RocketMQ文章

RocketMQ 的设计理念和目标

设计理念

RocketMQ 设计基于主题的发布与订阅 模式 其核心功能包括消息发送、 消息存储Broker 消息消费，整体设 追求简单与性能第一，主要体在如下三个方面：

- 首先，nameSever 设计极其简单，摒弃了业界常用的使用 Zookeeper 当信息管理“注册中心”，而是自研 NameServer 元数据管理。从实际需求出发，因为 Topic 路由信息无须在集群之间保持强一致，追求最终一致性，并且能容忍分钟级的不一致 正是基于此种情况 RocketMQ NameServer 集群之间互不通信，极大地降低了 NameServer 现的 复杂程度， 对网络的要求也降低了不少性能相比较 Zookeeper 有了极大的提升
- 其次是高效的 IO 存储机 RocketMQ 求消息发送的高吞吐量， RocketMQ 消息存储文件设计成文件组的概念，组内单个文件大小固定，方便引人内存映射机制，所有主题的消息存储基于顺序写极大地提供了消息写性能，同时为了兼顾消息消费与息查找引入了消息消费队列文件与索引文件
- 最后是容 存在设计缺陷 ，适当将某些工作下放给 RocketMQ 使用者 消息中间件的实现者经常会遇到一个难题：如何保证消息一定能被消息消费者消费，并且保证只消费一次。RocketMQ 的设计者给出的解决办法是不解决这个难题，而是退而求其次，只保证消息被消费者消费，但设计上允许消息被重复消费，这样极大简化了消息中间件的内核，使得实现消息发送高可用变得非常简单与高效，消息重复问题由消费者在消息消费时实现幕等。

设计目标

RocketMQ 作为一款消息中间件 ，需要解决如下问题：

1. 架构模式
   RocketMQ 与大部分消息中间件一样，采用发布订阅模式，基本的参与组件主要包括消息发送者、消息服务器（消息存储）、消息消费、路由发现
2. 顺序消患

所谓顺序消息，就是消息消费者按照消息达到消息存储服务器的顺序消费RocketMQ可以严格保证消息有序

3. 消息过滤

   消息过滤是指在消息消费时，消息消费者可以对同一主题下的消息按照规 只消费自己感兴趣的消息RocketMQ 消息过滤支持在服务端与消费端的消息过滤机制。

   a. 消息在 Broker 端过滤 Broker 只将消息消费者感兴趣的消息发送给消息消费者

   b. 消息在消息消费端过滤，消息过滤方式完全由消息消费者自定义，但缺点是有很多无用的消息会从 Broker 传输到消费端

4. 消息存储

   消息中间件的 个核心实现是消息的存储 ，对消息存储一般有如下两个维度的考量：消息堆积能力和消息存储性能 RocketMQ 追求消息存储的高性能引人内存映射机制，所有主题的消息顺序存储在同一个文件中。同时为了避免消息无限在消息存储服务器中累积，引入了消息文件过期机制与文件存储 间报警机制

5. 消息高可用性

   通常影响消息可靠性的有以下几种情况

   a. Broker 正常关机

   b. Broker 异常 Crash

   c. OS Crash

   d. 机器断电，但 能立即恢复供电情况

   e. 机器无法开机（可能是 CPU 、主板、 内存等关键设备损坏）

   f. 磁盘设备损坏

      针对上述情况，情况 1~4 RocketMQ 在同步刷盘机制下可以确保不丢失消息，在异步刷盘模式下，会丢失少量消息 情况 5–6 属于单点故障，一旦发生，该节点上的消息全部丢失，如果开启了异步复制机制， RoketMQ 能保证只丢失少量消息

6. 消息到达(消费)低延迟

   低延RocketMQ 在消息不发生消息堆积时，以长轮询模式实现准实时的消息推送模式

7. 确保消息必须消费一次

   RocketMQ 通过消息消费确认机制（ACK）来确保消息至少被消费一次 ，但由于 ACK消息有可能丢失等其他原因， RocketMQ 无法做到消息只被消费次，有重复消费的可能

8. 回溯消息

   回溯消息是指消息消费端已经消费成功的消息，由于业务要求需要重新消费消息RocketMQ 支持按时间回溯消息，时间维度可精确到毫秒，可以向前或向后回溯

9. 消息堆积

   消息中间件的主要功能是异步解锢，必须具备应对前端的数据洪峰，提高后端系统的可用性，必然要

求消息中间件具备 定的消息堆积能力 RocketMQ 消息存储使用磁盘文件（内存映射机制），并且在物理布局上为多个大小相等的文件组成逻辑文件组，可以无限循环使用 RocketMQ 消息存储文件并不是永久存储在消息服务器端，而是提供了过期机制，默认保留3天

10. 定时消息

    时消息是指消息发送到 Broker 后， 不能被消息消费端立即消费，要到特定的时间点或者等待特定的时间后才能被消费 如果要支持任意精度的定时消息消费，必须在消息服务端对消息进行排序，势必带来很大的性能损耗，故 RocketMQ 不支持任意进度的定时消息，而只支持特定延迟级别

11. 消息重试机制

    消息重试是指消息在消费时，如果发送异常，消息中间件需要支持消息重新投递，RocketMQ 支持消息重试机制