

架构宝典

中生代技术社区

著



架构宝典

中生代技术社区 著



电子工业出版社

Publishing House of Electronics Industry

北京•BEIJING

内 容 简 介

本书从架构设计概念、方法、模式、实践案例等4个方面进行阐述,内容丰富,是一本可以很好指导实践的书籍,更是各位资深专家和架构师的经验总结。

本书共分为4部分:第一部分介绍架构设计的概念、方法论,以及架构师的核心能力;第二部分介绍微服务架构选型、架构模式与实践、数据一致性保证、微服务与DevOps的关系以及如何设计云微服务架构;第三部分介绍移动电商、消费信贷、支付系统、金融撮合等领域的优秀实践;第四部分介绍优化系统架构性能的方法论、案例、关键技术等。

本书适合即将步入架构领域的技术专家,以及致力于提升软件架构设计能力的架构师、设计人员和开发人员阅读,希望本书能够为读者的实际工作和个人成长提供帮助。

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

架构宝典 / 中生代技术社区著. —北京: 电子工业出版社, 2019.4

ISBN 978-7-121-36066-4

I. ①架… II. ①中… III. ①软件设计 IV. ①TP311.1

中国版本图书馆CIP数据核字(2019)第036380号

责任编辑: 刘恩惠

印 刷: 北京天宇星印刷厂

装 订: 北京天宇星印刷厂

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: 17.5 字数: 381千字

版 次: 2019年4月第1版

印 次: 2019年4月第1次印刷

定 价: 75.00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 zlts@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。

序

2015年年底，有感于成都技术氛围亟待提升，来自蚂蚁金服的于君泽（右军）和我组织成立了中生代技术社区，旨在为技术人员提供一个更便于交流的平台。中生代社区的成立，一方面可以壮大成都技术交流圈，另一方面可以汇集一线城市的技术力量，通过技术与分享助力技术水平的提升。3年时间里，社区连接了5000多位技术大咖，组织了数百场线上技术分享，以及数十场线下技术沙龙。

2017年，出版社的编辑老师找到我们，希望通过出版图书的方式将技术架构方面的知识传播给更多的读者。于是社区邀请了来自一线互联网公司的技术架构专家和架构领域布道者进行经验分享，并把这些技术大咖的分享内容进行了系统的整理，经过重新编排、精心打磨、多次迭代、推陈出新，中生代技术社区的《架构宝典》终于在两年后的今天与大家见面了。

本书凝聚了社区技术架构专家们的心血，涵盖架构设计、技术选型、实际案例、技术原理等方方面面的知识，对于即将步入架构领域的技术专家和需要继续提升专业素养的架构师而言都是不可或缺的案头书。

本书的出版离不开众多朋友的大力支持，感谢本书策划人右军、孔庆龙、李伟山，他们共同确立了本书的内容架构，同时还负责组织作者，沟通技术细节，为出版提供了有力保障。同时感谢石头（石涛生）和老曹（曹洪伟），他们在项目前期的策划和统稿阶段付出了许多心血。

感谢本书的作者，他们是李晓时、右军、杨波、张逸、孔庆龙、刘地生、田向阳、刘凡、王东、朱攀、朱永光、黄哲铿、王辉（觉远）、陈宗（铁手）、李伟山、陈显铭（山丘）、高磊，是他们近两年来夜以继日的努力，才让本书日臻完善。

感谢编辑刘恩惠和其他编辑老师，是他们的专业让本书的出版成为可能，他们不厌其烦地审阅、校正，保障了本书内容的专业性。

最后要特别感谢一直以来大力支持中生代技术社区的10多万名朋友，没有你们的共同参与，就没有社区的今天，更没有本书的诞生，感谢你们！

王友强
中生代技术社区创始人

前言

软件架构定义的多样性和各个企业对架构师的定位不同，导致很多程序员在向架构师转型的过程中有些茫然。本书汇聚了国内一线互联网公司的技术架构专家、架构领域布道者在架构方法论、微服务、业务架构、性能架构几个方面的经验，从系统架构设计和实践方法与技巧两个方面进行阐述，非常具有指导意义，希望读者能在这里找到解惑的钥匙或突破瓶颈的方法。

这本书在形式上与《论语》类似，是汇集各个领域专家经验的文集，读者可以根据自己的关注点有针对性地进行阅读，既可以纵向了解各个行业 IT 系统建设的演进过程，也可以横向对比分析某类问题的解决思路和方法。

本书共分为 4 部分：

第一部分将介绍架构设计的概念、方法论、架构治理，以及架构师需要具备的核心能力等基础内容，能让读者对架构设计的整个过程有一个基础的认知。

第二部分将介绍微服务的各个细节，包括架构设计的技术选型、架构模式与实践、数据一致性保证、微服务与 DevOps 的关系，以及如何设计云微服务架构。通过这一部分的学习，读者将了解微服务建设涉及的方方面面内容。

第三部分将介绍移动电商、消费信贷、支付系统、金融撮合几个业务领域的优秀实践案例，以及系统架构设计和迭代演进的过程，从支撑业务发展的视角阐述架构设计和业务的关系。

第四部分将介绍优化系统架构性能的方法论、案例、关键技术等，希望能使读者了解性能优化的常见招式，并通过这些招式突破系统性能的瓶颈，明确当前应用所处的发展阶段及下一步演进的目标。

回想自己从程序员成长为架构师的历程，期间有过迷茫，也走过弯路，很庆幸在成长的各个阶段都遇到了一些挚友指点迷津，使我能在致力于成为一名合格架构师的道路上坚持探索。

我认为，个人能力的高低取决于很多方面的因素，它们之间满足如下关系：

能力 = f (认知, 洞察, 实践, 经验总结)

在按照上述公式中的几个关键因素审阅本书初稿并与各位作者进行交流的过程中，我受益匪浅。如果本书的相关内容能为读者进一步提升个人能力带来一点思考和裨益，那么作者们就心满意足了。

本书非常适合各位 IT 从业者阅读、参考。能参与本书的出版，我深感荣幸，各位作者“开放、协作、分享”的互联网精神令我非常敬佩。感谢各位作者和本书编辑刘恩惠，他们认真负责、不辞劳苦地对本书进行了多次修订和审校，提出了很多有价值的意见，保证了本书的质量。感谢几位参与前期内容创作的作者，他们提供的思路和建议，为本书最终出版做出了巨大的贡献。

孔庆龙

2019 年 1 月

读者服务

轻松注册成为博文视点社区用户 (www.broadview.com.cn), 扫码直达本书页面。

- **提交勘误：**您对书中内容的修改意见可在 [提交勘误](#) 处提交，若被采纳，将获赠博文视点社区积分（在您购买电子书时，积分可用来抵扣相应金额）。
- **交流互动：**在页面下方 [读者评论](#) 处留下您的疑问或观点，与我们和其他读者一同学习交流。

页面入口：<http://www.broadview.com.cn/36066>



目录

第 1 篇 架构方法论

1. 有关架构的概念认知	2
1.1 引言	2
1.2 认识架构	2
1.3 规划、架构与设计	4
1.3.1 三个不同层次的概念	4
1.3.2 规划与架构	4
1.3.3 架构与设计	5
1.4 解耦与服务化	6
1.5 分工与组件化	6
1.6 集中与分布	7
1.7 规划还是演化	8
1.8 简化也很必要	9
1.9 架构与软件开发（软件过程）	11
1.9.1 敏捷软件开发	11
1.9.2 敏捷软件开发与微服务架构	12
1.10 探讨传统企业 IT 系统架构	12
1.10.1 从架构角度解决企业应用痛点	13
1.10.2 企业应用系统架构服务化迁移步骤	13
1.10.3 企业应用系统架构服务化参考模型	14
1.11 总结	15
2 以终为始的架构设计	16
2.1 缘起	16
2.2 我们的思考方式	17

2.3	为什么要做架构设计	18
2.3.1	由模型到实施	18
2.3.2	业务规模发展带来的复杂度	19
2.3.3	从沟通视角看软件架构	20
2.4	如何做架构设计	20
2.4.1	以终为始，不忘初心	21
2.4.2	PMC 框架	21
2.4.3	从多视角、多层次看架构	23
2.4.4	满足利益相关者的需要	25
2.4.5	聚焦 SLA	26
2.4.6	抽象、协作、扩展、复用	28
2.4.7	分析全息视图	29
2.5	架构是什么	29
2.5.1	架构兼具组成和决策的特点	30
2.5.2	架构是演进来的	30
2.5.3	无纯粹的非功能特性	31
2.6	总结	32
3	闭环架构方法	34
3.1	系统提升的一般性方法和反馈环	34
3.2	产品创新闭环	36
3.3	组织闭环	37
3.4	研发流程闭环	39
3.5	系统架构闭环	40
3.6	一个反馈闭环实例	41
3.7	总结	44
4	复杂与架构演进的关系	46
4.1	什么是复杂	46
4.1.1	规模	47
4.1.2	结构	51
4.1.3	变化	58

4.2	用架构思维控制复杂	67
4.2.1	分而治之，控制规模	67
4.2.2	保持架构的清晰与一致	67
4.2.3	拥抱变化	70
4.3	总结	71
5	架构师的核心能力	72
5.1	架构师承担的职责	72
5.2	架构师的核心能力	73
5.2.1	经验	73
5.2.2	沟通	74
5.2.3	快速学习	74
5.2.4	解决问题的能力	76
5.3	架构能力的修炼	76
5.4	总结	77

第2篇 面向架构的架构（微服务）

6	快速继承微服务实践	80
6.1	概述	80
6.2	我们希望的微服务是什么样子的	81
6.3	实现微服务	82
6.4	实现的考量及调研	83
6.4.1	gRPC 有什么问题	83
6.4.2	如何提高易用性	83
6.5	让微服务快速落地	87
6.6	总结	89
7	微服务架构下的事务处理	90
7.1	概述	90
7.2	使用传统的本地事务和分布式事务保证一致性	90
7.3	在微服务架构中应满足数据最终一致性原则	92
7.4	微服务架构实现最终一致性的三种模式	92

7.4.1	可靠事件模式	93
7.4.2	业务补偿模式	101
7.4.3	TCC (Try-Confirm-Cancel) 模式	106
7.5	总结	109
8	微服务架构模式与实践	110
8.1	微服务的主要架构模式	110
8.1.1	一体化模式	110
8.1.2	聚合模式	110
8.1.3	代理模式	111
8.1.4	资源共享模式	112
8.1.5	异步消息模式	112
8.2	大量微服务带来的挑战	112
8.3	微服务在好雨的解决方案	113
8.3.1	服务发现和注册	113
8.3.2	弹性伸缩与负载均衡	115
8.3.3	容错处理 (断路器与限流)	116
8.3.4	监控与报警	116
8.3.5	数据存储与共享	116
8.3.6	日志分析	117
8.4	总结	118
9	微服务与 DevOps 架构实践	119
9.1	概述	119
9.2	某金融创业公司的微服务架构选型	122
9.3	设计思想	124
9.4	总体架构	125
9.4.1	总体架构的特性	125
9.4.2	具体实现	126
9.5	对微服务的支撑	128
9.6	DevOps 平台总体架构	130
9.7	面向微服务的研发团队介绍	131

9.8 总结	133
10 基于云的微服务架构	134
10.1 概述	134
10.2 德比软件数据对接平台的架构	134
10.3 德比软件微服务架构基础设施	135
10.4 API 网关	136
10.5 服务框架	136
10.5.1 高可用 RPC	137
10.5.2 服务依赖管理	138
10.6 基础服务	140
10.6.1 配置中心	140
10.6.2 安全数据服务	141
10.6.3 数据存储服务	141
10.6.4 订单服务	142
10.7 服务降级	142
10.8 自动化	142
10.9 日志处理	143
10.10 调用链追踪	144
10.11 服务健康状态	144
10.11.1 报警	144
10.11.2 监控	145
10.12 发布管理	146
10.13 总结	147
11 Service Fabric: 平台架构解析	148
11.1 Service Fabric 概述	148
11.1.1 Service Fabric 的特点	148
11.1.2 Service Fabric 的适用场景	149
11.2 Service Fabric 编程模式	150
11.3 Service Fabric 服务通信	152
11.3.1 Service Fabric 的微服务开发工具	153

11.3.2 对 Service Fabric 的微服务进行监控和诊断	153
11.4 Service Fabric 可靠服务概述	154
11.4.1 Service Fabric 可靠服务——有状态服务的体系结构	154
11.4.2 Service Fabric 可靠服务——无状态服务的体系结构	156
11.4.3 Service Fabric 可靠服务——备份	157
11.5 Service Fabric 可靠执行组件	157
11.6 Service Fabric 容器应用程序	159
11.7 总结	160

第 3 篇 面向业务的架构

12 如何搭建高可伸缩的移动电商架构	162
12.1 概述微服务的主要架构模式	162
12.2 移动端的混合架构	162
12.3 服务器端的 SOA 架构	164
12.4 PC 端和移动端的开发协作管理	166
12.5 基于容器的虚拟化	166
12.6 关于电商的私有云建设	167
12.7 如何应用弹性云来应对电商大促	169
12.8 应对电商大促峰值的“独孤九剑”	170
13 消费信贷系统“白付美”是如何持续优化的	172
13.1 “白付美”介绍	172
13.2 “白付美”从 0 到 1	172
13.3 大促期间的性能保证	174
13.4 大促期间的稳定性保证	177
13.5 为服务多平台所做的优化	178
13.6 未来的优化方向	179
13.7 总结	180
14 美丽联合集团支付系统架构演进	182
14.1 概述	182
14.2 支付系统 1.0	182

14.2.1 业务问题	183
14.2.2 系统问题	184
14.2.3 资金问题	185
14.3 支付系统 2.0	185
14.3.1 拆分系统边界	185
14.3.2 支付系统 2.0 系统详解	187
14.3.3 统一平台业务上下文	192
14.3.4 直面数据一致性挑战	192
14.4 性能提升	194
14.5 稳定性提升	201
14.5.1 监控先行	201
14.5.2 分离核心链路	201
14.5.3 服务依赖梳理	203
14.5.4 限流降级	203
14.6 压测	204
14.7 成效	204
14.7.1 业务支撑能力	204
14.7.2 系统服务能力	205
14.8 总结	206
15 金融撮合架构	207
15.1 概述	207
15.2 系统总体设计	208
15.2.1 系统核心模块	208
15.2.2 撮合算法设计	209
15.2.3 基于内存撮合	212
15.2.4 灾备的多机设计	212
15.2.5 状态机复制	213
15.2.6 系统架构	215
15.3 总结	218

第4篇 面向性能的架构

16	一线架构师带你玩性能优化	220
16.1	什么是系统优化	220
16.2	系统优化的方法论、思路 and 原则	220
16.2.1	常用方法论	221
16.2.2	优化思路	221
16.2.3	优化原则	222
16.3	性能优化	222
16.3.1	常见的性能问题	222
16.3.2	性能优化的具体工作	223
16.3.3	JVM 优化	230
16.3.4	数据库优化	233
16.3.5	负载均衡优化	235
16.4	总结	237
17	性能优化的常见模式及趋势	238
17.1	性能优化的优缺点	238
17.2	性能优化的两种模式	239
17.3	单应用优化	239
17.3.1	优化的基本思路	239
17.3.2	确定性能瓶颈/热点的常见方法	240
17.3.3	压测时通常观察的内容及其所使用的工具	240
17.3.4	常见的优化手段及模式	240
17.4	结构型优化	241
17.5	两个结构型优化的案例	245
17.5.1	处理单点/网络瓶颈的可行方式	245
17.5.2	处理数据库连接池瓶颈的可行手段	245
17.6	总结	247
18	性能优化之几种常见的压测模型及其优缺点	248
18.1	压测模型抽象	248

18.1.1	压测环境准备	249
18.1.2	压测任务准备	249
18.1.3	压测执行闭环	249
18.2	线下压测模型	250
18.3	线上引流压测模型	251
18.4	线上全链路压测模型	252
18.5	总结	253
19	缓存为王——无线缓存架构优化	254
19.1	无线缓存的定义、限制条件及影响	254
19.2	无线缓存要从全局考虑	254
19.2.1	服务器端架构对无线缓存的影响	254
19.2.2	无线缓存对服务器端架构的影响	255
19.2.3	大流量下无线缓存作用的劣化	256
19.2.4	无线缓存与本机移动端组件的关系	256
19.2.5	无线缓存存储介质的选择	257
19.3	数据、资源缓存及失效策略	257
19.3.1	架构详论	257
19.3.2	实现失效策略	259
19.4	总结	263

第 1 篇 架构方法论

Mary Shaw 在《软件体系结构：一门初露端倪学科的展望》（清华大学出版社）一书中论及软件系统的架构时，将系统描述为组件及组件之间的交互，而“Rational 统一过程”重点表达的观点是，软件架构包含了关于以下问题的重要决策：

- 如何对软件系统进行组织。
- 如何选择组成系统的结构元素和它们之间的接口，以及如何设计这些元素相互协作时所体现的行为。
- 如何组合这些元素，使它们逐渐合成更大的子系统。
- 如何让用户知道这个系统组织的架构风格：这些元素及它们之间的接口、协作和组合。

因此架构素来有“组成派”和“决策派”之争。其实架构兼具组成和决策的特点，另外架构设计也不仅仅是靠组成和决策两部分就能完全概括的。比如，什么是抽象，如何设计闭环架构，复杂的治理如何演进，等等很多话题。本章试图从架构方法论的角度略叙述一二：

- 有关架构的概念认知——概念辨析。
- 以终为始的架构设计——从哪里来，到哪里去。
- 闭环架构方法——论反馈。
- 复杂与架构演进的关系。
- 架构师的核心能力。

如果通过这几部分的论述，能让读者对架构方法论有较为基本的认知，作者们就欢欣了。