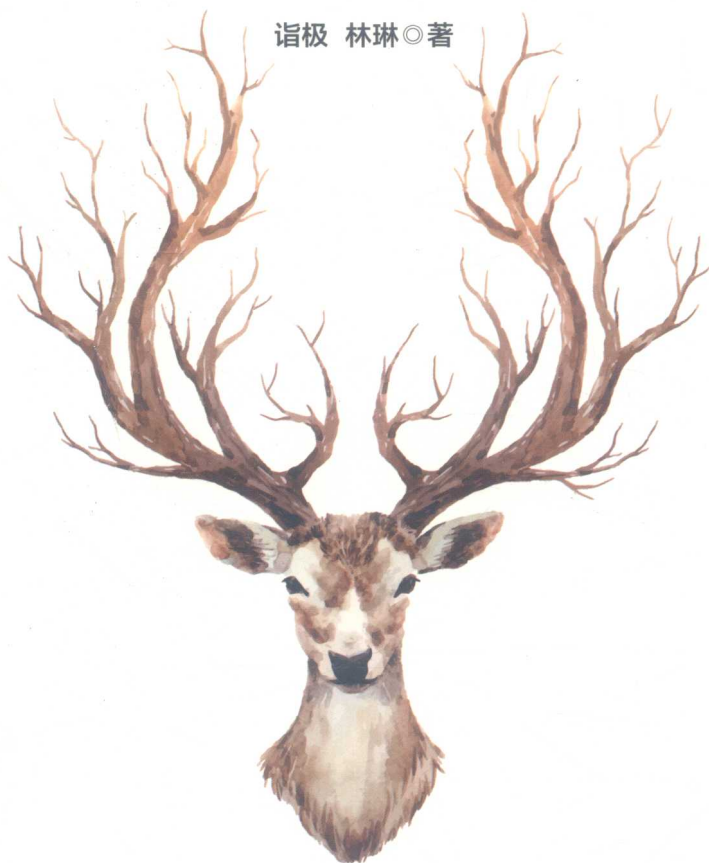


深入理解 Apache Dubbo 与实战

诣极 林琳◎著



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

深入理解 Apache Dubbo 与实战

诣极 林琳◎著



电子工业出版社

Publishing House of Electronics Industry

北京•BEIJING

内 容 简 介

本书首先介绍 Dubbo 的简史、后续的规划和整体架构大图；接着介绍 Dubbo 环境配置，并基于 Dubbo 开发第一款应用程序；然后介绍 Dubbo 内置的常用注册中心的实现原理，Dubbo 扩展点加载的原理和实现，Dubbo 的启动、服务暴露、服务消费和优雅停机的机制，Dubbo 中 RPC 协议细节、编解码和服务调用实现原理，Dubbo 集群容错、路由和负载均衡机制，Dubbo 的扩展点相关知识，Dubbo 高级特性的实现和原理，Dubbo 常用的 Filter 的实现原理，Dubbo 中新增 etcd3 注册中心的实战内容和 Dubbo 服务治理平台的相关知识；最后介绍 Dubbo 未来生态和 Dubbo Mesh 的相关知识。

本书适合对 Dubbo 有一定了解、对 Dubbo 框架感兴趣的读者，也适合想深入理解 Dubbo 原理的资深开发者阅读。

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

深入理解 Apache Dubbo 与实战 / 诣极，林琳著. —北京：电子工业出版社，2019.7
ISBN 978-7-121-36634-5

I. ①深… II. ①诣… ②林… III. ①分布式操作系统 IV. ①TP316.4

中国版本图书馆 CIP 数据核字（2019）第 100420 号

责任编辑：陈晓猛

印 刷：三河市良远印务有限公司

装 订：三河市良远印务有限公司

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：18.75

字数：360 千字

版 次：2019 年 7 月第 1 版

印 次：2019 年 9 月第 3 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 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。

序一

近年来，随着业务规模的发展和复杂度的增加，传统的单体应用已经很难适应业务迭代的诉求，越来越多的公司开始进行服务化的改造。很高兴看到 Apache Dubbo 被许多公司采用，作为服务化改造的基础架构进行演进。这里面就包括了许多互联网公司、国字头的大型企业，以及金融行业的巨头公司。Apache Dubbo 因为良好的设计和扩展性受到许多开发者的欢迎，然而当开发者需要深入了解 Dubbo 底层的架构设计和实现的时候，往往会有些不知所措。网上也有很多爱好者撰写的源码分析等文章，虽有所有启发和裨益，但总觉得不够成体系。令人遗憾的是，市面上始终缺乏一本完整的、体系化的对 Apache Dubbo 进行深入原理剖析的书。

本书的出现填补了这一空白，本书由浅入深，娓娓道来。既有对 Dubbo 基本概念的讲解、底层原理的分析，同时又不乏生动的实战内容。作为进阶的内容，对扩展点、过滤器，以及 Dubbo 的高级特性，都有深入浅出的介绍。非常适合想了解 Dubbo，对 Dubbo 内部实现感兴趣的读者阅读。

我和本书作者因为共同参与 Dubbo 社区而认识，彼时 Dubbo 刚刚加入 Apache 基金会进行孵化。在参与社区开发的过程中，经常有过深入的交流，在此过程中，本书作者所展现出来的对 Dubbo 内核机制透彻的理解，以及对细节的准确把握，让我受益匪浅。正式因为本书作者出色的贡献，他也成为 Dubbo 加入 Apache 孵化器之后首个 committer（提交者，对代码具有提交权限），后续更是成为首个 PMC Member（项目管理委员会成员，对项目发展具有决定性的投票权）。由他提笔来撰写本书，我认为再合适不过。在阅读本书的过程中，我收获颇丰。

工欲善其事，必先利其器。希望更多的读者因为本书而受益，也希望有更多的人阅读本书之后，了解 Dubbo，参与到社区中，一起把社区发展得越来越好。

写在 Dubbo 从 Apache 基金会毕业之际。

张乎兴，阿里云技术专家，

Apache Member, Apache Dubbo/Tomcat PMC Member

序二

过去十多年互联网产业的高速发展，在给社会带来深刻变革的同时，也催生了服务架构的演进：从传统的单体应用到面向服务的 SOA，再到现今主流的微服务架构，而 Apache Dubbo 就是微服务领域中的先行者和佼佼者。

Apache Dubbo 是阿里巴巴于 2011 年开源的一款高性能 Java RPC 框架，开源伊始就在业界产生了很大的影响，被大量公司广泛使用，甚至在很多公司自研的 RPC 框架中也能看到不少 Dubbo 的影子，可以说 Dubbo 在国内服务化体系演进过程中扮演了一个非常重要的角色。虽然中间经历了几年时间的沉寂，不过 2017 年阿里巴巴又重启了对 Dubbo 的开源维护，受到了社区的广泛欢迎，社区活跃度也随之迅速提升，Dubbo 也正在从一个微服务领域的高性能 RPC 框架，逐步演变为一个完整的微服务生态。

之前由于工作关系，对 Dubbo 有过比较深入的研究，在阅读 Dubbo 的源码过程中，深深体会到了作者的匠心设计和深厚功底。我们知道服务往往是一家公司技术体系中最核心的部分，是整个业务的基石，所以作为一款涵盖了服务交互全过程的通用 RPC 框架，就必须能非常简单、无侵入地支持业务对服务调用过程中的各个阶段做扩展和定制，比如通信协议、序列化格式、路由策略、负载均衡、监控运维等。而 Dubbo 通过微内核设计、SPI 扩展的方式完美地解决了这一难题，不管是 Dubbo 自身的特性，还是业务方的扩展，都统一通过 SPI 来实现，在 Dubbo 内核面前都是“一等公民”，从而保证了框架自身的可持续性和稳定性。有特殊需求的业务或公司可以在不修改一行 Dubbo 源码的情况下加载自己的扩展或接入自己的产品和运维体系，这一点非常值得我们学习。

和诣极认识也是源于 Dubbo，他是 Apache Dubbo 项目的 PMC，为 Dubbo 贡献了很多有价值的特性，如 HTTP2 协议、etcd 注册中心等，现在有幸成为同事，对他也有了更深的了解。在我看来，他是一个很纯粹的技术人，熟读众多开源框架的源码并对其原理了然于胸，对技术也充满了热情，能在工作非常饱和的情况下坚持完成本书的写作，可见一斑。受其热情感染，所以当收到邀请为本书作序时，我也是毫不犹豫地就答应了。

收到书稿后第一时间细细通读了一遍，本书作为第一本体系化讲解 Apache Dubbo 的书籍，深入浅出地解释了 Dubbo 的工作原理，配以源码级解析，向读者展现了一款优秀的分布式中间

件的设计用心和实现细节，非常适合有志于成为优秀架构师的技术爱好者细细阅读和品味。同时本书在 Dubbo 的协议设计、编解码原理、线程模型等方面也做了深度剖析，使读者不仅知其然也能知其所以然，有助于理解分布式环境下的服务通信范式，对问题排查也会有很大益处。所以在此向大家推荐本书，相信不管是初学者还是有多年经验的资深工程师，通过阅读本书，都会有所收获。

宋顺，蚂蚁金服高级技术专家

开源配置中心 Apollo 作者

前言

本书的由来

在 Apache Dubbo（以下简称 Dubbo）重新开源之前，Dubbo 已经被很多公司广泛用于生产环境并获得了良好的反馈，很多公司内部也会建立私有分支自己维护，其中 Dubbox 就是基于 Dubbo 分支进行扩展并二次维护的。重新开源后，社区维护的 Dubbo 版本进行了大量“bug fix”和特性支持，收到了大量 Dubbo 用户的支持和参与。编写本书的想法是在开源后提出来的，因此本书取名《深入理解 Apache Dubbo 与实战》。

我最早接触 Dubbo 是 2015 年在参与云基础设施建设和微服务框架搭建时，那时的 Dubbo 基本是分布式系统的不二之选。我在使用过程中遇到了不少问题，不得不通过调试 Dubbo 源码来解决。在源码调试过程中，我逐渐理解 Dubbo 的设计理念，越发惊叹其架构设计的巧妙，正所谓：优秀的设计万里挑一。

后来我加入另外一家公司担任架构师，在基础架构组负责中间件与框架的研发，也是基于 Dubbo 等框架做二次开发。此时，我与本书另一位作者诣极成为同事，我们志趣相投，也为编写本书埋下了伏笔。

2017 年，Dubbo 经历了数年的停滞，终于又重启维护。我还依稀记得阿里巴巴公司宣布的那天诣极的兴奋状，诣极是首批投入 Apache Dubbo 开源工作中去的开发者，期间为 Apache Dubbo 贡献了众多特性，如 HTTP 2、etcd 注册中心等，现在已经成为 Apache Dubbo PMC。

Dubbo 在重启维护后，短时间内有了大量的更新。此时我正就职于蚂蚁金服，每天苦于繁重的 CRUD。我发现市面上没有一本系统讲解 Dubbo 的书籍，Dubbo 相关的资料大都是一些博客文章，并且内容也相对陈旧。于是我“怂恿”诣极与我一起写这本书，作为这些年来工作的一个总结，他也欣然答应。

然后就是历时半年的写作过程，一波三折。中途有一些朋友加入，又因为各种不可抗拒的原因退出。我和诣极平时的工作都非常饱和，又要关注开源工作，常常焦头烂额。道虽迩，不行不至；事虽小，不为不成。时间是一点一点挤出来的，最终我们还是坚持完成了本书，并且

没有偏离原计划太多。由于时间紧迫和能力有限，本书的内容难免有错漏，读者的勘误可以发送到以下电子邮箱：

yiji@apache.org 或 315157973@qq.com。

面向的读者

由于 Dubbo 重启维护后，官方的使用文档已经非常详细，本书如果再讲使用方法就没有太多的意义，所以一开始就按源码解析、设计原理说明的方向定位，因此需要读者有一定的基础。涉及源码的讲解，注定会有很多地方枯燥乏味，我们在编写的时候也尽量用自己的语言去提炼总结，让读者可以少看代码。

在开始动笔编写本书时，Apache Dubbo 2.7 正在开发中，考虑到代码的稳定性，我们最终决定基于公开发布的 2.6.5 版本（release）来写。当本书快完成的时候，Apache Dubbo 2.7 正式发布了。因此，本书的内容并不是基于最新的代码的，写书的速度永远比不上框架更新的速度，也希望读者能够谅解。

本书内容

- 第 1 章主要介绍 Dubbo 的简史、后续的规划和整体架构大图。
- 第 2 章主要介绍 Dubbo 的环境配置和基于 Dubbo 开发第一款应用程序。
- 第 3 章主要介绍 Dubbo 内置的常用注册中心的实现原理。
- 第 4 章主要介绍 Dubbo 扩展点加载的原理和实现。
- 第 5 章主要介绍 Dubbo 的配置解析、服务暴露、服务消费和优雅停机的机制。
- 第 6 章主要介绍 Dubbo 的 RPC 协议细节、编解码和服务调用的实现原理。
- 第 7 章主要介绍 Dubbo 的集群容错、路由和负载均衡机制。
- 第 8 章主要介绍 Dubbo 扩展点的相关知识。
- 第 9 章主要介绍 Dubbo 高级特性的实现和原理。
- 第 10 章主要介绍 Dubbo 过滤器的实现原理。
- 第 11 章主要介绍 Dubbo 中新增的 etcd3 注册中心的实战内容。
- 第 12 章主要介绍 Dubbo 服务治理平台的相关知识。
- 第 13 章主要介绍 Dubbo 的未来生态和 Dubbo Mesh 相关知识。

致谢

首先感谢我的领导与同事，也感谢我的团队，你们给了我很大的帮助。

然后要感谢我妻子，在我写作期间，她承担了家里的大小事务，让我可以有更多的时间投入写作。如果没有妻子的支持，那么这本书我肯定是无法完成的。

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。

本书页面入口：<http://www.broadview.com.cn/36634>。



目录

第 1 章 Dubbo——高性能 RPC 通信框架	1
1.1 应用架构演进过程.....	1
1.1.1 单体应用	1
1.1.2 分布式应用	3
1.2 Dubbo 简介.....	6
1.2.1 Dubbo 的发展历史.....	7
1.2.2 Dubbo 是什么	7
1.2.3 Dubbo 解决什么问题.....	9
1.2.4 谁在使用 Dubbo	10
1.2.5 Dubbo 后续的规划.....	11
1.3 Dubbo 总体大图.....	11
1.3.1 Dubbo 总体分层	11
1.3.2 Dubbo 核心组件	12
1.3.3 Dubbo 总体调用过程.....	13
1.4 小结	15
第 2 章 开发第一款 Dubbo 应用程序	16
2.1 配置开发环境	16
2.1.1 下载并安装 JDK	17
2.1.2 下载并安装 IDE.....	17
2.1.3 下载并配置 Maven	18
2.1.4 下载并配置 ZooKeeper	18
2.1.5 使用 IDEA 调试 Dubbo 源码	18
2.2 基于 XML 配置实现.....	21
2.2.1 编写 Echo 服务器	21
2.2.2 编写 Echo 客户端	24
2.3 基于注解实现	26
2.3.1 基于注解编写 Echo 服务器	26

2.3.2 基于注解编写 Echo 客户端	28
2.4 基于 API 实现	30
2.4.1 基于 API 编写 Echo 服务器	30
2.4.2 基于 API 编写 Echo 客户端	31
2.5 构建并运行	32
2.6 小结	34
第 3 章 Dubbo 注册中心	35
3.1 注册中心概述	35
3.1.1 工作流程	36
3.1.2 数据结构	37
3.1.3 ZooKeeper 原理概述	37
3.1.4 Redis 原理概述	39
3.2 订阅/发布	40
3.2.1 ZooKeeper 的实现	40
3.2.2 Redis 的实现	44
3.3 缓存机制	48
3.3.1 缓存的加载	49
3.3.2 缓存的保存与更新	50
3.4 重试机制	50
3.5 设计模式	51
3.5.1 模板模式	51
3.5.2 工厂模式	52
3.6 小结	54
第 4 章 Dubbo 扩展点加载机制	55
4.1 加载机制概述	55
4.1.1 Java SPI	56
4.1.2 扩展点加载机制的改进	57
4.1.3 扩展点的配置规范	59
4.1.4 扩展点的分类与缓存	60
4.1.5 扩展点的特性	61
4.2 扩展点注解	62
4.2.1 扩展点注解: @SPI	62
4.2.2 扩展点自适应注解: @Adaptive	63
4.2.3 扩展点自动激活注解: @Activate	65
4.3 ExtensionLoader 的工作原理	66
4.3.1 工作流程	66
4.3.2 getExtension 的实现原理	67

4.3.3	getAdaptiveExtension 的实现原理	70
4.3.4	getActivateExtension 的实现原理	73
4.3.5	ExtensionFactory 的实现原理	73
4.4	扩展点动态编译的实现	76
4.4.1	总体结构	77
4.4.2	Javassist 动态代码编译	78
4.4.3	JDK 动态代码编译	79
4.5	小结	80
第 5 章	Dubbo 启停原理解析	81
5.1	配置解析	81
5.1.1	基于 schema 设计解析	82
5.1.2	基于 XML 配置原理解析	85
5.1.3	基于注解配置原理解析	91
5.2	服务暴露的实现原理	97
5.2.1	配置承载初始化	97
5.2.2	远程服务的暴露机制	97
5.2.3	本地服务的暴露机制	105
5.3	服务消费的实现原理	106
5.3.1	单注册中心消费原理	106
5.3.2	多注册中心消费原理	113
5.3.3	直连服务消费原理	114
5.4	优雅停机原理解析	115
5.5	小结	116
第 6 章	Dubbo 远程调用	117
6.1	Dubbo 调用介绍	117
6.2	Dubbo 协议详解	119
6.3	编解码器原理	122
6.3.1	Dubbo 协议编码器	123
6.3.2	Dubbo 协议解码器	128
6.4	Telnet 调用原理	136
6.4.1	Telnet 指令解析原理	136
6.4.2	Telnet 实现健康监测	140
6.5	ChannelHandler	141
6.5.1	核心 Handler 和线程模型	141
6.5.2	Dubbo 请求响应 Handler	145
6.5.3	Dubbo 心跳 Handler	148
6.6	小结	150

- 第 7 章 Dubbo 集群容错 151
 - 7.1 Cluster 层概述 151
 - 7.2 容错机制的实现 153
 - 7.2.1 容错机制概述 153
 - 7.2.2 Cluster 接口关系 155
 - 7.2.3 Failover 策略 157
 - 7.2.4 Failfast 策略 158
 - 7.2.5 Failsafe 策略 158
 - 7.2.6 Failback 策略 159
 - 7.2.7 Available 策略 160
 - 7.2.8 Broadcast 策略 160
 - 7.2.9 Forking 策略 161
 - 7.3 Directory 的实现 162
 - 7.3.1 总体实现 162
 - 7.3.2 RegistryDirectory 的实现 163
 - 7.4 路由的实现 166
 - 7.4.1 路由的总体结构 166
 - 7.4.2 条件路由的参数规则 167
 - 7.4.3 条件路由的实现 168
 - 7.4.4 文件路由的实现 169
 - 7.4.5 脚本路由的实现 170
 - 7.5 负载均衡的实现 171
 - 7.5.1 包装后的负载均衡 171
 - 7.5.2 负载均衡的总体结构 173
 - 7.5.3 Random 负载均衡 175
 - 7.5.4 RoundRobin 负载均衡 176
 - 7.5.5 LeastActive 负载均衡 178
 - 7.5.6 一致性 Hash 负载均衡 179
 - 7.6 Merger 的实现 181
 - 7.6.1 总体结构 181
 - 7.6.2 MergeableClusterInvoker 机制 183
 - 7.7 Mock 185
 - 7.7.1 Mock 常见的使用方式 185
 - 7.7.2 Mock 的总体结构 186
 - 7.7.3 Mock 的实现原理 187
 - 7.8 小结 189

第 8 章 Dubbo 扩展点	190
8.1 Dubbo 核心扩展点概述	190
8.1.1 扩展点的背景	191
8.1.2 扩展点整体架构	191
8.2 RPC 层扩展点	192
8.2.1 Proxy 层扩展点	192
8.2.2 Registry 层扩展点	194
8.2.3 Cluster 层扩展点	195
8.3 Remote 层扩展点	198
8.3.1 Protocol 层扩展点	199
8.3.2 Exchange 层扩展点	202
8.3.3 Transport 层扩展点	203
8.3.4 Serialize 层扩展点	206
8.4 其他扩展点	207
第 9 章 Dubbo 高级特性	210
9.1 Dubbo 高级特性概述	210
9.2 服务分组和版本	211
9.3 参数回调	214
9.4 隐式参数	217
9.5 异步调用	218
9.6 泛化调用	219
9.7 上下文信息	220
9.8 Telnet 操作	221
9.9 Mock 调用	224
9.10 结果缓存	226
9.11 小结	226
第 10 章 Dubbo 过滤器	227
10.1 Dubbo 过滤器概述	227
10.1.1 过滤器的使用	228
10.1.2 过滤器的总体结构	228
10.2 过滤器链初始化的实现原理	231
10.3 服务提供者过滤器的实现原理	233
10.3.1 AccessLogFilter 的实现原理	233
10.3.2 ExecuteLimitFilter 的实现原理	234
10.3.3 ClassLoaderFilter 的实现原理	235
10.3.4 ContextFilter 的实现原理	237

10.3.5	ExceptionHandler 的实现原理	237
10.3.6	TimeoutFilter 的实现原理	238
10.3.7	TokenFilter 的实现原理	238
10.3.8	TpsLimitFilter 的实现原理	239
10.4	消费者过滤器的实现原理	240
10.4.1	ActiveLimitFilter 的实现原理	240
10.4.2	ConsumerContextFilter 的实现原理	242
10.4.3	DeprecatedFilter 的实现原理	242
10.4.4	FutureFilter 的实现原理	243
10.5	小结	244
第 11 章	Dubbo 注册中心扩展实践	245
11.1	etcd 背景介绍	245
11.2	etcd 数据结构设计	246
11.3	构建可运行的注册中心	248
11.3.1	扩展 Transporter 实现	248
11.3.2	扩展 RegistryFactory 实现	249
11.3.3	新增 JEtcdClient 实现	250
11.3.4	扩展 FailbackRegistry 实现	260
11.3.5	编写单元测试	263
11.4	搭建 etcd 集群并在 Dubbo 中运行	263
11.4.1	单机启动 etcd	264
11.4.2	集群启动 etcd	265
11.5	小结	266
第 12 章	Dubbo 服务治理平台	267
12.1	服务治理平台总体结构	267
12.2	服务治理平台的实现原理	269
12.3	小结	273
第 13 章	Dubbo 未来展望	274
13.1	Dubbo 未来生态	274
13.1.1	开源现状	274
13.1.2	后续发展	275
13.2	云原生	281
13.2.1	面临的挑战	281
13.2.2	Service Mesh 简介	283
13.2.3	Dubbo Mesh	284
13.3	小结	285



第 1 章

Dubbo——高性能 RPC 通信框架

本章主要内容：

- 应用架构演进过程；
- Dubbo 简介；
- Dubbo 总体大图。

本章主要是对 Dubbo 总体的介绍，让读者对 Dubbo 有一个总体的认识。首先介绍后台应用架构的演进过程，从最初的 JEE 到现在的微服务架构都会介绍；然后简单介绍一下 Dubbo，包括它的发展历史、未来方向等；最后讲解 Dubbo 的总体大图，通过分层的方式讲解 Dubbo 的总体架构，并介绍 Dubbo 的核心组件及总体流程。

1.1 应用架构演进过程

1.1.1 单体应用

1. JEE 时期

JEE 是 Java Platform Enterprise Edition 的简称，提供了企业级软件开发的运行环境与开发工

具。JEE 的出现极大地推动了企业信息化进程。

JEE 把企业软件划分为展示层、业务逻辑层和数据存储层，如图 1-1 所示。



图 1-1 JEE 层次划分

最终，各层的组件会被聚合到一起运行在通用的服务器上，如 JBoss、WebSphere、Tomcat 等。注意，Tomcat 只实现了 JEE Web 部分的规范。JEE 应用会把大量通用的业务逻辑、流程封装到通用组件中，少量定制化逻辑则通过配置文件的方式来访问通用组件。

JEE 的优点和缺点都非常明显。优点：分层设计明确了不同团队的分工，职责清晰、分工明确，衍生出了前端团队、后端团队和 DBA 团队；JEE 的开发简单，所有的类都能直接本地引用及使用，事务的处理只需要依赖数据库即可。由于企业级应用通常面向内部用户，使用者较少，也不需要考虑高并发等场景，加上 JEE 的稳定性，所以基本能满足日常的需求。缺点：大多数应用都在一个 JVM 中，随着应用的增大，性能会不断下降；业务之间的耦合严重，即使企业内部有不同的规范和约束，但随着业务逻辑复杂度的增加，开发人员的不断流动，整个应用的维护会变得越来越难；EJB 使用大量的 XML 作为配置文件，后期要配置出一个服务的学习成本很高。各种各样的规范约束加大了开发者的开发成本。

2. MVC 框架时期

由于 EJB 的各种问题严重影响了软件开发效率，为了降低成本提高生产力，开源框架开始成为企业的标配，这个时期比较主流的框架有 Spring、Struts、Hibernate 等。架构也从之前的分层架构变为了 MVC 架构，如图 1-2 所示。

其中 Model 属于模型部分，主要负责具体的业务逻辑和数据存取；View 是视图，主要处理应用的展示部分；Controller 是控制器，主要处理用户的交互，它会从视图层读取数据并传递给模型。