也是老朋友。我从他的身上，很明显地感受到他已经从一位大牛架构师，变成了一位“不但能讲人话”，还能“授人以渔”的优秀讲师。本书的目录编排、讲述方式，深谙程序员所求。对“实用主义者”，它直接“show you the code”；对“底层探究者”，它又会进阶地补充上一些原理性的内容，增强理解。相信无论你是新接触 Spring Cloud Alibaba，还是和我当时一样正在团队推进技术转型，这本书都能为你提供很好的帮助，减小你“踩坑”的概率。

目前国内的开源项目越来越多地在开源社区发光发热，Apache 基金会的顶级项目中，也有越来越多的“中国制造”。感谢 Mic 老师这本“干货”教材对 Spring Cloud Alibaba 的推广，为更多的 IT 开发者、IT 研发团队赋能增效。

顾冬煜

赋优信息技术 CTO、沪江网前技术总监

前 言

我看过市面上很多关于分布式和微服务的书，对于初学者或者想要进阶的人来说，总感觉缺少一点什么。我希望能够写一本书帮助更多的同学快速并且正确地成长。当然，促使我写这本书的另外一个原因是我的职业生涯的转折。

我从事 Java 开发工作接近 10 年时间，然后在职业教育这个“风口”下选择以讲师的身份创业，幸运的是创业过程还算顺利，也有不少收获。从程序员转行到讲师之后，我接触了几十万名有不同背景和不同经验的程序员，慢慢总结了一些课程设计的方法和授课的方法，最大的收获是要针对不同层次的学员合理设计内容的铺垫方式和技术的讲解方式，从而达到事半功倍的效果。因此，我想基于这套方法论，结合目前比较主流的微服务架构，写这样一本书去帮助大家。

诚然，写书和讲课或者写代码是完全不一样的。对于这本书中的每一句话、每一个字，我都仔细斟酌，抱着一种程序员对于代码的极致追求的心态来写这本书。实际上我们都知道程序是不可能完美的，希望读者能够抱着容忍程序中出现小 bug 的心态来原谅我文字表达的不完美。

我从 2012 年开始接触 Dubbo，当时公司正面临大规模服务化之后的服务治理和服务监控等一系列问题，Dubbo 的出现解决了很多快速发展起来的互联网公司的技术难题。从那以后，我开始深入研究 Dubbo 的底层原理。

2015 年，公司内部全面推动微服务架构，当时采用 Spring Boot+Dubbo 来构建微服务，基于微服务的治理生态和运维生态的逐步完善，我逐步认识到微服务带来的价值。在 Spring Cloud 还没有在国内大规模运用时，我们要解决服务注册与发现、服务熔断和降级、服务监控、服务路由等问题，采用的技术大部分是大公司或者 Apache 组织开源的。各种技术的拼凑和组装给企业带来了比较大的挑战。

从 2016 年开始，Spring Cloud 逐步在国内普及，它提供了微服务开发、服务治理、监控等一体化解决方案。有了这个生态，中小型互联网公司要构建微服务，不需要再去找各种开源组

件来集成。通过 Spring Cloud 可以轻松快速地构建微服务。

Spring Cloud 服务治理的整个体系采用的是 Netflix 公司提供的开源组件，但是 Netflix 公司开源组件的活跃度不是很高，并且 Netflix 公司官方也声明了对于很多组件逐步进入维护模式。所以 Spring Cloud 官方发布了声明，Spring Cloud Netflix 下的相关组件也会逐步进入维护模式（进入维护模式的意思就是从目前一直到以后一段时间 Spring Cloud Netflix 提供的服务和功能就这么多，不再开发新的组件和功能了）。同时 Spring Cloud 官方也在积极孵化其他替代品，以满足 Spring Cloud 版本迭代的需求。

值得高兴的是，2018 年 10 月 31 日的凌晨，Spring Cloud Alibaba 正式入驻 Spring Cloud 官方孵化器，并在 Maven 中央库发布了第一个版本；2019 年 8 月 1 日，在 Alibaba 仓库发布了第一个毕业版本。

Spring Cloud Alibaba 生态下的各个组件其实在国内很多公司很早之前就有使用，它们在服务治理方面比 Spring Cloud Netflix 更加强大，而且比较符合国内互联网公司的业务特性，未来可期！

本书会从技术背景到基本使用再到深层次的设计思想和原理，对 Spring Cloud Alibaba 进行全面的分析，帮助大家全面构建 Spring Cloud Alibaba 技术体系。

本书特色

- 系统化的深度整理

Spring Cloud Alibaba 在官网上有很多非常详细的资料，但是这些资料太散，没有系统性和深度。很多开发者需要花费大量的时间和精力去研究，很低效。本书系统分析了各个组件的使用方法和原理，对于初学者或者有一定经验的开发者来说，都能提供非常大的帮助。

- 高效的技术学习模型

以往我们学习一个技术往往直接从应用着手，这种方式会使得读者知其然而不知其所以然。本书对于 Spring Cloud Alibaba 生态下的技术组件采用高效的技术学习模型“场景→需求→解决方案→应用→原理”来展开，可以让读者更加了解技术背景，更好地理解技术的本质。

- 不仅有应用，还有核心原理

一个“不安分”的程序员，必然不会仅停留在技术的使用层次，技术实现原理的探索会给他带来更多的惊喜。本书在介绍 Spring Cloud Alibaba 生态下各个技术组件的使用方法的同时，

会剖析其实现原理，并且结合一些核心源码分析设计方案的落地过程。这无疑给那些充满好奇心的读者带来了“福利”。

- 大量图形化设计，化繁为简

对于复杂的技术原理，如果一味地用书面化的文字来表述，难免会增加理解成本，读者可能要花大半天的时间去理解一句话。“书”对于开发者来说，是一个很好的学习渠道，理应化繁为简，所以本书采用“动不动就画图”的方式，在书中穿插了大量的图形，帮助读者更好地掌握技术原理。

示例源码

本书相关示例源码可以在 GitHub 站点下载：<https://github.com/2227324689/Spring-Cloud-Alibaba->。

致谢

这本书笔者筹划了很久，中间因为各种原因，一直拖到 2019 年的 11 月份才正式动笔，接着到一月份又赶上了疫情在家里隔离，索性就利用隔离的这段时间没日没夜地把这本书写完了，也算是圆了自己写一本书的小梦想。

本书能够顺利出版，首先得感谢合伙人谭勇德、蒋孟枝的不断督促，以及本书的责任编辑董英的鼓励。其次，我要感谢我的妻子，是你的支持使我在写书期间可以“两耳不闻窗外事”，专注写作，从而顺利交稿。

我还要特别感谢我的好友汤陈，为本书提供了第 7 章的初稿。在整个写作过程中我们对于技术问题的交流，为我的写作提供了很多很好的思路。

读者服务

微信扫码回复：38824



- 获取博文视点学院 20 元付费内容抵扣券
- 获取免费增值资源
- 加入读者交流群，与本书作者互动
- 获取精选书单推荐

目 录

第 1 章 微服务的发展史	1
1.1 从单体架构到分布式架构的演进	2
1.1.1 单体架构	2
1.1.2 集群及垂直化	3
1.1.3 SOA	4
1.1.4 微服务架构	5
1.2 微服务架构带来的挑战	7
1.2.1 微服务架构的优点	7
1.2.2 微服务架构面临的挑战	7
1.3 如何实现微服务架构	8
1.3.1 微服务架构图	8
1.3.2 微服务架构下的技术挑战	9
第 2 章 微服务解决方案之 Spring Cloud	10
2.1 什么是 Spring Cloud	11
2.2 Spring Cloud 版本简介	12
2.3 Spring Cloud 规范下的实现	13
2.4 Spring Cloud Netflix	14
2.5 Spring Cloud Alibaba	15

2.5.1	Spring Cloud Alibaba 的优势	15
2.5.2	Spring Cloud Alibaba 的版本	16
第 3 章	Spring Cloud 的核心之 Spring Boot	18
3.1	重新认识 Spring Boot	18
3.1.1	Spring IoC/DI	19
3.1.2	Bean 装配方式的升级	21
3.1.3	Spring Boot 的价值	23
3.2	快速构建 Spring Boot 应用	25
3.3	Spring Boot 自动装配的原理	26
3.3.1	自动装配的实现	27
3.3.2	EnableAutoConfiguration	28
3.3.3	AutoConfigurationImportSelector	29
3.3.4	自动装配原理分析	31
3.3.5	@Conditional 条件装配	34
3.3.6	spring-autoconfigure-metadata	37
3.4	手写实现一个 Starter	38
3.4.1	Starter 的命名规范	38
3.4.2	实现基于 Redis 的 Starter	38
3.5	本章小结	41
第 4 章	微服务架构下的服务治理	42
4.1	如何理解 Apache Dubbo	43
4.2	Apache Dubbo 实现远程通信	44
4.3	Spring Boot 集成 Apache Dubbo	48
4.4	快速上手 ZooKeeper	52
4.4.1	ZooKeeper 的安装	52
4.4.2	ZooKeeper 的数据结构	52
4.4.3	ZooKeeper 的特性	53
4.4.4	Watcher 机制	54
4.4.5	常见应用场景分析	54

4.5	Apache Dubbo 集成 ZooKeeper 实现服务注册.....	56
4.5.1	Apache Dubbo 集成 ZooKeeper 实现服务注册的步骤.....	57
4.5.2	ZooKeeper 注册中心的实现原理.....	59
4.6	实战 Dubbo Spring Cloud.....	60
4.6.1	实现 Dubbo 服务提供方.....	60
4.6.2	实现 Dubbo 服务调用方.....	63
4.7	Apache Dubbo 的高级应用.....	65
4.7.1	集群容错.....	66
4.7.2	负载均衡.....	67
4.7.3	服务降级.....	68
4.7.4	主机绑定规则.....	69
4.8	Apache Dubbo 核心源码分析.....	70
4.8.1	源码构建.....	71
4.8.2	Dubbo 的核心之 SPI.....	71
4.8.3	无处不在的自适应扩展点.....	81
4.8.4	Dubbo 中的 IoC 和 AOP.....	85
4.8.5	Dubbo 和 Spring 完美集成的原理.....	89
4.9	本章小结.....	95
第 5 章	服务注册与发现.....	97
5.1	什么是 Alibaba Nacos.....	98
5.2	Nacos 的基本使用.....	99
5.2.1	Nacos 的安装.....	99
5.2.2	Nacos 服务注册发现相关 API 说明.....	100
5.2.3	Nacos 集成 Spring Boot 实现服务注册与发现.....	101
5.3	Nacos 的高可用部署.....	103
5.3.1	安装环境要求.....	104
5.3.2	安装包及环境准备.....	104
5.3.3	集群配置.....	104
5.3.4	配置 MySQL 数据库.....	105
5.3.5	启动 Nacos 服务.....	105

5.4	Dubbo 使用 Nacos 实现注册中心	106
5.5	Spring Cloud Alibaba Nacos Discovery	109
5.5.1	服务端开发	109
5.5.2	消费端开发	113
5.6	Nacos 实现原理分析	115
5.6.1	Nacos 架构图	116
5.6.2	注册中心的原理	116
5.7	深入解读 Nacos 源码	117
5.7.1	Spring Cloud 什么时候完成服务注册	117
5.7.2	NacosServiceRegistry 的实现	122
5.7.3	从源码层面分析 Nacos 服务注册的原理	123
5.7.4	揭秘服务提供者地址查询	127
5.7.5	分析 Nacos 服务地址动态感知原理	130
5.8	本章小结	132
第 6 章	Nacos 实现统一配置管理	133
6.1	Nacos 配置中心简介	134
6.2	Nacos 集成 Spring Boot 实现统一配置管理	134
6.2.1	项目准备	134
6.2.2	启动 Nacos Server	135
6.2.3	创建配置	135
6.2.4	启动服务并测试	137
6.3	Spring Cloud Alibaba Nacos Config	137
6.3.1	Nacos Config 的基本应用	137
6.3.2	动态更新配置	139
6.3.3	基于 Data ID 配置 YAML 的文件扩展名	140
6.3.4	不同环境的配置切换	141
6.3.5	Nacos Config 自定义 Namespace 和 Group	142
6.4	Nacos Config 实现原理解析	145
6.4.1	配置的 CRUD	145
6.4.2	动态监听之 Pull Or Push	146

6.5	Spring Cloud 如何实现配置的加载	148
6.5.1	PropertySourceBootstrapConfiguration	148
6.5.2	PropertySourceLocator	150
6.6	Nacos Config 核心源码解析	152
6.6.1	NacosFactory.createConfigService	154
6.6.2	NacosConfigService 构造	155
6.6.3	ClientWorker	155
6.6.4	ClientWorker.checkConfigInfo	157
6.6.5	LongPollingRunnable.run	157
6.6.6	服务端长轮询处理机制	161
6.6.7	ClientLongPolling	164
6.7	本章小结	167
第 7 章	基于 Sentinel 的微服务限流及熔断	168
7.1	服务限流的作用及实现	168
7.1.1	计数器算法	169
7.1.2	滑动窗口算法	170
7.1.3	令牌桶限流算法	171
7.1.4	漏桶限流算法	172
7.2	服务熔断与降级	172
7.3	分布式限流框架 Sentinel	174
7.3.1	Sentinel 的特性	174
7.3.2	Sentinel 的组成	175
7.3.3	Sentinel Dashboard 的部署	175
7.4	Sentinel 的基本应用	176
7.4.1	Sentinel 实现限流	176
7.4.2	资源的定义方式	178
7.4.3	Sentinel 资源保护规则	179
7.4.4	Sentinel 实现服务熔断	183
7.5	Spring Cloud 集成 Sentinel 实践	185
7.5.1	Sentinel 接入 Spring Cloud	185

7.5.2	基于 Sentinel Dashboard 来实现流控配置	187
7.5.3	自定义 URL 限流异常	189
7.5.4	URL 资源清洗	189
7.6	Sentinel 集成 Nacos 实现动态流控规则	190
7.7	Sentinel Dashboard 集成 Nacos 实现规则同步	194
7.7.1	Sentinel Dashboard 源码修改	194
7.7.2	Sentinel Dashboard 规则数据同步	199
7.8	Dubbo 集成 Sentinel 实现限流	200
7.8.1	Dubbo 服务接入 Sentinel Dashboard	200
7.8.2	Dubbo 服务限流规则配置	201
7.9	Sentinel 热点限流	203
7.9.1	热点参数限流的使用	204
7.9.2	@SentinelResource 热点参数限流	205
7.9.3	热点参数规则说明	206
7.10	Sentinel 的工作原理	206
7.11	Spring Cloud Sentinel 工作原理分析	207
7.12	Sentinel 核心源码分析	210
7.12.1	限流的源码实现	210
7.12.2	实时指标数据统计	216
7.12.3	服务降级的实现原理	225
7.13	本章小结	229
第 8 章	分布式事务	230
8.1	分布式事务问题的理论模型	231
8.1.1	X/Open 分布式事务模型	231
8.1.2	两阶段提交协议	233
8.1.3	三阶段提交协议	234
8.1.4	CAP 定理和 BASE 理论	236
8.2	分布式事务问题的常见解决方案	238
8.2.1	TCC 补偿型方案	238
8.2.2	基于可靠性消息的最终一致性方案	239