自成一派的架构设计方法论
教你体系化的架构设计思维

Broadview®
www.broadview.com.cn

软件架构设计

大型网站技术架构与业务架构融合之道

余春龙 / 著

中国工信出版集团
電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

软件架构设计

大型网站技术架构与业务架构融合之道

余春龙 / 著

電子工業出版社
Publishing House of Electronics Industry
北京•BEIJING

内 容 简 介

本书系统化地阐述了技术架构与业务架构的方法论与实践。本书内容分为 5 大部分，第 1 部分从行业背景出发定义架构的概念与范畴；第 2 部分细致讨论架构所需的计算机功底，包括编程语言、操作系统、数据库、网络、框架、中间件；第 3 部分从高并发、高可用、稳定性、分布式事务、Paxos/Raft 一致性算法、CAP 理论等方面探讨技术架构；第 4 部分从业务架构思维、微服务、领域驱动设计、技术架构与业务架构融合的角度探讨业务架构；第 5 部分从个人素质、团队能力两大方面，诠释从技术到管理的转变方法。通过本书，读者可以对大型业务系统的架构方法论有全局的认识，同时对软件架构的核心能力有深刻的理解，对个人的技术成长起到一定的借鉴作用。

本书不仅适合工程师、架构师阅读，也适合企业系统开发人员在内的软件开发从业人员阅读。

图书在版编目（CIP）数据

软件架构设计：大型网站技术架构与业务架构融合之道 / 余春龙著. —北京：电子工业出版社，2019.2
ISBN 978-7-121-35603-2

Ⅰ. ①软… Ⅱ. ①余… Ⅲ. ①网站建设－软件开发－架构 Ⅳ. ①TP393.092.2

中国版本图书馆 CIP 数据核字（2018）第 296162 号

责任编辑：宋亚东
印　　刷：三河市君旺印务有限公司
装　　订：三河市君旺印务有限公司
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：16　字数：321.5 千字
版　　次：2019 年 2 月第 1 版
印　　次：2019 年 4 月第 2 次印刷
定　　价：79.00 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。

前　言

为什么写本书？

当我在写本书时，脑子里还会浮现当初读研究生期间一个劲儿地啃 UML 建模、软件架构设计书籍的情景。对于当时一个没有太多项目经历的人来说，这种理论知识显得又晦涩又抽象。但也正是这种“过早熏陶”，使得我在工作后从事的一个接一个的项目中，会去“多想”一些架构方面的事情。

为什么说是“多想”了呢？稍微有些职场经验的人都知道，无论是在面试还是日常工作中，在技术方面大家更多谈论的是语言、数据结构与算法、操作系统原理、某种框架或中间件的原理与使用方式等这些“硬”性的东西，因为这些“硬”性的东西容易表述，里面的学问深浅也容易衡量。而软件建模、架构设计这种“软”性的东西，就不那么容易衡量了。大家都知道它们很重要，但又说不清楚里面到底包含了哪些学问，所以谈论这些东西通常都比较“虚”。最终就是大家很少在方法论方面谈论它们，而是等到项目中具体问题具体解决，这非常符合实用主义思维。

另一方面，随着互联网的发展，很多大型网站或系统要处理海量的用户访问，需要解决高并发、高可用和由此带来的数据一致性问题，这也使得大家把大部分精力都用在了解决这些问题上面。我把这些问题称为“显性问题”，因为如果解决不好，会造成系统宕机，用户体验受损，给企业带来严重损失，大家都能意识到这种问题很重要。解决的思路通常有两个：第一，利用分布式系统的特性，不断地分拆，把大系统拆小，降低风险，各个击破；第二，小步快跑，快速迭代，设计不合理没关系，可以不断重构，不断发布新版本。

但还有一类问题是“隐性问题”，是指系统的可重用性、可扩展性、可维护性等。因为一个系统由于设计得不好导致研发人力的投入和时间成本的增加，往往是没有办法显式地衡量的。可能不是系统设计得不好，而是业务本身就很复杂，又或者各部门之间的沟通协调问题，所以导致开发效率低。再说，即使系统设计得不好，做新功能有沉重的历史包袱，还能通过加班加点解决。但其实“隐性问题”比“显性问题”影响更大，因为它会让技术拖业务后腿，当有新需求的时候，系统无法跟随业务快速变化。

本书不想偏废二者中的任意一个。因为对于一个系统来说，可能既面临高并发高可用的技术问题，又面临复杂的业务问题，如何很好地处理二者的关系，从而打通技术和业务的任督二脉，是本书想要探讨的。

如何阅读本书？

架构是一种综合能力，而不是某一方面的技能。也正因为如此，本书提供的是一个全面的解决方案、方法论、成体系的设计思维。因此，本书将从基础技术谈起，再到高层技术、再到业务、管理，提供一个架构能力的全局视图，从而让大家明白一个架构师的能力模型究竟是什么样的。

具体来说，全书分为 5 大部分：

第 1 部分：从行业背景出发，对架构做一个宏观概述。让读者知道，当我们说架构的时候，都在说什么。

第 2 部分：计算机功底。功底非常重要，这是做架构的基本门槛。大学的教科书上教的全是功底，但经过多年实践之后，再回过头看书本内容，体会完全不一样。

第 3 部分：技术架构。这部分是纯技术，讲如何应对高并发、高可用、一致性方面的问题。

第 4 部分：业务架构。在这部分，我们将看到如何从技术延展到业务，如何跳出技术细节去抽象思考问题，如何通过业务建模把技术和业务进行融合。

第 5 部分：从职业发展的角度，从技术延展到管理。建立起对公司、商业、团队管理的一些基本认知。

对于刚入行的新人来说，建议从头看到尾，从而对架构的能力体系有一个全面认知；对于有经验的从业者，可以选取自己感兴趣的章节翻看。

由于编写时间紧张，书中难免存在不足之处，望广大读者批评指正。意见和建议请发送至邮箱 272142606@qq.com。

作　者

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。

- **下载资源**：本书如提供示例代码及资源文件，均可在 下载资源 处下载。
- **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/35603

目　录

第 1 部分　什么是架构

第 2 部分　计算机功底

第 3 部分 技术架构之道

第 4 部分 业务架构之道

第 5 部分　从架构到技术管理

第1部分　什么是架构

因为不同的技术方向、业务领域和公司对于架构的定义有很大差异，所以如果给架构下一个精确定义，要么会片面、局限，要么会过于抽象、大而空洞。

即使如此，作者还是想尝试用一句话来抽象：架构是针对所有重要问题做出的重要决策。很显然，不同公司或同一家公司的不同历史阶段面临的“重要问题”也不同，所以架构所做的事情自然也不一样。

在这样的认知基础上，我们将先从架构的职业分类、技术方向分类、架构的道与术等方面对架构做一个概览；之后，着眼于“大型网站”或者“大型的在线业务系统”领域，系统化地探讨这个领域的架构包含了哪些思路和方法。

当然，这些思路和方法在其他技术领域也同样适用，正所谓大道相通。

第1章 五花八门的架构师职业

1.1 架构师职业分类

随便找一个招聘网站或者猎头发布的招聘广告，我们就能看到各式各样的架构师头衔：Android/iOS 架构师、PHP 架构师、Java 架构师、前端架构师、后端架构师、数据架构师、搜索架构师、中间件架构师、大数据架构师……五花八门，不一而足。

然后看用人单位对应聘者工作年限的要求，有些是 3~5 年，有些是 8~10 年。

从这些岗位需求可以看出，“架构师”“架构”其实都是很“虚”的词，不同技术领域和行业对员工要求的能力模型和工作经验差异很大，不是能用一个简单的“架构师”就可以概括的。

本书并不是要把所有不同种类的架构所需要的能力逐一分析一遍，而是希望借业务架构和技术架构的融合，来建立一种系统化的思维方式和学习方法。这种系统化的思维方法既可以帮助开发人员形成系统化的方法论，也可以为在校学生和刚进入职场的开发人员起到一个引路作用，在职业发展过程中少走弯路。

1.2 架构的分类

撇开市场上招聘岗位的分类，单纯从技术角度来看，把软件系统自底往上分层，通常会得到如图 1-1 所示的软件系统架构分层。

1. 第一层：基础架构

基础架构指云平台、操作系统、网络、存储、数据库和编译器等。随着目前云计算越来越普及，很多的中小型公司都选择了大公司的云计算平台，而不是自己研发和维护基础架构。

2. 第二层：中间件与大数据平台

（1）中间件架构。例如分布式服务中间件、消息中间件、数据库中间件、缓存中间件、监控系统、工作流引擎和规则引擎等。

（2）大数据架构。例如开源的 Hadoop 生态体系，Hive、Spark、Storm、Flink 等。

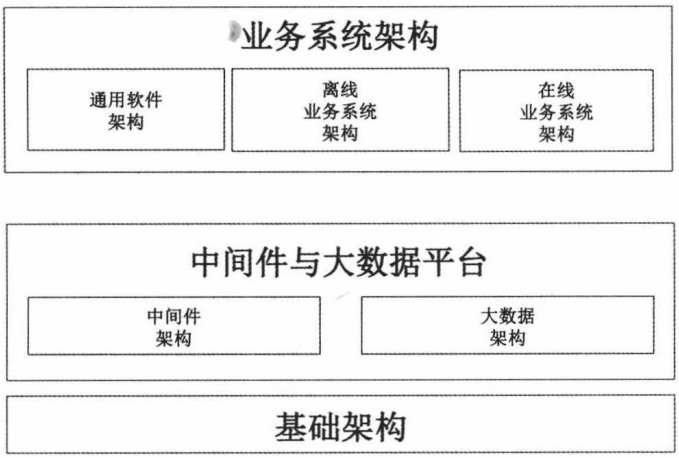

图 1-1　软件系统架构分层

3．第三层：业务系统架构

（1）通用软件系统。例如最常用的办公软件、浏览器、播放器等。

（2）离线业务系统。例如各种基于大数据的 BI 分析、数据挖掘、报表与可视化等。

（3）大型在线业务系统。例如搜索、推荐、即时通信、电商、游戏、广告、企业 ERP 或 CRM 等。

关于架构的这种分类方法，有两点需要说明：

- 对于中小型公司，可能没有第二层，或者即便有第二层，也只有很小的一部分。对于大型公司，在第二层和第三层的处理策略上也不一样：有些公司会让业务团队同时做第二层的工作，做在线业务系统的同时做了中间件的工作，做大数据业务系统的同时搭建和维护了大数据架构；有些公司会安排专门的团队做中间件与大数据平台，供上层各个业务系统使用。当然，实际也没有这么绝对，某个业务团队如果觉得中间件团队所做的工作不能满足业务需求，可能会选择自己造轮子。
- 对于第三层的划分，此处并不是很绝对，因为现实中软件的种类实在太多，比如嵌入式系统。同样，通用性软件和业务软件的界限也并不是泾渭分明的。一个业务系统随着技术的进步，很多功能将被通用化、标准化，最终变成了一个通用系统。比如搜索，在以前是一个专业性很强的业务系统，随着搜索技术的不断进步，现在搜索的很多功能已经被通用化了，有了 ES 这样的搜索平台，可以服务于电商、广告等其他各种业务，而不仅限于搜索本身。再比如权限控制、工作流引擎、规则引擎等，以前只是在某个业务系统里使用。后来大家发现很多业务系统里都需要类似的东西，于是抽象出来，成了通用的业务中间件。通用化的过程是一个技术不断进步的过程，也是一个使用门槛不断被降低的过程。

本书聚焦在大型在线业务系统的架构，即图 1-1 中第三层的第三部分。对于大规模的在线业务系统，一方面要处理高并发、高可用等技术问题；另一方面要面对各种复杂的业务需求，并且这些需求还在一直变化。如何把业务和技术很好地结合起来，处理好两者的关系，是本书要重点探讨的一个方向。

但本书并不只讲第三层，相反，要从第一层讲起。因为只有对下面的原理有了深刻理解，才可能对上面所构建的业务系统有深刻认识。

但需要面对的现实是，本书不可能同时把基础架构和业务架构讲得很透彻。一方面，作者不是基础架构方面的专家；另一方面，任何一个基础架构的系统，或者中间件、大数据的系统，都是一个很专业的子领域，钻进去都可以耗尽一个人毕生的精力。

第2章 架构的道与术

2.1 何为道，何为术

假设用 Java 语言绘制一个大型网站，其典型架构如图 2-1 所示。

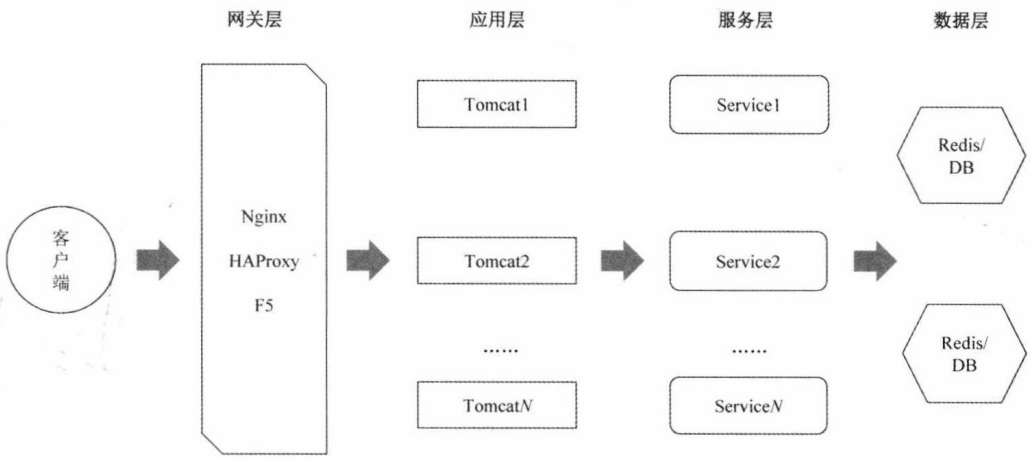

图 2-1 大型网站的典型架构

但这张图并不能说明什么问题，因为实际的架构决策并不能在这张图上反映出来。比如：

- 如何拆分服务？
- 如何组织服务与服务之间的层次关系？
- 如何设计接口？
- 缓存数据结构与更新策略是什么样的？
- 缓存宕机后系统是否可用？
- 数据库如何分库分表？
- 消息队列在什么地方使用？
- 需要部署多少台 Tomcat？需要部署多少台 RPC 服务？

……