

仅限非商业用途或学习研究使用

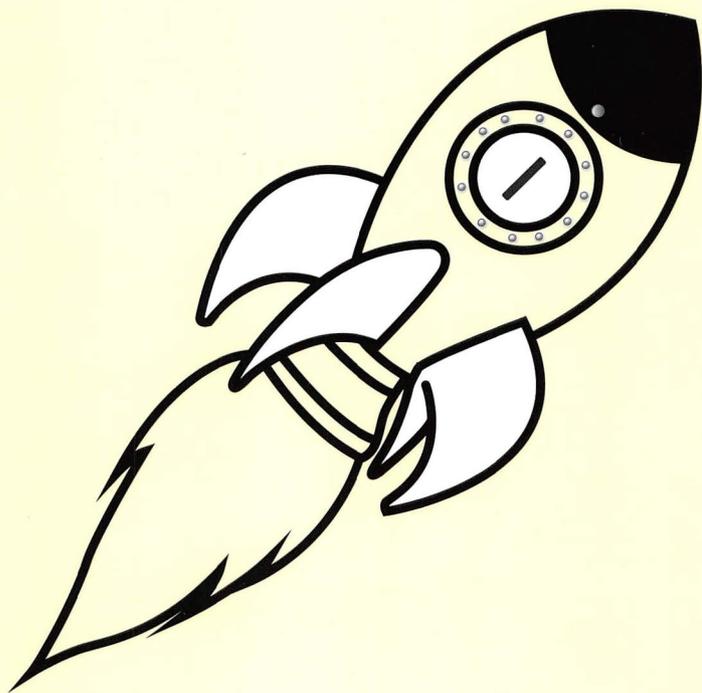
HZ BOOKS
华章IT

云栖社区 系列丛书

RocketMQ

实战与原理解析

杨开元◎著



APACHE ROCKETMQ
PRINCIPLE AND PRACTICE



机械工业出版社
China Machine Press

阿里巴巴数据专家RocketMQ源码贡献者撰写，RocketMQ官方开发团队鼎力推荐

内容简介

本书由云栖社区官方出品。

作者是阿里资深数据专家，对RocketMQ有深入的研究，并有大量的实践经验。在写这本书之前，作者不仅系统、深入地阅读了RocketMQ的源代码，而且还向RocketMQ的官方开发团队深入了解了它的诸多设计细节。作者结合自己多年使用RocketMQ的经验，从开发和运维两个维度，给出了大部分场景下的优秀实践，能帮助读者在学会使用和用好RocketMQ的同时，尽量少“踩坑”。同时，本书也结合源码分析了分布式消息队列的原理，使读者可以在复杂业务场景下定制有特殊功能的消息队列。

全书共13章，在逻辑上分为两大部分。

第一部分（第1~8章）：RocketMQ实战

第1~2章详细讲解了RocketMQ如何快速入门，以及在生产环境下的配置和使用；

第3~4章具体讲解了不同类型生产者和消费者的特点，以及分布式消息队列的协调者NameServer；

第5章从消息的存储、发送、复制和高可用等多个维度讲解了RocketMQ的内部机制；

第6章讨论了消息的可靠性，以及如何让消息队列在满足业务逻辑需求的同时稳定、可靠地长期运行；

第7章讨论了在大流量场景下，吞吐量优先时RocketMQ的使用方法；

第8章介绍了RocketMQ与SpringBoot、Spark、Flink以及自定义的运维工具等其他系统的对接方法；

第二部分（第9~13章）：RocketMQ原理

首先对RocketMQ的源码结构进行了整体介绍，然后深入分析了NameServer、各种常用消费类、主从同步机制，以及基于Netty的通信的源码实现。掌握这些源代码以后，读者可以快速定制属于自己的具有特殊功能的消息中间件。

华章 IT
HZBOOKS | Information Technology



APACHE ROCKETMQ
PRINCIPLE AND PRACTICE

RocketMQ

实战与原理解析

杨开元◎著



机械工业出版社
China Machine Press



图书在版编目 (CIP) 数据

RocketMQ 实战与原理解析 / 杨开元著. —北京: 机械工业出版社, 2018.5
(云栖社区系列)

ISBN 978-7-111-60025-1

I. R… II. 杨… III. 计算机网络—软件工具 IV. TP393.07

中国版本图书馆 CIP 数据核字 (2018) 第 109610 号

RocketMQ 实战与原理解析

出版发行: 机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码: 100037)

责任编辑: 张锡鹏

责任校对: 殷虹

印刷: 北京市兆成印刷有限责任公司

版次: 2018 年 6 月第 1 版第 1 次印刷

开本: 186mm×240mm 1/16

印张: 10.5

书号: ISBN 978-7-111-60025-1

定价: 59.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88379426 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

读者信箱: hzit@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京大成律师事务所 韩光 / 邹晓东



Foreword 推荐序

在阿里巴巴技术发展初期，伴随着淘宝业务的快速发展，网站流量呈现几何级增长。单体巨无霸式的应用无法处理爆发式增长的流量，阿里内部从业务、组织层面进行了一次大的水平与垂直切分，拆分出用户中心、商品中心、交易中心、评价中心等平台型应用，分布式电商系统的雏形由此诞生。阿里的消息引擎就是在这样的大背景下诞生的，并被应用于各个应用系统之间的异步解耦和削峰填谷。

从最初的日志传输领域到后来阿里集团全维度在线业务的支撑，RocketMQ 被广泛用于交易、数据同步、缓存同步、IM 通讯、流计算、IoT 等场景。在近几年的双 11 全球狂欢节中，RocketMQ 以万亿级的消息总量支撑了全集团 3000 多个应用，为复杂的业务场景提供了系统解耦、削峰填谷的能力，保障了核心交易链路消息流转的低延迟、高吞吐，为阿里集团大中台的稳定性发挥了举足轻重的作用。

为了更好地发展 RocketMQ 社区生态，2016 年双 11 前后，阿里巴巴将 RocketMQ 捐赠给 Apache 基金会，吸引了全球的开源爱好者参与到 RocketMQ 社区中，并于 2017 年 9 月成为 Apache 基金会的顶级项目。在开源社区的帮助下，RocketMQ 具备了对接主流大数据流计算平台、离在线数据处理以及对接存储平台的能力。



本书介绍了分布式消息中间件 RocketMQ 的方方面面，作者为大数据领域的技术专家，在分布式领域具有很丰富的理论积累和实战经验。书如其人，书中各章节尽展实战经验，庖丁解牛般剖析了 Apache RocketMQ 的原理和架构设计。本书深入浅出地分析了 RocketMQ 的整体架构，分享了部署和运维的经验，涵盖 RocketMQ 的核心特性——高可用、高可靠机制，以及开源生态等。

本书作为国内首本全面解析 Apache RocketMQ 的书籍，对于希望了解 RocketMQ 技术内幕，以及想要掌握分布式系统设计理念的技术人员来说的确不容错过。

——周新宇，Apache RocketMQ 项目管理委员会成员



Preface 前言

为什么要写这本书

几年前在做一个项目的时候，若需要用到消息队列，简单调研一下就会决定用 Kafka，因为当时还不知道有 RocketMQ。在我加入阿里后，当时有个项目需要用到消息中间件，试用了 RocketMQ，发现阿里开源的消息中间件性能非常强大，但是上手有点费劲，因为现有文档多是零零散散的博文。在没有合适文档指导的情况下，对系统中用到的 RocketMQ 模块心里没底，系统偶尔出现异常时总会束手无策，需要通过查看很多源码，才能保证系统的稳定运行。

熟悉 RocketMQ 以后，我发现它是一款非常优秀的中间件产品，可以确保不丢消息，而且效率很高。同时因为它用 Java 开发的，所以修改起来比较容易。在阿里内部，RocketMQ 很好地服务了集团大小上千个应用，在每年的双十一当天，更有不可思议的万亿级消息通过 RocketMQ 流转（在 2017 年的双 11 当天，整个阿里巴巴集团通过 RocketMQ 流转的线上消息达到了万亿级，峰值 TPS 达到 5600 万），在阿里大中台策略上发挥着举足轻重的作用。所以如果有合适的参考文档，RocketMQ 会被更多人接受和使用，让更多人不必重复造“轮子”。

我做了很多年开发，在学校课本上学的开发知识有限，大多数是通过看书和上网学到的，其中很多优秀的文章对自己帮助很大。所以我很希望能用这本书回馈技术社区中有需要的开发者们。



动笔写这本书前，我系统地阅读了 RocketMQ 的源码，有些理解不够透彻的地方请教了阿里 RocketMQ 开发团队的同事，然后也总结了自己多年实际工作中的一些经验。希望这本书能简明扼要地说清楚 RocketMQ 的使用方法和核心原理。

读者对象

- 希望学习分布式系统或分布式消息队列的开发人员。
- 服务端系统开发者，他们可以借助高质量中间件来提高开发效率。
- 软件架构师，他们可以通过消息队列优化复杂系统的设计。

本书特色

本书系统地介绍了 RocketMQ 这款优秀的分布式消息队列软件，通过阅读本书，读者可以快速把 RocketMQ 应用到自己的项目中，也可以通过更改源码定制符合自身业务的消息中间件。

如何阅读本书

本书分为两大部分：

第一部分是 RocketMQ 实战，包括第 1~8 章。这是本书的主体内容，可帮助读者快速用好 RocketMQ 这个分布式消息队列。

这部分是按照由浅入深的方式撰写的，为了让读者快速上手，首先介绍了搭建一个简单 RocketMQ 集群的方法，以此来发送和接收消息；然后详细介绍了如何用好 Consumer 和 Producer，如何选择合适的类以及进行参数设置；再进一步根据应用，说明如何让 RocketMQ 在各种异常情况下保持稳定可靠，以及如何增大 RocketMQ 的吞吐量，从而在单位时间内处理更多的消息。



第二部分是源码分析，包括第 9~13 章。当读者有特殊的业务需求，需要更改或扩展 RocketMQ 现有功能的时候，这部分内容能帮助读者快速熟悉源码，找到要下手更改的地方，快速实现想要的功能。

这部分也适合想通过源码，深入学习消息队列的读者阅读。学习别人优秀的代码是提升自己技术水平的一条有效途径。

勘误和支持

由于水平有限，编写时间仓促，书中难免会出现一些错误或者不准确的地方，恳请读者批评指正。有任何的意见或建议，都可以通过邮箱 rocketmqqa@163.com 和我联系，真挚期待你的反馈。

致谢

写技术书籍很耗费时间，加之互联网行业快节奏的工作方式，导致我写这本书的时间大多是在周末和夜晚。在此感谢家人对我的支持和理解，尤其感谢我的妻子，没有她对家庭的照顾和对我的鼓励，这本书是无法完成的。

感谢阿里消息中间件团队的 Leader 王小瑞，是你从技术和写作思路给我很大的帮助。感谢消息中间件团队的其他同学，你们为开源社区贡献了一个高质量的软件，你们写的很多高质量博文使开发者更容易理解 RocketMQ。

感谢机械工业出版社的编辑杨福川、张锡鹏，感谢云栖社区的刁云怡，阿里的校友耿嘉安，是你们始终支持我的写作，你们的引导和帮助使我能顺利完成全部书稿。

谨以本书献给我最亲爱的家人，以及众多热爱软件开发工作的朋友们！

杨开元



目 录 Contents

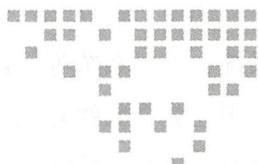
推荐序	2.2 多机集群配置和部署	9
前言	2.2.1 启动多个 NameServer 和 Broker	10
第 1 章 快速入门	2.2.2 配置参数介绍	11
1.1 消息队列功能介绍	2.3 发送 / 接收消息示例	13
1.1.1 应用解耦	2.4 常用管理命令	15
1.1.2 流量消峰	2.5 通过图形界面管理集群	21
1.1.3 消息分发	2.6 本章小结	22
1.2 RocketMQ 简介	第 3 章 用适合的方式发送和接收消息	23
1.3 快速上手 RocketMQ	3.1 不同类型的消费者	23
1.3.1 RocketMQ 的下载、安装和配置	3.1.1 DefaultMQPushConsumer 的使用	23
1.3.2 启动消息队列服务	3.1.2 DefaultMQPushConsumer 的处理流程	25
1.3.3 用命令行发送和接收消息	3.1.3 DefaultMQPushConsumer 的流量控制	28
1.3.4 关闭消息队列	3.1.4 DefaultMQPullConsumer	30
1.4 本章小结		
第 2 章 生产环境下的配置和使用		
2.1 RocketMQ 各部分角色介绍		



3.1.5 Consumer 的启动、关闭 流程	32	5.2 消息存储结构	58
3.2 不同类型的生产者	33	5.3 高可用性机制	60
3.2.1 DefaultMQProducer	34	5.4 同步刷盘和异步刷盘	61
3.2.2 发送延迟消息	36	5.5 同步复制和异步复制	62
3.2.3 自定义消息发送规则	36	5.6 本章小结	63
3.2.4 对事务的支持	37	第 6 章 可靠性优先的使用场景	64
3.3 如何存储队列位置信息	38	6.1 顺序消息	64
3.4 自定义日志输出	42	6.1.1 全局顺序消息	64
3.5 本章小结	44	6.1.2 部分顺序消息	65
第 4 章 分布式消息队列的 协调者	45	6.2 消息重复问题	67
4.1 NameServer 的功能	45	6.3 动态增减机器	67
4.1.1 集群状态的存储结构	46	6.3.1 动态增减 NameServer	67
4.1.2 状态维护逻辑	47	6.3.2 动态增减 Broker	69
4.2 各个角色间的交互流程	48	6.4 各种故障对消息的影响	70
4.2.1 交互流程源码分析	48	6.5 消息优先级	72
4.2.2 为何不用 ZooKeeper	50	6.6 本章小结	73
4.3 底层通信机制	50	第 7 章 吞吐量优先的使用场景	74
4.3.1 Remoting 模块	51	7.1 在 Broker 端进行消息过滤	74
4.3.2 协议设计和编解码	54	7.1.1 消息的 Tag 和 Key	74
4.3.3 Netty 库	56	7.1.2 通过 Tag 进行过滤	75
4.4 本章小结	56	7.1.3 用 SQL 表达式的方式进行 过滤	75
第 5 章 消息队列的核心机制	57	7.1.4 Filter Server 方式过滤	77
5.1 消息存储和发送	57	7.2 提高 Consumer 处理能力	78
		7.3 Consumer 的负载均衡	80

7.3.1 DefaultMQPushConsumer 的 负载均衡	80	9.3 源码结构	99
7.3.2 DefaultMQPullConsumer 的 负载均衡	81	9.4 不断迭代的代码	100
7.4 提高 Producer 的发送速度	83	9.5 本章小结	102
7.5 系统性能调优的一般流程	85		
7.6 本章小结	87		
第 8 章 和其他系统交互	88	第 10 章 NameServer 源码	
8.1 在 SpringBoot 中使用 RocketMQ	88	解析	103
8.1.1 直接使用	88	10.1 模块入口代码的功能	103
8.1.2 通过 Spring Messaging 方式 使用	90	10.1.1 入口函数	103
8.2 直接使用云上 RocketMQ	91	10.1.2 解析命令行参数	104
8.3 RocketMQ 与 Spark、Flink 对接	93	10.1.3 初始化 NameServer 的 Controller	105
8.4 自定义开发运维工具	93	10.2 NameServer 的总控逻辑	106
8.4.1 开源版本运维工具功能 介绍	94	10.3 核心业务逻辑处理	107
8.4.2 基于 Tools 模块开发自定义 运维工具	95	10.4 集群状态存储	109
8.5 本章小结	96	10.5 本章小结	111
第 9 章 首个 Apache 中间件		第 11 章 最常用的消费类	112
顶级项目	97	11.1 整体流程	112
9.1 RocketMQ 的前世今生	97	11.1.1 上层接口类	112
9.2 Apache 顶级项目 (TLP) 之路	98	11.1.2 DefaultMQPushConsumer 的实现者	114
		11.1.3 获取消息逻辑	116
		11.2 消息的并发处理	118
		11.2.1 并发处理过程	118
		11.2.2 ProcessQueue 对象	121
		11.3 生产者消费者的底层类	122
		11.3.1 MQClientInstance 类的 创建规则	122

11.3.2 MQClientInstance 类的 功能	124	13.2.2 统一的异步 I/O 接口	137
11.4 本章小结	127	13.2.3 基于拦截链模式的事件 模型	138
第 12 章 主从同步机制	128	13.2.4 高级组件	139
12.1 同步属性信息	128	13.3 Netty 用法示例	140
12.2 同步消息体	130	13.3.1 Discard 服务器	140
12.3 sync_master 和 async_master ..	132	13.3.2 查看收到的数据	144
12.4 本章小结	134	13.4 RocketMQ 基于 Netty 的通信 功能实现	145
第 13 章 基于 Netty 的通信实现 ..	135	13.4.1 顶层抽象类	145
13.1 Netty 介绍	135	13.4.2 自定义协议	148
13.2 Netty 架构总览	136	13.4.3 基于 Netty 的 Server 和 Client	151
13.2.1 重新实现 ByteBuffer	136	13.5 本章小结	152



快速入门

本章可以让读者了解 RocketMQ 和分布式消息队列的功能，然后搭建好单机版的消息队列，进而能够发送并接收简单的消息。

1.1 消息队列功能介绍

简单来说，消息队列就是基础数据结构课程里“先进先出”的一种数据结构，但是如果消除单点故障，保证消息传输的可靠性，并且还能应对大流量的冲击，对消息队列的要求就很高了。现在互联网“微架构”模式兴起，原有大型集中式的 IT 服务因为各种弊端，通常被分拆成细粒度的多个“微服务”，这些微服务可以在一个局域网内，也可能跨机房部署。一方面对服务之间松耦合的要求越来越高，另一方面，服务之间的联系却越来越紧密，对通信质量的要求也越来越高。分布式消息队列可以提供应用解耦、流量消峰、消息分发等功能，已经成为大型互联网服务架构里标配的中间件。

1.1.1 应用解耦

复杂的应用里会存在多个子系统，比如在电商应用中有订单系统、库存系

统、物流系统、支付系统等。这个时候如果各个子系统之间的耦合性太高，整体系统的可用性就会大幅降低。多个低错误率的子系统强耦合在一起，得到的是一个高错误率的整体系统。

以电商应用为例，用户创建订单后，如果耦合调用库存系统、物流系统、支付系统，任何一个子系统出了故障或者因为升级等原因暂时不可用，都会造成下单操作异常，影响用户使用体验。

如图 1-1 所示，当转变成基于消息队列的方式后，系统可用性就高多了，比如物流系统因为发生故障，需要几分钟的时间来修复，在这几分钟的时间里，物流系统要处理的内容被缓存在消息队列里，用户的下单操作可以正常完成。当物流系统恢复后，补充处理存储在消息队列里的订单信息即可，终端用户感知不到物流系统发生过几分钟的故障。

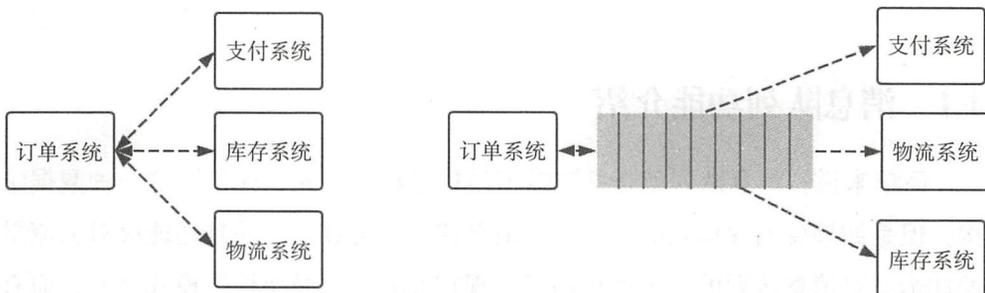


图 1-1 消息队列的解耦功能

1.1.2 流量消峰

每年的双十一，淘宝的很多活动都在 0 点的时候开启，大部分应用系统流量会在瞬间猛增，这个时候如果没有缓冲机制，不可能承受住短时大流量的冲击。通过利用消息队列，把大量的请求暂存起来，分散到相对长的一段时间内处理，能大大提高系统的稳定性和用户体验。

举个例子，如果订单系统每秒最多能处理一万次下单，这个处理能力应对