

# 高可用可伸缩微服务架构



[高可用可伸缩微服务架构\\_下载链接1](#)

著者:程超

出版者:电子工业出版社

出版时间:2019-5-1

装帧:平装

isbn:9787121362132

近年来微服务架构已经成为大规模分布式架构的主流技术，越来越多的公司已经或开始转型为微服务架构。本书不以某一种微服务框架的使用为主题，而是对整个微服务生态进行系统性的讲解，并结合工作中的大量实战案例为读者呈现一本读完即可实际上手应用的工具书。

书中的理论部分介绍了微服务架构的发展历程，通俗地讲解了领域驱动设计，帮助读者更好地利用DDD来建模和划分服务；微服务稳定性保证的常用手段和微服务下如何保证事务的一致性这两章凝聚了作者多年的积累和思考，相信读者看完后会有不一样的感触和收获；书中实战部分的内容非常丰富，以项目为基础，逐层介绍常见的Dubbo、Spring Cloud和Service Mesh框架的具体使用方法，并对实现原理进行剖析；书中还以具体案例全面介绍了微服务双活体系建设、微服务监控与告警、微服务编排、百亿流量微服务网关的设计与实现，以及基于支付场景下的微服务改造等，并让读者了解如何借助微服务来增强和重构现有的遗留系统。

不管是刚接触微服务的新手，还是正在尝试借助微服务解放生产力的开发人员或运维人员，甚至是立志于构建高可用可伸缩的微服务体系的技术Leader和架构师，阅读本书，对读者必有裨益。

## 作者介绍:

程超，网名小程故事多，现任某公司高级架构师，超过12年的Java研发经验，8年技术管理和架构经验，熟悉支付和电商领域，擅长微服务生态建设和运维监控，对Dubbo、Spring Cloud和gRPC等微服务框架有深入研究，帮助多家公司进行过微服务建设和改造。合著作品《深入分布式缓存》，阿里云MVP、云栖社区外部专家、Codingfly社区特聘技术专家、CSDN博主专家。

梁桂钊，现任某互联网公司高级开发工程师，参与过内容分发、K12教育、淘系电商等项目。目前，专注于新零售电商服务的业务摸索和电商服务创新实践。具有Java核心技术、微服务、分布式、高并发等领域一线实战经验，并对新兴技术方向和各种开源框架有浓厚兴趣。公众号「服务端思维」的作者。

秦金卫（KimmKing），现任某公司高级技术总监/Apache Dubbo PPMC，阿里前架构师/某商业银行北京研发中心负责人。关注互联网、电商、金融、支付、区块链等领域，10多年研发管理和架构经验，对于中间件、SOA、微服务，以及各种开源技术非常热衷，活跃于Dubbo、Fastjson、Mule、ActiveMQ等多个开源社区。个人博客<http://kimmking.github.io>。

方志斌，现任某物联网公司高级研发工程师。目前专注于大型物联网平台架构的设计与开发工作。对于微服务、分布式、集群有一定的研究和实战经验。对Java领域的开源框架有浓厚的兴趣，喜欢深入分析、总结框架源码。SpringForAll社区核心成员，组织多次社区技术专题、问答等活动。

张逸，架构编码实践者，微服务架构设计者，领域驱动设计布道师，大数据平台架构师。著译作包括《软件设计精要与模式》《恰如其分的软件架构》《人件》等。个人微信公众号为「逸言」，个人博客：<http://zhangyi.xyz>。

杜琪，网名阿杜，现任蚂蚁金服高级研发工程师，2015年6月毕业于南开大学，计算机系统结构硕士。毕业后开始接触分布式业务系统开发，曾在有赞负责用户中心基础服务，对分布式业务系统的稳定性、可靠性有丰富的经验。喜欢研究底层技术，喜欢研究疑难技术问题，例如JVM内存问题排查、GC调优，等等。有对外输出分享的习惯，是公众号javaadu的维护者。

殷琦，网名涤生，现任“美团点评”技术专家，2015年3月毕业于东华大学，软件工程硕士。2015年3月加入“美团点评”基础架构部，开始接触微服务架构，之后一直从事服务框架的研发工作，对微服务架构发展与演进有非常深刻的认识。个人比较喜欢研究并分享新技术，时刻关注并实践微服务架构最前沿的技术，如Service

Mesh、Serverless 等。

肖冠宇，曾就职于小米、人民网等互联网公司，具有丰富的大数据一线实战经验，专注大数据处理技术及机器学习算法研究。著有《企业大数据处理：Spark、Druid、Flume与Kafka应用实践》《Python3快速入门与实战》等书籍。

目录:	1.1 什么是架构	1
	1.2 几个相关概念	3
	1.3 从软件的生命周期看架构设计	7
	1.4 架构的形式与特点	9
	1.5 架构的目标与方法	13
	1.6 架构的不同风格	18
第2章	微服务领域驱动设计	30
2.1	领域驱动设计	31
2.1.1	领域驱动设计概览	31
2.1.2	问题域与解决方案域	34
2.1.3	限界上下文	35
2.1.4	上下文映射	40
2.1.5	领域架构	43
2.2	微服务的设计	48
2.2.1	限界上下文的边界	48
2.2.2	限界上下文即微服务	51
2.2.3	识别限界上下文	53
2.2.4	微服务之间的协作	59
2.3	小结	63
第3章	Apache Dubbo框架的原理与实现	64
3.1	Dubbo框架的选型与使用	64
3.1.1	Dubbo框架的选型	64
3.1.2	Dubbo框架的使用	65
3.2	Dubbo框架的原理分析	70
3.2.1	总体架构分析	70
3.2.2	Dubbo Bean的加载	73
3.2.3	Dubbo Extension机制	77
3.2.4	Dubbo消费端	91
3.2.5	Dubbo服务端	126
3.2.6	Dubbo的通信机制	144
3.3	基于Dubbo的自动化Mock系统	165
3.3.1	Mock模拟系统的产生	166
3.3.2	Dubbo Mock的使用	167
3.3.3	Dubbo Mock的原理解析	168
3.3.4	自动化Mock系统的实现	170
第4章	Spring Boot/Spring Cloud实践	177
4.1	Spring Boot原理剖析	178
4.1.1	Spring Boot Quick Start	178
4.1.2	Spring Boot之SpringApplication	180
4.1.3	spring-boot-loaded模块分析	186
4.1.4	spring-boot-autoconfigure模块分析	190
4.1.5	Spring Boot Conditional注解分析	195
4.2	Dubbo Spring Boot Starter	198
4.2.1	Dubbo Spring Boot Starter简介	198
4.2.2	Dubbo Initializr及sample	198
4.2.3	dubbo-spring-boot-autoconfigure模块	201
4.2.4	dubbo-spring-boot-actuator模块	204

- 4.3 Spring Cloud栈 211
  - 4.3.1 为什么微服务架构需要Spring Cloud 212
  - 4.3.2 Spring Cloud技术栈总览 214
  - 4.3.3 spring-cloud-scaffold基础库集合 215
- 4.4 基于Maven Archetype的脚手架 239
  - 4.4.1 Maven Archetype 239
  - 4.4.2 脚手架的搭建 240
  - 4.4.3 生成脚手架 247
- 第5章 微服务稳定性保证的常用手段 250
  - 5.1 微服务的稳定性 250
  - 5.2 高可用 250
    - 5.2.1 限流原理与实现 250
    - 5.2.2 断路器原理与实现 256
    - 5.2.3 超时与重试 260
  - 5.3 高并发 262
    - 5.3.1 异步 263
    - 5.3.2 缓存 270
  - 5.4 总结 272
- 第6章 微服务下如何保证事务的一致性 273
  - 6.1 从本地事务到分布式事务的演变 273
  - 6.2 强一致性解决方案 276
    - 6.2.1 二阶段提交协议 276
    - 6.2.2 三阶段提交协议 278
  - 6.3 最终一致性解决方案 280
    - 6.3.1 TCC模式 280
    - 6.3.2 补偿模式 285
    - 6.3.3 可靠事件模式 286
  - 6.4 开源项目的分布式事务实现解读 288
    - 6.4.1 Apache RocketMQ 288
    - 6.4.2 ServiceComb 289
- 第7章 百亿流量微服务网关的设计与实现 293
  - 7.1 API网关概述 293
    - 7.1.1 分布式服务架构、微服务架构与 API 网关 293
    - 7.1.2 API网关的定义、职能与关注点 296
    - 7.1.3 API网关的分类与技术分析 298
  - 7.2 开源网关的分析与调研 300
    - 7.2.1 常见的开源网关介绍 300
    - 7.2.2 四大开源网关的对比分析 308
    - 7.2.3 开源网关的技术总结 309
  - 7.3 百亿流量交易系统API网关设计 310
    - 7.3.1 百亿流量交易系统API网关的现状和面临的问题 310
    - 7.3.2 业务网关的设计与最佳实践 313
- 第8章 微服务编排 317
  - 8.1 Netflix Conductor 317
  - 8.2 Netflix Conductor的架构 319
  - 8.3 Conductor的使用案例 320
  - 8.4 Netflix Conductor源码分析 328
    - 8.4.1 Client层源码分析 330
    - 8.4.2 Server端源码分析 337
    - 8.4.3 core端源码分析 344
- 第9章 微服务数据抽取与统计 356
  - 9.1 案例小故事 356
  - 9.2 数据仓库概述 358
    - 9.2.1 什么是数据仓库 358

- 9.2.2 数据仓库架构 360
- 9.2.3 数据仓库建模方法 361
- 9.2.4 数据抽取、转换和加载 361
- 9.2.5 数据统计 362
- 9.3 数据仓库工具Hive 362
  - 9.3.1 安装Hive 364
  - 9.3.2 Hive数据库 366
  - 9.3.3 Hive表 367
- 9.4 使用Sqoop抽取数据 369
  - 9.4.1 安装Sqoop 370
  - 9.4.2 将MySQL表数据导入Hive表 372
- 9.5 ETL作业调度 375
- 第10章 微服务双活体系建设 377
  - 10.1 系统高可用 377
  - 10.2 双活数据中心 379
    - 10.2.1 单机房部署 379
    - 10.2.2 双机房部署 380
    - 10.2.3 基于支付场景的双活体系建设 383
- 第11章 基于支付场景下的微服务改造与性能优化 399
  - 11.1 支付场景的介绍 399
  - 11.2 支付业务建模和服务划分 400
  - 11.3 支付场景下微服务架构的详解与分析 402
    - 11.3.1 业务架构分析 402
    - 11.3.2 技术平台详解 404
  - 11.4 从代码层面提升微服务架构的性能 407
    - 11.4.1 从代码和设计的角度看 407
    - 11.4.2 从整体架构的角度看 418
  - 11.5 微服务架构中常见的一些故障分析技巧 421
- 第12章 遗留系统的微服务架构改造 426
  - 12.1 代码分层结构的转变 426
  - 12.2 遗留系统的债券与思考 430
  - 12.3 从单体系统拆分服务的方法论 431
  - 12.4 遗留系统的微服务架构改造 436
    - 12.4.1 从代码重构开始 436
    - 12.4.2 拆分服务需要面向服务进行架构设计 437
    - 12.4.3 改造是一个渐进的过程 437
    - 12.4.4 单元测试是基石 440
    - 12.4.5 面向失败的设计 440
    - 12.4.6 前后端分离 441
    - 12.4.7 共享现有数据库 441
    - 12.4.8 灰度发布的必要性 442
    - 12.4.9 日志聚合与全链路监控 442
- 第13章 Service Mesh详解 443
  - 13.1 Service Mesh是什么 443
  - 13.2 Service Mesh的背景 443
  - 13.3 Service Mesh介绍 444
    - 13.3.1 Service Mesh架构 444
    - 13.3.2 Service Mesh能做什么 445
  - 13.4 Service Mesh的价值 446
  - 13.5 Service Mesh现状 447
  - 13.6 Service Mesh存在的问题 448
  - 13.7 Istio详解 449
    - 13.7.1 Istio架构 449
    - 13.7.2 数据平面 450

13.7.3 控制平面 455  
13.7.4 Isito案例 458  
13.8 Service Mesh展望 461  
第14章 微服务监控实战 463  
14.1 APM原理与应用 464  
14.1.1 什么是APM 464  
14.1.2 APM监控点 464  
14.1.3 APM深入解析 464  
14.2 Prometheus监控系统介绍 492  
14.2.1 Prometheus的主要特点 493  
14.2.2 Prometheus的架构及组件介绍 493  
14.2.3 Prometheus的安装 494  
14.2.4 使用Prometheus对MySQL进行监控 498  
14.2.5 Prometheus的告警机制 501  
• • • • • [\(收起\)](#)

[高可用可伸缩微服务架构\\_下载链接1](#)

## 标签

微服务

架构

ServiceMesh

Java

DDD

计算机

java

Web开发

## 评论

国内超过两个作者的都是烂书

-----  
本书是截图、代码的堆砌，毫无价值可言

-----  
本书很像那种很多明星的拼盘演唱会，当然这种拼盘演唱会的目的大多就是走穴捞金。

-----  
从J2EE的史前时代走过来的开发者，应该对当初的ESB、Spring、SOA等名称有很深的印象，随着技术的发展，微服务作为新兴的技术开始在互联网技术领域大展身手，本书讨论了微服务的起源、核心框架和很多关键问题的理论和最佳实践，最后还对Service Mesh这种近年来新兴的技术进行了分析和阐述。阅读这本书，你可以学习到一线开发者的经验和总结，一定能够有所收获。

-----  
估计是作者太多的原因，内容太杂，各章质量参差不齐、而且互相独立，没有一条把内容串起来的主线。

-----  
只想看第十二章，结果很失望

-----  
国人著，好几个作者，没有一个统一的主线，各个章节相互独立，没有连贯性，概念超多，代码片段超多

-----  
#沐雨斋悦读#对一本书评价时，要看对其定位，这本作为入门书三星还是可以有的，方方面面都讲了点，也算全面。但你较真要比入门高点，那么截图代码凑页数、每章体例不一各自为政，这种感觉是KPI产物的书就确实不堪了。

-----  
认识其中两个作者

-----  
1.微服务是当今互联网技术的主流技术。2.dubbo 和spring cloud  
是现在最流行的微服务平台。3.本书是国内程序员的必读书目。4.读书过程中受益匪浅  
，学到了很多，谢谢本书作者的知识分享。——朝天王

-----  
可能我水平不够吧 我觉这本书真是非常之差 搞得我专门来评论  
就是代码和死板概念的堆砌 而且代码和概念在书中的编排凌乱 非常失望  
希望作者如果下次出书要用心点

-----  
这本书只是大量代码的堆砌，好多东西一笔带过，不值得一看。所以，基本不值得购买  
。国内技术书籍透露着浮躁的气息。

-----  
评价和书评日期大都在同一天 根本刷的 真没意思

-----  
本书从微服务和领域驱动开发的角度阐述高可用和可伸缩架构，知识点覆盖全面。书籍  
作者由多名一线互联网资深人员联合出品，体现了现代技术书籍的合作共赢的模式。各  
位作者取长补短，将最好的内容呈现至读者面前。在架构类的书层出不穷的当今，本书  
特点鲜明，是我眼中的优秀书籍，推荐读者品读。

-----  
书籍从理论到实践全面的讲解了高可用可伸缩微服务架构，由多位经验丰富的技术专家  
联合撰写，推荐对微服务感兴趣的同学入手一本！

-----  
在下通过这两天如饥似渴的研读，感觉对微服务架构的理解与使用又精进了许多，收获  
颇丰。希望各位作者以后还能继续用文字的形式，带给我们更多的知识，为我们在微服  
务技术方面提供更多宝贵的经验。最后，这真的是一本好书。

-----  
这本书写得很好，都是有丰富经验的大佬写的，梁大也很负责回答每个问题，微服务方  
面这本书写得很好

-----  
本书系统的介绍了微服务的框架，读完收获很大，感谢各位大佬



-----  
程超为首出品的微服务书籍讲解精彩纷呈，几位作者的篇章针对微服务高可用架构实战内容很详细，在市面上很难找到如此全面的微服务书籍，推荐购买。

-----  
不知道在讲啥，根本没有主线，还不如看看博客

-----  
[高可用可伸缩微服务架构\\_下载链接1](#)

## 书评

微服务的起源是什么时候？微服务这个概念最早是在2011年5月威尼斯的一个软件架构会议上讨论并提出的，微服务架构则是由Fred George在2012年的一次技术大会上所提出，在大会的演讲中他讲解了如何分拆服务以及如何利用MQ来进行服务间的解耦，这就是最早的微服务架构雏形。本书较...

-----  
微服务架构是当前最流行的架构模式，微服务的范围非常广，涉及的知识点也非常多。目前微服务相关的书籍比较多，可能非常难以选择。这里隆重推荐本书，本书非常全面且深入的介绍了微服务框架相关的知识，另外还对微服务的新趋势 ServiceMesh做了相关讲解。书的作者都非常的资深...

-----  
这本书很特别，由多位经验丰富的技术大佬共同撰写，将其所在微服务的经验积累与思考分享了出来。书籍透彻分享了微服务的理念及当前热门的三个框架的实践，是一本不可多得的好书。书籍可以结合专门的Dubbo、Spring Cloud、Service Mesh

的书一起来看，不局限在某个技术框架之中...

-----  
书中的理论部分介绍了微服务架构的发展历程，通俗地讲解了领域驱动设计，帮助读者更好地利用DDD来建模和划分服务；微服务稳定性保证的常用手段和微服务下如何保证事务的一致性这两章凝聚了作者多年的积累和思考  
作者很用心很用心很用心，看的收货很大，dubbo,spring cloud,serv...

-----  
[高可用可伸缩微服务架构\\_下载链接1](#)