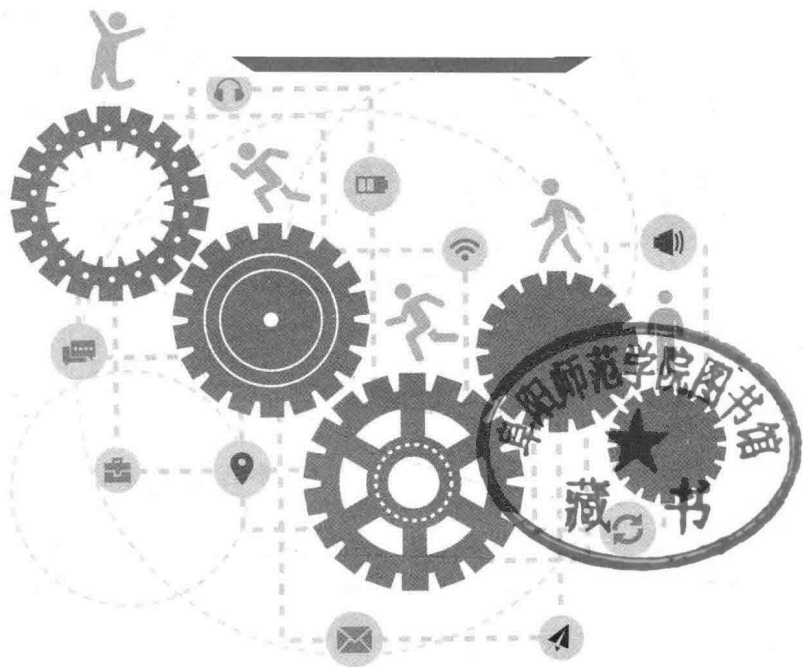


架构探险 从零开始写分布式服务框架

一线技术专家全方位解析分布式服务框架底层技术细节

李业兵 / 著



架构探险 从零开始写分布式服务框架

李业兵 / 著

电子工业出版社
Publishing House of Electronics Industry
北京·BEIJING

内 容 简 介

本书的初衷是希望把分布式服务框架的实现细节及分布式服务框架周边的知识点梳理清楚,为那些对分布式服务框架感兴趣的人打开一扇窗户,降低获取相关知识的门槛。所以本书围绕实现分布式服务框架所需的知识点,进行了比较详尽细致的介绍。包括常见的RPC框架、常见的序列化/反序列化方案及选型、分布式服务框架服务的发布引入实现细节、软负载实现、底层通信方案实现、服务注册与发现实现、服务治理常见的功能等。通过对这些知识点的逐步讲解,层层深入,最终完成一个可运行的分布式服务框架。

通过这本书,读者可以完整地了解实现一个分布式服务框架的所有技术细节和实现原理,希望对想了解分布式服务框架实现细节的读者有所启发和帮助。

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

架构探险:从零开始写分布式服务框架/李业兵著. —北京:电子工业出版社,2017.7
ISBN 978-7-121-31959-4

I. ①架… II. ①李… III. ①分布式计算机系统—研究 IV. ①TP338.8

中国版本图书馆CIP数据核字(2017)第139692号

责任编辑:董英

印刷:三河市双峰印刷有限公司

装订:三河市双峰印刷有限公司

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 字数:365千字

版次:2017年7月第1版

印次:2017年7月第1次印刷

印数:3000册 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。



关于作者

李业兵

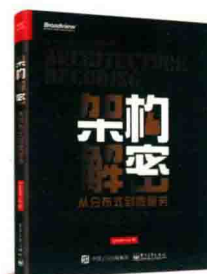
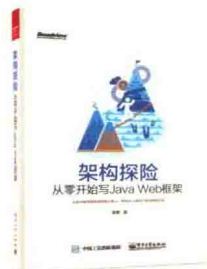
先后在支付宝运营支撑事业部、去哪儿网酒店事业部任职高级开发工程师。于2015年加入猫眼电影（原美团网旗下电影事业部），负责交易业务线架构与开发工作。

擅长电商交易领域系统设计与架构，在电商交易领域系统架构设计方面积累了较为丰富的实践经验。

对高并发系统设计、服务化架构、互联网中间件开发保持着浓厚的兴趣。

邮箱：liyebing1997@163.com

好书力荐



反馈意见或投稿，联系编辑：

邮箱：dongying@phei.com.cn

微信：yingzidd

试读结束：需要全本请在线购买：www.ertongbook.com

推荐序：做架构犹如去探险

2015 年，我写了自己的第一本书《架构探险：从零开始写 Java Web 框架》，书中记录了一款轻量级 Java Web 框架的整个开发过程。我将该框架取名为 Smart Framework，因为我认为 Java Web 框架应该更容易上手，运行得更快，更加轻量级。

为何我要写“架构探险”？

我希望有更多看过这本书的朋友，能够对 Java Web 框架的底层原理更加清楚，不仅能熟练使用框架，而且还能知道框架底层都做了些什么，做到“知其然，知其所以然”，我认为所有的开发者都应该这样要求自己。

此外，我认为学习技术的最有效方式就是分享，只有分享才能让自己得到更快的进步。但在分享之前，我们需要花时间去不断学习，这个学习过程也许是痛苦的，但当自己将所学技能分享出去的时候却是快乐的。不仅如此，我们通过分享还会收到大家更多的反馈，这些反馈能够帮助我们得到更加全面的成长。

架构和“探险”有何关系？

架构说简单点就是一堆技术、框架、工具的组合，至于怎么组合，这就非常考验架构师的经验和水平。一个优秀的架构，可以让开发效率变得更加高效，为企业节省更多的成本。程序员可将自己更多的精力放在业务需求的实现上，不会被底层的复杂技术细节所干扰。

架构师（或从事架构工作的人）就像是探险者一样，他们走在团队的前面，为大家铺路，带领大家找到成功捷径。此外，做架构工作不能照搬别人所谓的最佳实践，而要根据自身实际情况，因地制宜地灵活选择，设计最为合理的架构。架构的目的是为了让业务变得更加容易落地，降低开发成本与统一开发规范。架构师的职责就是避免大家踩坑，他们需要将自己的经验总结下来，并带领大家走最正确的路。架构师不只是在体验探险的凶险和快乐，而是把探险所积累的经验沉淀下来，让后面更多的人从中受益。

架构探险是一种信仰

其实我们都在架构中探险，或许自己目前正身陷险境，我们希望得到一本“宝典”，教会自己应该如何找到架构的成功捷径。李业兵老师写的这本《架构探险：从零开始写分布式服务框架》就能顺利地带领我们走出险境，让我们对分布式服务框架所涉及的技术了然于胸，并能合理地利用这些技术，搭建符合自身需求的分布式服务框架。

这本书传递了“架构探险”的精神，让我更加深刻地感受到，其实架构探险是一种信仰，它指引我们在架构之路上继续前进。

黄勇·特赞科技 CTO

2017年6月

前言

缘起

随着互联网浪潮风起云涌，互联网行业发展非常迅猛。此时将所有业务集中实现在一个应用上的做法已经满足不了公司及业务发展的需要了。基于面向服务体系架构来构建系统成了互联网架构师构建系统的不二选择，而面向服务体系架构能够落地的基础技术之一就是分布式服务框架。

要完全掌握分布式服务框架存在一定的技术门槛，市面上不乏一些非常出色的开源分布式服务框架。但对于新手而言，直接通过阅读源码来掌握分布式服务框架原理也并不是一件特别容易的事。

现在市面上也有专门的书籍来介绍分布式服务框架，但大都是从理论和方法论的角度来描述其原理的。有句俗语：“有些事，难不难，做了才知道；有些山，陡不陡，爬了才知道。”对于程序员来说，总是想通过具体的代码来了解一个分布式服务框架的实现细节，这样更为直观和深刻。为了帮助部分想了解分布式服务框架内部实现原理，甚至想自己实

现一个分布式服务框架的读者，我写了这样一本介绍如何从零开始写一个分布式服务框架的书，希望能够对想了解分布式服务框架实现细节的读者有所帮助。

内容大纲

全书一共 8 章。按照分布式服务框架的各个组成部分及各个组成部分所需的知识点或者这些知识点适当的延伸来组织每一章节的内容。建议读者按照全书章节的组织顺序来阅读。

第 1 章主要介绍日常开发常用的 RPC 框架，包括 RMI、CXF、Axis2、Thrift、gRPC、HttpClient，并就每一种 RPC 框架给出了实际可运行的代码示例，以及自己实现的一个简易版的 RPC 框架。

第 2 章对于基于服务体系架构做了介绍，对分布式服务框架总体架构及实现分布式服务框架所需的技术做了概要性介绍。

第 3 章介绍 9 种序列化/反序列化方案，每一种序列化/反序列化方案均给出了相应的代码示例，并给出了具体的选型建议。同时，将这 9 种序列化/反序列化实现集成在一起，实现了可配置化的序列化/反序列化引擎，最终整合在分布式服务框架实现内部。

第 4 章对 Spring 做了概要性介绍。有针对性地对 FactoryBean 周边知识及如何使用 FactoryBean 实现分布式服务的发布和引入做了详细介绍并给出代码实现。

第 5 章介绍 ZooKeeper 常用知识及如何使用 ZooKeeper 实现服务的注册与发现，并给出了具体的代码实现。

第 6 章围绕系统之间底层通信相关的知识点来组织，从 Java I/O 体系（阻塞 I/O、NIO、NIO2）到 Netty 相关知识均做了详细介绍。最后就使用 Netty 实现分布式服务框架底层通信给出了代码实现。

第 7 章介绍常用的软负载算法，并针对每一种算法给出了代码实现。同时将实现的多

种软负载算法集成可配置的软负载工具引擎，最终整合在分布式服务框架实现内部。

第8章介绍分布式服务框架服务治理相关的概念及方法论，并就部分服务治理功能给出了具体实现。

全书完整地实现了一个可以实际运行的分布式服务框架，全书所有代码均提供下载。

致谢

首先感谢我的妻子，在写书这段时间，宝宝的出生给了我人生中最好的礼物，宝宝的咿呀学语、一个不经意的笑容都能给我莫大的支持和鼓励。

同时，感谢猫眼电影公司的同事和领导，给了我宽松的学习与工作氛围，学到了很多知识，也得到了很多成长的机会。

最后，感谢辛苦劳作的编辑，本书能够出版有你们很大的一份功劳。

写在最后

这本书偏向实战，会有很多代码实现细节的描述，全书完整的代码实现会另给下载链接。本书所实现的分布式服务框架并未经历严苛生产环境的考验，定有很多不足之处，希望日后有机会再继续完善。写书对我来说是一个比较大的挑战，因为一个技术点，自己能理解和会用文字表达出来让别人也能理解是完全不同的层次。心中特别忐忑，担心因为自己对知识理解不够深入，以及文字表达水平不够，导致对读者有所误导。书中难免有错误和疏漏之处，在此恳请读者批评指正。

李业兵

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/31959>



目录

第 1 章 常用的 RPC 框架	1
1.1 RPC 框架原理	1
1.2 RMI 介绍	2
1.2.1 原生 RMI 代码示例	3
1.2.2 RMI 穿透防火墙	5
1.3 CXF/Axis2 介绍	7
1.3.1 CXF 介绍	7
1.3.2 Axis2 介绍	14
1.4 Thrift 介绍	21
1.4.1 Thrift 工作原理介绍	23
1.4.2 Thrift IDL 语法说明	26
1.4.3 基于 Apache Thrift 的 Java 版完整案例	28
1.4.4 基于 Java 注解的简化实现	36
1.5 gRPC 介绍	42
1.5.1 protobuf3 语法介绍	43
1.5.2 gRPC 使用示例	45

1.6	HTTP Client 介绍	53
1.6.1	构建 HttpClient 对象	54
1.6.2	构建 URI 对象	55
1.6.3	构建请求对象 (HttpGet、HttpPost)	56
1.6.4	HttpClient 发起调用及获取调用返回结果	56
1.7	实现自己的 RPC 框架	61
1.8	RPC 框架与分布式服务框架的区别	68
1.9	本章小结	68
第 2 章	分布式服务框架总体架构与功能	69
2.1	面向服务的体系架构 (SOA)	69
2.1.1	面向服务架构范式	69
2.1.2	服务拆分原则	71
2.2	分布式服务框架现实需求	72
2.3	分布式服务框架总体架构及所需的技术概述	72
2.4	本章小结	74
第 3 章	分布式服务框架序列化与反序列化实现	75
3.1	序列化原理及常用的序列化介绍	75
3.2	Java 默认的序列化	77
3.3	XML 序列化框架介绍	80
3.4	JSON 序列化框架介绍	82
3.5	Hessian 序列化框架介绍	87
3.6	protobuf 序列化框架介绍	88
3.7	protostuff 序列化框架介绍	93
3.8	Thrift 序列化框架介绍	98
3.9	Avro 序列化框架介绍	100
3.9.1	Avro 介绍	100
3.9.2	Avro IDL 语言介绍	101

3.9.3	Schema 定义介绍	103
3.9.4	Maven 配置及使用 IDL 与 Schema 自动生成代码	103
3.9.5	Avro 序列化/反序列化实现	105
3.10	JBoss Marshalling 序列化框架介绍	110
3.11	序列化框架的选型	112
3.12	实现自己的序列化工具引擎	113
3.13	本章小结	118
第 4 章	实现分布式服务框架服务的发布与引入	119
4.1	Spring Framework 框架概述	119
4.1.1	Spring Framework 介绍	119
4.1.2	Spring Framework 周边生态项目介绍	121
4.2	FactoryBean 的秘密	122
4.2.1	FactoryBean 的作用及使用场景	123
4.2.2	FactoryBean 实现原理及示例说明	124
4.3	Spring 框架对于已有 RPC 框架集成的支持	127
4.3.1	Spring 支持集成 RPC 框架介绍	127
4.3.2	基于 RmiProxyFactoryBean 实现 RMI 与 Spring 的集成	128
4.3.3	基于 HttpInvokerProxyFactoryBean 实现 HTTP Invoker 与 Spring 的集成	131
4.3.4	基于 HessianProxyFactoryBean 实现 Hessian 与 Spring 的集成	133
4.4	实现自定义服务框架与 Spring 的集成	136
4.4.1	实现远程服务的发布	136
4.4.2	实现远程服务的引入	144
4.5	在 Spring 中定制自己的 XML 标签	150
4.6	本章小结	158
第 5 章	分布式服务框架注册中心	159
5.1	服务注册中心介绍	159
5.2	ZooKeeper 实现服务的注册中心原理	161

5.2.1	ZooKeeper 介绍	161
5.2.2	部署 ZooKeeper	161
5.2.3	ZkClient 使用介绍	164
5.2.4	ZooKeeper 实现服务注册中心	173
5.3	集成 ZooKeeper 实现自己的服务注册与发现	175
5.3.1	服务注册中心服务提供方	175
5.3.2	服务注册中心服务消费方	176
5.3.3	服务注册中心实现	178
5.4	本章小结	189
第 6 章	分布式服务框架底层通信实现	190
6.1	Java I/O 模型及 I/O 类库的进化	190
6.1.1	Linux 下实现的 I/O 模型	190
6.1.2	Java 语言实现的 I/O 模型	194
6.1.3	Java Classic I/O (Blocking I/O) 介绍	194
6.1.4	Java Non-blocking I/O (NIO) 介绍	211
6.1.5	NIO2 及 Asynchronous I/O 介绍	233
6.2	Netty 使用介绍	255
6.2.1	Netty 开发入门	256
6.2.2	Netty 粘包/半包问题解决	265
6.3	使用 Netty 构建服务框架底层通信	320
6.3.1	构建分布式服务框架 Netty 服务端	320
6.3.2	构建分布式服务框架服务调用端 Netty 客户端	330
6.4	本章小结	347
第 7 章	分布式服务框架软负载实现	348
7.1	软负载的实现原理	348
7.2	负载均衡常用算法	349
7.2.1	软负载随机算法实现	349

7.2.2	软负载加权随机算法实现	350
7.2.3	软负载轮询算法实现	351
7.2.4	软负载加权轮询算法实现	352
7.2.5	软负载源地址 hash 算法实现	354
7.3	实现自己的软负载机制	355
7.4	软负载在分布式服务框架中的应用	357
7.5	本章小结	361
第 8 章	分布式服务框架服务治理	362
8.1	服务治理介绍	362
8.2	服务治理的简单实现	364
8.2.1	服务分组路由实现	364
8.2.2	简单服务依赖关系分析实现	374
8.2.3	服务调用链路跟踪实现原理	380
8.3	本章小结	380
附录 A	如何配置运行本书完成的分布式服务框架	381

第 1 章

常用的 RPC 框架

1.1 RPC 框架原理

RPC (Remote Procedure Call, 远程过程调用) 一般用来实现部署在不同机器上的系统之间的方法调用, 使得程序能够像访问本地系统资源一样, 通过网络传输去访问远端系统资源。RPC 框架实现的架构原理都是类似的, 如图 1-1 所示。

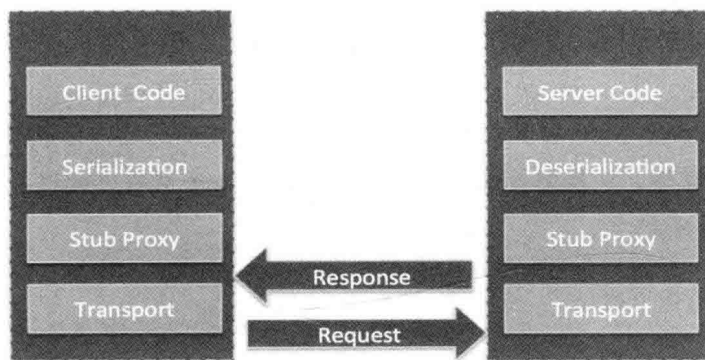


图 1-1 RPC 框架原理