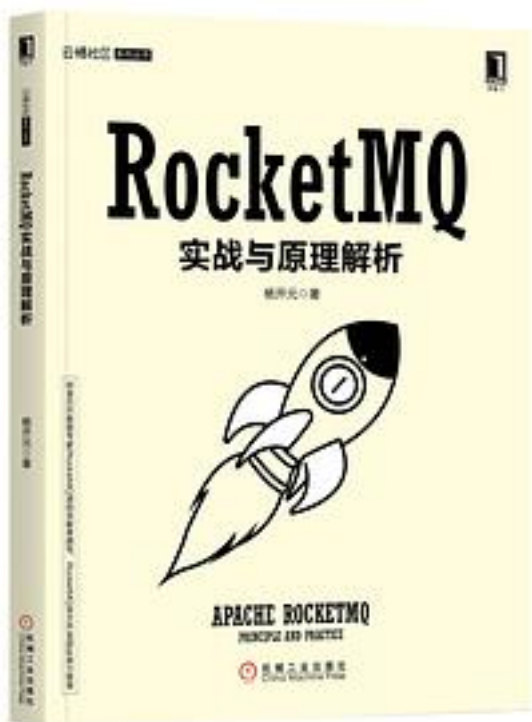


# RocketMQ实战与原理解析



[RocketMQ实战与原理解析\\_下载链接1](#)

著者:杨开元

出版者:机械工业出版社

出版时间:2018-6-30

装帧:平装

isbn:9787111600251

- RocketMQ由阿里开源，Apache开源项目，经受多年流量峰值考验，在多个性能指标上远超同类产品
- 作者是阿里资深数据专家，有多年RocketMQ使用经验，深入研究RocketMQ源代码，写作前与RocketMQ官方团队有深入沟通

• 云栖社区官方出品，得到RocketMQ官方研发团队以及业界的多位专家的肯定和推荐

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，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RocketMQ与SpringBoot、Spark、Flink以及自定义的运维工具等其它系统的对接方法；

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，然后深入地分析了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。

作者介绍:

杨开元

阿里巴巴数据专家，毕业于北京大学，有10年IT行业研发经验。对RocketMQ有深入的研究，是RocketMQ源码贡献者。曾就职于甲骨文和猎豹移动，专注于大数据和实时计算。在大量的工作实践中，对MySQL、J2EE、JVM、Spring、Hadoop、Kafka、Storm、Flink都有深入研究。喜欢剖析源码，分析原理，为开源项目贡献代码。

云栖社区

云栖社区是面向开发者的开放型技术平台，服务于云计算技术全生态。包含博客、问答、培训、设计研发、资源下载等产品，以分享专业、优质、高效的技术为己任，帮助技术人快速成长与发展。

欢迎关注云栖社区微信公众号：yunqiinsight，打开精彩代码生活！

## 目录: 推荐序

### 前言

## 第1章 快速入门1

### 1.1 消息队列功能介绍1

#### 1.1.1 应用解耦1

#### 1.1.2 流量消峰2

#### 1.1.3 消息分发3

### 1.2 RocketMQ简介4

### 1.3 快速上手RocketMQ4

#### 1.3.1 RocketMQ的下载、安装和配置 5

#### 1.3.2 启动消息队列服务6

#### 1.3.3 用命令行发送和接收消息6

#### 1.3.4 关闭消息队列6

### 1.4 本章小结7

## 第2章 生产环境下的配置和使用8

### 2.1 RocketMQ各部分角色介绍8

### 2.2 多机集群配置和部署9

#### 2.2.1 启动多个NameServer和Broker10

#### 2.2.2 配置参数介绍11

### 2.3 发送/接收消息示例13

### 2.4 常用管理命令15

### 2.5 通过图形界面管理集群21

### 2.6 本章小结22

## 第3章 用适合的方式发送和接收消息23

### 3.1 不同类型的消费者23

#### 3.1.1 DefaultMQPushConsumer的使用23

#### 3.1.2 DefaultMQPushConsumer的处理流程25

#### 3.1.3 DefaultMQPushConsumer的流量控制28

#### 3.1.4 DefaultMQPullConsumer30

#### 3.1.5 Consumer的启动、关闭流程32

### 3.2 不同类型的生产者33

#### 3.2.1 DefaultMQProducer 34

#### 3.2.2 发送延迟消息36

#### 3.2.3 自定义消息发送规则36

#### 3.2.4 对事务的支持37

### 3.3 如何存储队列位置信息38

### 3.4 自定义日志输出42

### 3.5 本章小结44

## 第4章 分布式消息队列的协调者45

### 4.1 NameServer的功能45

#### 4.1.1 集群状态的存储结构46

#### 4.1.2 状态维护逻辑47

### 4.2 各个角色间的交互流程48

#### 4.2.1 交互流程源码分析48

#### 4.2.2 为何不用ZooKeeper50

### 4.3 底层通信机制50

#### 4.3.1 Remoting模块51

#### 4.3.2 协议设计和编解码54

#### 4.3.3 Netty库56

### 4.4 本章小结56

## 第5章 消息队列的核心机制57

### 5.1 消息存储和发送57

### 5.2 消息存储结构58

5.3 高可用性机制	60
5.4 同步刷盘和异步刷盘	61
5.5 同步复制和异步复制	62
5.6 本章小结	63
第6章 可靠性优先的使用场景	64
6.1 顺序消息	64
6.1.1 全局顺序消息	64
6.1.2 部分顺序消息	65
6.2 消息重复问题	67
6.3 动态增减机器	67
6.3.1 动态增减NameServer	67
6.3.2 动态增减Broker	69
6.4 各种故障对消息的影响	70
6.5 消息优先级	72
6.6 本章小结	73
第7章 吞吐量优先的使用场景	74
7.1 在Broker端进行消息过滤	74
7.1.1 消息的Tag和Key	74
7.1.2 通过Tag进行过滤	75
7.1.3 用SQL表达式的方式进行过滤	75
7.1.4 Filter Server方式过滤	77
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
7.3.2 DefaultMQPullConsumer的负载均衡	81
7.4 提高Producer的发送速度	83
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
8.1 在SpringBoot中使用RocketMQ	88
8.1.1 直接使用	88
8.1.2 通过Spring Messaging方式使用	90
8.2 直接使用云上RocketMQ	91
8.3 RocketMQ与Spark、Flink对接	93
8.4 自定义开发运维工具	93
8.4.1 开源版本运维工具功能介绍	94
8.4.2 基于Tools模块开发自定义运维工具	95
8.5 本章小结	96
第9章 首个Apache中间件顶级项目	97
9.1 RocketMQ的前世今生	97
9.2 Apache顶级项目（TLP）之路	98
9.3 源码结构	99
9.4 不断迭代的代码	100
9.5 本章小结	102
第10章 NameServer源码解析	103
10.1 模块入口代码的功能	103
10.1.1 入口函数	103
10.1.2 解析命令行参数	104
10.1.3 初始化NameServer的Controller	105
10.2 NameServer的总控逻辑	106
10.3 核心业务逻辑处理	107
10.4 集群状态存储	109
10.5 本章小结	111
第11章 最常用的消费类	112

- 11.1 整体流程112
  - 11.1.1 上层接口类112
  - 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
- 11.4 本章小结127
- 第12章 主从同步机制128
  - 12.1 同步属性信息128
  - 12.2 同步消息体130
  - 12.3 sync\_master和async\_master132
  - 12.4 本章小结134
- 第13章 基于Netty的通信实现135
  - 13.1 Netty介绍135
  - 13.2 Netty架构总览136
    - 13.2.1 重新实现ByteBuffer136
    - 13.2.2 统一的异步 I/O接口137
    - 13.2.3 基于拦截链模式的事件模型138
    - 13.2.4 高级组件139
  - 13.3 Netty用法示例140
    - 13.3.1 Discard服务器140
    - 13.3.2 查看收到的数据144
  - 13.4 RocketMQ基于Netty的通信功能实现145
    - 13.4.1 顶层抽象类145
    - 13.4.2 自定义协议148
    - 13.4.3 基于Netty的Server和Client151
  - 13.5 本章小结152

• • • • • ([收起](#))

[RocketMQ实战与原理解析\\_下载链接1](#)

## 标签

消息队列

rocketmq

分布式

java

中间件

数据平台

mq

Java

## 评论

本书逻辑非常不清晰，想到哪写到哪那种。先介绍消费者再介绍生产者是哪样？很多细节都是蜻蜓点水，不建议阅读。

-----  
这本书更多说的是应用和基本原理，没有源码分析。讲的是原理分析不是源码分析。

-----  
本书介绍了RocketMQ的一些基本概念，并分析了源码。只是感觉东西有点偏少。

-----  
页数不多，内容还成

-----  
\* 使用轻量级的NameServer（多台），没有使用ZK（rocketmq没有选举） \* broker：master、slave模式（预先配置好的），master挂了不能写，还可以继续读（read可以从master和slave读） \* 与kafka还是有很多不同的设计的 \* 一共一个commitLog（尽量顺序写，可以支持多个topic），同时topic对应于多个MessageQueue \* 支持事物，两阶段提交 \* producer：localTransaction + localTransaction查询 \* broker：half queue + op queue

-----  
非常好的入门书籍。

-----  
退货

-----  
比较好的中间件

-----  
比官方文档强，入门足够了

-----  
适合使用入门和阅读代码入门..

-----  
有点过于简单。好多关键的东西都一笔带过了

-----  
…内容过于碎片化，不缜密，不系统，不全面也不够深刻，没写清楚

-----  
很适合入门并且对rocketmq有个粗浅的了解，关于代码的部分就显得有些没有诚意了，缺乏原理性的解读。

-----  
讲解使用。但很多地方的源码贴的没头没尾的。解说也不觉得很深入的地方。

-----  
对RocketMQ的脉络做了一个大概的说明吧，深入细节的东西还是需要自己看代码

-----  
内容简单；排版空隙巨大，有凑页数的嫌疑；代码大段，无诚意；自己的深入了解及原理讲解很少；比较浅；

-----  
它什么都讲了，但是什么都没讲清楚～带着问题而来，很不幸，都没有得到解答： 1)

所有逻辑队列共享一个 commitlog，那索引如何设计，如何保证高吞吐 2) 相对于 kafa，主、从关系需要在配置时指定，topic 和 逻辑队列需要通过命令指定，相对于 kafa 的自动分配，不觉得有何先进之处，为什么这么设计？ 3) 和 kafka 相比，优势在哪？劣势在哪？ 4) 主从同步模型（推 or 拉），是否具有类似 isr 的概念等等 感觉所以这本书真没啥价值，相对于 kafka 的书（入《kafka权威指南》），差距不是一点半点

-----  
@微信读书

2020年第4本书，首先要说本书不适合没有任何mq基础的读者阅读。前半部分像api介绍也像理解rocketmq的人的讲解。说了rocketmq的一些点，说得太泛，而又没到重点。比如作者说要设计一个高性能支持高并发的mq不容易。那么不容易在哪？没说。后半部分的源码大部分只粗粗讲解，看到最后突然结束我都有点愕然，并不推荐阅读。

-----  
学习rocketmq

-----  
100多页，基本上半天就看完了，  
将的比较浅显，大段大段的贴代码，但是实际的解析却非常少，有点坑

-----  
[RocketMQ实战与原理解析\\_下载链接1](#)

## 书评

RocketMQ广泛应用于交易、数据同步、缓存同步、IM通讯、流计算、IoT等场景。

第1章 快速入门

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。

1、应用解耦 2、流量消峰 3、消息分发

2007年Notify到2010年的Napoli，2011年升级后改为MetaQ，然后到2012年开...

-----  
[RocketMQ实战与原理解析\\_下载链接1](#)