

介绍分布式系统基础理论，分析分布式系统中常用的主流技术，
分享实战案例，做到理论与实践相结合。

Broadview[®]
www.broadview.com.cn



分布式系统

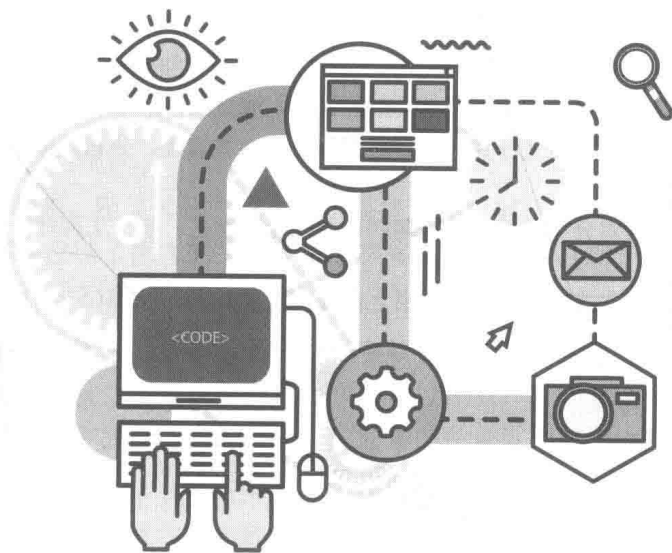
常用技术及案例分析

(第2版)

柳伟卫◎编著

 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



分布式系统 常用技术及案例分析

(第2版)

柳伟卫◎编著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书全面介绍在设计分布式系统时所要考虑的技术方案，内容丰富、案例新颖，相关理论与技术实践前瞻性较强。本书不仅介绍分布式系统的原理、基础理论，同时引入大量市面上常用的最新分布式系统技术，不仅告诉读者怎么用，同时分析为什么这么用，并阐述这些技术的优缺点。

本书分为两部分，即分布式系统基础理论、分布式系统常用技术。第一部分主要介绍分布式系统基础理论知识，总结一些在设计分布式系统时需要考虑的范式、知识点，以及可能会面临的问题，包括线程、通信、一致性、容错性、CAP 理论、安全性和并发等相关内容；同时讲述分布式系统的常见架构体系。第二部分主要列举了在分布式系统应用中经常采用的一些主流技术，并介绍这些技术的作用和用法，这些技术涵盖了分布式消息服务、分布式计算、分布式存储、分布式监控、分布式版本控制系统、RESTful、微服务、容器等。

本书主要面向的读者是对分布式系统感兴趣的计算机专业学生、软件工程师、系统架构师等。

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

分布式系统常用技术及案例分析 / 柳伟卫编著. —2 版. —北京：电子工业出版社，2019.1
ISBN 978-7-121-35677-3

I. ①分… II. ①柳… III. ①分布式操作系统—研究 IV. ①TP316.4

中国版本图书馆 CIP 数据核字（2018）第 271143 号

责任编辑：陈晓猛

印 刷：北京京科印刷有限公司

装 订：北京京科印刷有限公司

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：34.75

字数：667.2 千字

版 次：2017 年 2 月第 1 版

2019 年 1 月第 2 版

印 次：2019 年 1 月第 1 次印刷

定 价：99.00 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。

再 版 序

时光荏苒，岁月匆匆，距离《分布式系统常用技术及案例分析》第1版出版已经一载有余。热心的读者对于本书也投以了极大的关注，提了很多中肯的建议。对于这些建议，不管褒贬，一并全收，于是才有了第2版的出版。

对于技术型书籍的创作，笔者倾向于采用当今软件开发主流的方式——敏捷。敏捷写作打通了编写、校稿、出版、发行的整个流程，让知识可以在第一时间呈现给读者。读者在阅读本书之后，也可以及时对书中的内容进行反馈，从而帮助作者完善书中内容，最终形成良好的反馈闭环。第2版所更新的内容，希望正是读者所期待的。

第2版修改篇幅较大，修改内容大致包括以下几个方面：

- (1) 删除软件安装等比较简单的内容。
- (2) 每章的开头新增“概述”，让各个章节的技术点可以关联起来。
- (3) 每章增加“实战”案例，让技术点更具可操作性。
- (4) 修改第1版中的措辞、插图。

完整的修改内容，读者可以扫描封面上的二维码，参阅本书的在线文档“本书第1版与第2版的差异对比”。

柳伟卫

2018年5月22日于深圳

前 言

写作背景

我一直想写一本关于分布式系统的书。一方面想把个人工作中涉及的分布式技术做一下总结，另一方面想把个人多年的经验分享给广大的读者朋友。由于我的开发工作大都以 Java 为主，所以一开始设想的主题是“分布式 Java”，书也以开源方式发布在互联网上（网址为 <https://github.com/waylau/distributed-java>）。

后来，陈晓猛编辑看到了这本开源书，以及我关于分布式系统方面的博文，问我是否有兴趣出版分布式相关题材的图书。当然，书的内容不仅仅是“分布式 Java”。

对于出书一事，我犹豫良久。首先，本身工作挺忙，实在无暇顾及其他；其次，虽然我之前写过超过一打的书籍（<https://waylau.com/books/>），但多是开源电子书，时间、内容方面自然不会有太多约束，几乎是“想写就写，没有时间就不写”，这个跟正式出版还是存在比较大的差异的；最后，这本书涉及面相对较广，需要查阅大量资料，实在是太耗费精力。

但陈晓猛编辑还是鼓励我去做这个事情。思索再三，最终我答应了。当然，最后这本书还是在规定时间内完成了。它几乎耗尽了我写作期间所有的业余和休息时间。

“不积跬步，无以至千里；不积小流，无以成江海。”虽然整本书从构思到编写完成的时间不足一年，但书中的大部分知识点，都是我在多年的学习、工作中积累下来的。之所以能够实现快速写作，一方面是做了比较严格的时间管理，另一方面得益于我多年坚持写博客和开源书的习惯。

内容介绍

本书为两部分，即分布式系统基础理论和分布式系统常用技术。第一部分为第 1 章和第 2 章，主要介绍分布式系统基础理论知识，总结一些在设计分布式系统时需要考虑的范式、知识点及可能会面临的问题。第二部分为第 3 章到第 8 章，主要列举了在分布式系统应用中的一些主流技术，并介绍这些技术的作用和用法。

第 1 章介绍分布式系统基础理论知识，总结一些在设计分布式系统时需要考虑的范式、知识点及可能会面临的问题，包括线程、通信、一致性、容错性、CAP 理论、安全性和并发等相关内容。

第 2 章详细介绍分布式系统的架构体系，包括传统的基于对象的体系结构、SOA。

第 3 章介绍常用的分布式消息服务框架，包括 Apache ActiveMQ、Apache RabbitMQ、Apache RocketMQ、Apache Kafka 等。

第 4 章介绍分布式计算理论和应用框架方面的内容，包括 MapReduce、Apache Hadoop、Apache Spark、Apache Mesos 等。

第 5 章介绍分布式存储理论和应用框架方面的内容，包括 Bigtable、Apache HBase、Apache Cassandra、Memcached、Redis、MongoDB 等。

第 6 章介绍分布式监控方面常用的技术，包括 Nagios、Zabbix、Consul、ZooKeeper 等。

第 7 章介绍常用的分布式版本控制工具，包括 Bazaar、Mercurial、Git 等。

第 8 章介绍 RESTful API、微服务及容器相关的技术，着重介绍 Jersey、Spring Boot、Docker 等技术的应用。

源代码

本书提供源代码下载，下载地址为 <https://github.com/waylau/distributed-systems-technologies-and-cases-analysis>。

勘误和交流

本书如有勘误，会在 <https://github.com/waylau/distributed-systems-technologies-and-cases-analysis> 上发布。由于笔者能力有限，时间仓促，书中难免有错漏，欢迎读者批评指正。读者也可以到博文视点官网的本书页面进行交流（www.broadview.com.cn/00000）。

您也可以直接联系我：

博客：<https://waylau.com>

邮箱：waylau521@gmail.com

微博：<http://weibo.com/waylau521>

GitHub：<https://github.com/waylau>

致谢

首先，感谢电子工业出版社博文视点公司的陈晓猛编辑，是您鼓励我将本书付诸成册，并在我写作过程中审阅了大量稿件，给予我很多指导和帮助。感谢工作在幕后的电子工业出版社评审团队对于本书在校对、排版、审核、封面设计、错误改进方面所给予的帮助，使本书得以顺利出版发行。

其次，感谢在我十几年求学生涯中教育过我的所有老师，是你们将知识和学习方法传递给了我。感谢我曾经工作过的公司和单位，感谢和我一起共事过的同事和战友，你们的优秀一直是我追逐的目标，你们所给予的压力正是我不断改进的动力。

感谢我的父母、妻子 Funny 和两个女儿。由于撰写本书，我牺牲了很多陪伴家人的时间。感谢你们对于我工作的理解和支持。

最后，特别要感谢这个时代，互联网让所有人可以公平地享受这个时代的成果。感谢那些为计算机、互联网做出贡献的先驱，是你们让我可以站在更高的“肩膀”上！感谢那些为本书提供灵感的佳作，包括《分布式系统原理与范式》、*UNIX Network Programming*、*Enterprise SOA*、*MapReduce Design Patterns*、*Hadoop: The Definitive Guide Learning Hbase*、*Advanced Analytics with Spark*、*Pro Git*、*Docker in Action*、《淘宝技术这十年》、*Hatching Twitter*，等等，详细的书单可以参阅本书在线资源中的“参考文献”部分。

柳伟卫

目 录

第 1 章 分布式系统基础知识.....	1
1.1 概述.....	2
1.1.1 什么是分布式系统.....	2
1.1.2 集中式系统与分布式系统.....	2
1.1.3 如何设计分布式系统.....	4
1.1.4 分布式系统所面临的挑战.....	4
1.2 线程.....	5
1.2.1 什么是线程.....	5
1.2.2 进程和线程.....	6
1.2.3 线程和纤程.....	7
1.2.4 编程语言中的线程对象.....	7
1.2.5 SimpleThreads 示例.....	11
1.3 通信.....	13
1.3.1 网络 I/O 模型的演进.....	13
1.3.2 远程过程调用 (RPC).....	28
1.3.3 面向消息的通信.....	35
1.4 一致性.....	38
1.4.1 以数据为中心的一致性模型.....	38
1.4.2 以客户为中心的一致性.....	39
1.5 容错性.....	40
1.5.1 基本概念.....	41
1.5.2 故障分类.....	41
1.5.3 使用冗余来掩盖故障.....	42
1.5.4 分布式提交.....	42

1.6	CAP 理论	46
1.6.1	什么是 CAP 理论	47
1.6.2	为什么 CAP 只能三选二	48
1.6.3	CAP 常见模型	49
1.6.4	CAP 的意义	50
1.6.5	CAP 最新发展	50
1.7	安全性	51
1.7.1	基本概念	52
1.7.2	加密算法	54
1.7.3	安全通道	57
1.7.4	访问控制	66
1.8	并发	68
1.8.1	线程与并发	69
1.8.2	并发与并行	69
1.8.3	并发带来的风险	70
1.8.4	同步 (Synchronization)	72
1.8.5	原子访问 (Atomic Access)	77
1.8.6	无锁化设计提升并发能力	78
1.8.7	缓存提升并发能力	78
1.8.8	更细颗粒度的并发单元	79
第 2 章	分布式系统架构体系	80
2.1	基于对象的体系结构	81
2.1.1	分布式对象	81
2.1.2	Java RMI	82
2.2	面向服务的架构 (SOA)	85
2.2.1	SOA 的基本概念	86
2.2.2	基于 Web Services 的 SOA	88
2.2.3	SOA 的演变	103
2.3	REST 风格的架构	103
2.3.1	什么是 REST	103
2.3.2	-REST 有哪些特征	104

2.3.3	Java 实现 REST 的例子.....	106
2.3.4	REST API 最佳实践	116
2.4	微服务架构 (MSA)	119
2.4.1	什么是 MSA.....	119
2.4.2	MSA 与 SOA.....	121
2.4.3	何时采用 MSA.....	124
2.4.4	如何构建微服务	125
2.5	容器技术	129
2.5.1	虚拟化技术	129
2.5.2	容器与虚拟机	130
2.5.3	基于容器的持续部署	132
2.6	Serverless 架构.....	140
2.6.1	什么是 Serverless 架构.....	141
2.6.2	Serverless 典型的应用场景	142
2.6.3	Serverless 架构原则.....	144
2.6.4	例子: 使用 Serverless 实现游戏全球同服	146
第 3 章	分布式消息服务.....	152
3.1	分布式消息概述	153
3.1.1	基本概念	153
3.1.2	使用场景	153
3.1.3	常用技术	154
3.2	Apache ActiveMQ	154
3.2.1	例子: producer-consumer	154
3.2.2	例子: 使用 JMX 来监控 ActiveMQ.....	155
3.2.3	例子: 使用 Java 实现 producer-consumer.....	157
3.3	RabbitMQ	162
3.3.1	例子: Work Queues.....	162
3.3.2	例子: Publish/Subscribe.....	168
3.3.3	例子: Routing	172
3.3.4	例子: Topics.....	176
3.3.5	例子: RPC.....	181

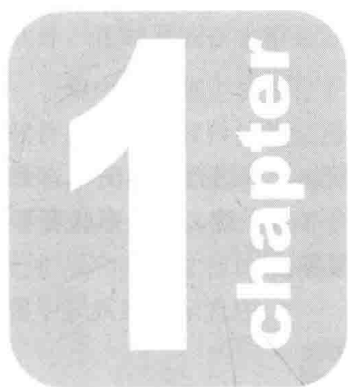
3.4	Apache RocketMQ.....	186
3.4.1	例子: 使用 Java 实现 producer-consumer.....	189
3.4.2	RocketMQ 最佳实践.....	193
3.5	Apache Kafka	198
3.5.1	Apache Kafka 的核心概念	199
3.5.2	Apache Kafka 的使用场景	202
3.6	实战: 基于 JMS 的消息发送和接收	203
3.6.1	项目概述	203
3.6.2	项目配置	205
3.6.3	编码实现	209
3.6.4	运行	215
第 4 章	分布式计算	221
4.1	分布式计算概述	222
4.1.1	使用场景	222
4.1.2	常用技术	222
4.2	MapReduce.....	223
4.2.1	MapReduce 简介	223
4.2.2	MapReduce 的编程模型	223
4.2.3	MapReduce 接口实现	228
4.2.4	MapReduce 的使用技巧	234
4.3	Apache Hadoop.....	236
4.3.1	Apache Hadoop 的核心组件.....	237
4.3.2	例子: 词频统计 WordCount 程序	238
4.4	Spark.....	240
4.4.1	Spark 简介	240
4.4.2	Spark 与 Hadoop 的关系	241
4.4.3	Spark 2.0 的新特性	242
4.4.4	Spark 集群模式.....	246
4.5	Mesos.....	248
4.5.1	Mesos 简介	249
4.5.2	设计高可用的 Mesos framework.....	250

4.6 实战：基于 Spark 的词频统计	257
4.6.1 项目概述	257
4.6.2 项目配置	257
4.6.3 编码实现	258
4.6.4 运行	259
第 5 章 分布式存储	262
5.1 分布式存储概述	263
5.1.1 使用场景	263
5.1.2 常用技术	263
5.2 Bigtable	264
5.2.1 Bigtable 的数据模型	264
5.2.2 Bigtable 的实现	266
5.2.3 Bigtable 的性能优化	270
5.3 Apache HBase	273
5.3.1 Apache HBase 的基本概念	274
5.3.2 Apache HBase 的架构	281
5.4 Apache Cassandra	296
5.4.1 Apache Cassandra 简介	296
5.4.2 Apache Cassandra 的应用场景	299
5.4.3 Apache Cassandra 的架构和数据模型	300
5.4.4 用于配置 Apache Cassandra 的核心组件	301
5.5 Memcached	302
5.5.1 Memcached 简介	303
5.5.2 Memcached 的架构	303
5.5.3 Memcached 客户端	305
5.6 Redis	313
5.6.1 Redis 简介	313
5.6.2 Redis 的下载与简单使用	314
5.6.3 Redis 的数据类型及抽象	314
5.7 MongoDB	334
5.7.1 MongoDB 简介	334

5.7.2	MongoDB 核心概念	335
5.7.3	MongoDB 的数据模型	340
5.7.4	示例: Java 连接 MongoDB	354
5.8	实战: 基于 Redis 的分布式锁	355
5.8.1	项目概述	355
5.8.2	项目配置	356
5.8.3	编码实现	357
5.8.4	运行	360
第 6 章	分布式监控	364
6.1	分布式监控概述	365
6.1.1	使用场景	365
6.1.2	常用技术	365
6.2	Nagios	365
6.2.1	Nagios 监控	366
6.2.2	Nagios 插件	384
6.3	Zabbix	386
6.3.1	Zabbix 对容器的支持	386
6.3.2	Zabbix 的基本概念	389
6.4	Consul	399
6.4.1	Consul 架构	400
6.4.2	Consul agent	401
6.5	ZooKeeper	411
6.5.1	ZooKeeper 简介	411
6.5.2	ZooKeeper 内部工作原理	415
6.5.3	例子: ZooKeeper 实现 barrier 和 producer-consumer queue	419
6.6	实战: 基于 ZooKeeper 的服务注册和发现	426
6.6.1	项目概述	426
6.6.2	项目配置	427
6.6.3	编码实现	428
6.6.4	运行	433

第 7 章 分布式版本控制系统.....	435
7.1 分布式版本控制系统概述.....	436
7.1.1 集中式与分布式.....	436
7.1.2 分布式版本控制系统的核心概念.....	437
7.2 Bazaar.....	437
7.2.1 Bazaar 的核心概念.....	437
7.2.2 Bazaar 的使用.....	438
7.3 Mercurial.....	443
7.3.1 Mercurial 的核心概念.....	444
7.3.2 Mercurial 的使用.....	447
7.4 Git.....	454
7.4.1 Git 的基础概念.....	454
7.4.2 Git 的使用.....	457
7.5 Git Flow——团队协作最佳实践.....	483
7.5.1 分支定义.....	483
7.5.2 新功能开发工作流.....	484
7.5.3 Bug 修复工作流.....	485
7.5.4 版本发布工作流.....	485
第 8 章 RESTful API、微服务及容器技术.....	487
8.1 Jersey.....	488
8.1.1 Jersey 简介.....	488
8.1.2 Jersey 的模块和依赖.....	488
8.1.3 JAX-RS 核心概念.....	492
8.1.4 例子：用 SSE 构建实时 Web 应用.....	503
8.2 Spring Boot.....	511
8.2.1 Spring Boot 简介.....	512
8.2.2 Spring Boot 的安装.....	513
8.2.3 Spring Boot 的使用.....	518
8.2.4 Spring Boot 的属性与配置.....	524
8.3 Docker.....	529
8.3.1 Docker 简介.....	529

8.3.2	Docker 的核心组成、架构及工作原理	529
8.3.3	Docker 的使用	535
8.4	实战：基于 Docker 构建、运行、发布微服务	537
8.4.1	编写微服务	537
8.4.2	微服务容器化	538
8.4.3	构建 Docker image	538
8.4.4	运行 image	540
8.4.5	访问应用	541
8.4.6	发布微服务	541



第 1 章

分布式系统基础知识

1.1 概述

毫无疑问，计算机改变了人类的工作和生活方式，而计算机系统也正在进行一场变革。无论是手机应用，还是智能终端，都离不开背后那个神秘的巨人——分布式系统。正是那些看不见的分布式系统，每天处理着数以亿计的计算，提供可靠而稳定的服务。

本章就揭开分布式系统的神秘面纱。

1.1.1 什么是分布式系统

《分布式系统原理与范型》一书中是这样定义分布式系统的：

“分布式系统是若干独立计算机的集合，这些计算机对于用户来说就像单个系统。”

这里面包含了两个要点：

- 硬件独立；
- 软件统一。

什么是硬件独立？所谓硬件独立，是指机器本身是独立的。一个大型的分布式系统，由若干计算机组成系统的基础设施。而软件统一，是指对于用户来说，用户就像跟单个系统打交道。就好比我们每天上网看视频，视频网站对我们来说就是一个系统软件，它们背后是如何运作的、部署了几台服务器、每台服务器是干什么的，这些对用户来说是不可见的。用户不关心背后的这些服务器，用户所关心的是，今天能看什么样的节目、视频运行是否流畅、清晰度如何等。

软件统一的另外一个方面是指，分布式系统的扩展和升级都比较容易。分布式系统中的某些节点发生故障，不会影响整体系统的可用性。用户和应用程序交互时，不会察觉哪些部分正在替换或维修，也不会感知新加入的部分。

1.1.2 集中式系统与分布式系统

集中式系统主要部署在 HP、IBM、Sun 等小型机以上档次的服务器中，把所有的功能都集成到主服务器上（这对服务器的要求很高）。它们的主要特色在于宕机时间只有几小时，所以又统称为 z 系列（zero，零）。AS/400 主要应用在银行和制造业，还用于 Domino，主要的技术包括 TIMI（技术独立机器界面）和单级存储。有了 TIMI，可以实现硬件与软件相互独立。RS/6000 比较常见，用于科学计算和事务处理等。这类主机的单机性能一般都很强，带多个终端。终端没有数据处理能力，运算全部在主机上进行。现在的银行系统大部分都是集中式系统。