# 参考：

<https://blog.csdn.net/prestigeding/category_7034876.html>

# 关键设计

## 整体架构

* NameServer：轻量级方式提供服务发现和路由功能，每个NameServer存有全量路由信息，提供对等的读写服务，支持快速扩缩容；
* Broker：负责消息存储，以Topic维度支持轻量级队列，单机可支撑上万队列规模，支持消息推拉模型，具备多副本容错机制（2或3副本），强大削峰填谷及上亿消息堆积能力，同事可严格保证消息有序性。同时提供同城异地容灾能力，丰富Metrics统计及告警机制。//都是传统消息系统无法比拟的。
* Producer：由用户进行分布式部署，消息由Producer通过多种负载均衡模式发送到Broker集群，发送低延时，支持快速失败
* Consumer：支持PUSH和PULL消费模式，支持集群消费和广播消息，提供实时消息订阅机制

## 存储设计

# 关键能力

## 分布式事务

如何通过事务消息保障抢购业务的分布式一致性

<http://3ms.huawei.com/km/groups/100591/blogs/details/9918145>

实现：两阶段提交

消息重试机制

## 死信队列

### 应用场景

# 线程池

RocketMQ深度解析 http://juejin.cn/post/6844903952807378957

# 使用经验

消息积压判断与解决

<https://blog.csdn.net/HiJamesChen/article/details/120442156>

RocketMQ对百万消息积压问题的处理方案

<https://blog.csdn.net/weixin_42405670/article/details/119153035>

rocketmq面对消息积压的处理

<https://blog.csdn.net/u013078871/article/details/123066366>

RocketMQ消息积压，异步方案，缓存策略解决方案

<https://juejin.cn/post/7086266210513322015>

# RocketMQ 高性能解密

zhuanlan.zhihu.com/p/93602392