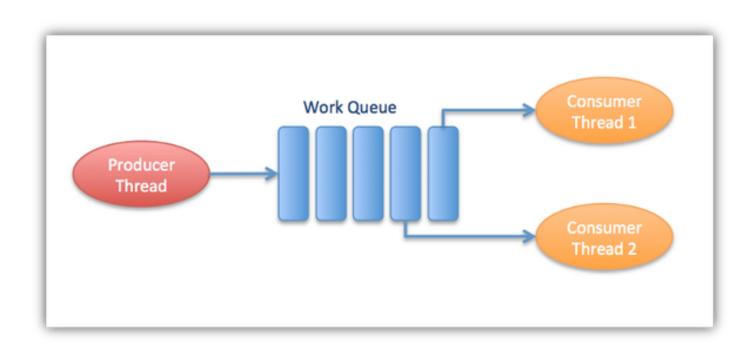


关注专栏

三 写文章 。。。



## RocketMQ源码阅读(1)

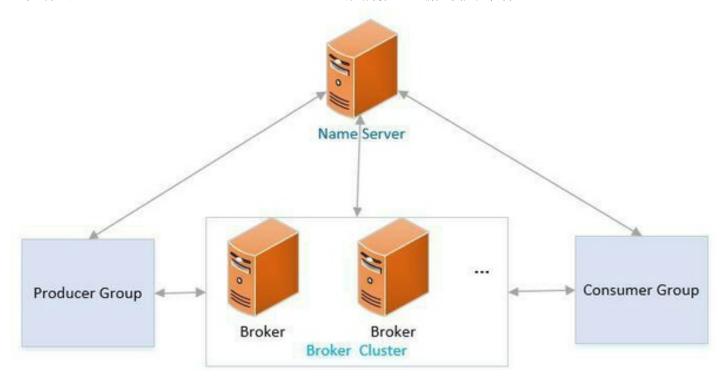


Chown · 2 个月前

## RocketMQ 简介

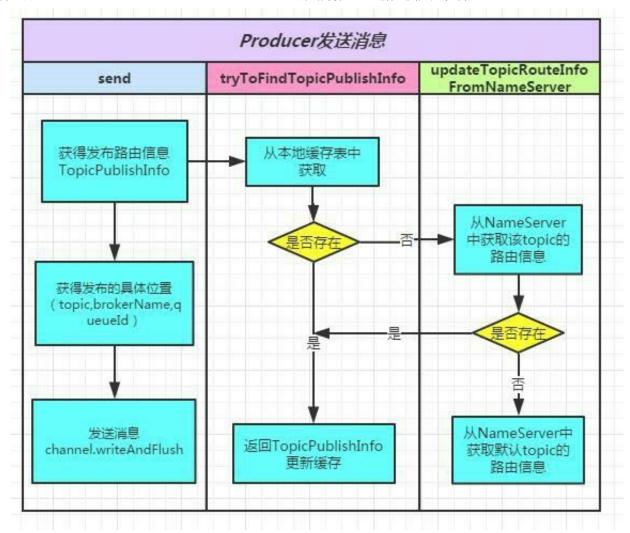
RocketMQ 是一款开源的消息中间件,采用Java实现,设计思想来自于Kafka(Scala实现),在具体设计时体现了自己的选择和需求,具体差别可以看RocketMQ与Kafka对比。接下来是自己阅读源码的一些探索。

RocketMQ的整体架构如下,可以看到各个组件充当的角色,Name Server 负责维护一些全局的路由信息:当前有哪些broker,每个Topic在哪个broker上; Broker具体处理消息的存储和服务;生产者和消费者是消息的源头和归宿。



#### Producer 发送消息

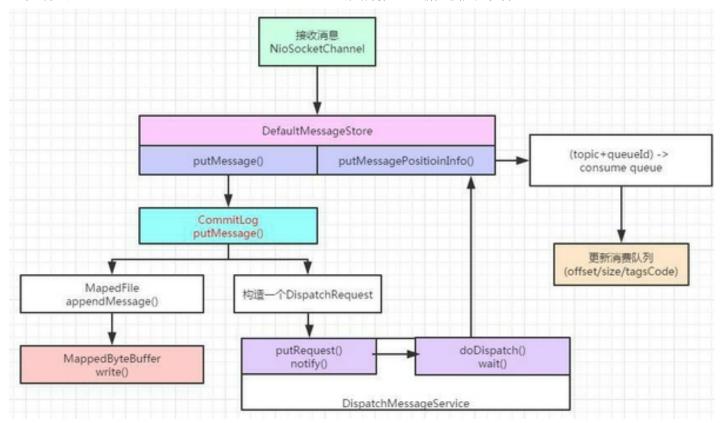
Producer发送消息是如何得知发到哪个broker的 ? 每个应用在收发消息之前,一般会调用一次producer.start()/consumer.start()做一些初始化工作,其中包括:创建需要的实例对象,如MQClientInstance;设置定时任务,如从Nameserver中定时更新本地的Topic route info,发送心跳信息到所有的 broker,动态调整线程池的大小,把当前producer加入到指定的组中等等。客户端会缓存路由信息TopicPublishInfo,同时定期从NameServer取Topic路由信息,每个Broker与NameServer集群中的所有节点建立长连接,定时注册Topic信息到所有的NameServer。Producer在发送消息的时候会去查询本地的topicPublishInfoTable(一个ConcurrentHashMap),如果没有命中的话就会询问NameServer得到路由信息(RequestCode=GET\_ROUTEINTO\_BY\_TOPIC)如果nameserver中也没有查询到(表示该主题的消息第一次发送),那么将会发送一个default的topic进行路由查询。具体过程如下图所示。



Producer 在得到了具体的通信地址后,发送过程就显而易见了。通过代码可以看到在选择消息队列进行发送时采用随机方式,同时和上一次发送的broker保持不同,防止热点。

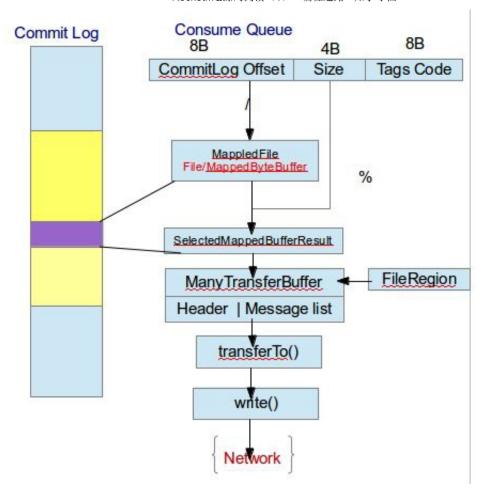
## Broker处理来自Producer的消息

每个producer在发送消息的时候都和对应的Broker建立了长连接,此时broker已经准备好接收Message, Broker的SendMessageProcessor.sendMessage处理消息的存储,具体过程如下。接收到消息后,会先写入Commit Log文件(顺序写,写满了会新建一个新的文件),然后更新Consume queue文件(存储如何由topic定位到具体的消息)。



## RocketMQ 存储特点

RocketMQ的消息采用顺序写到commitlog文件,然后利用consume queue文件作为索引,如图。RocketMQ采用零拷贝mmap+write的方式来回应Consumer的请求,RocketMQ宣称大部分请求都会在Page Cache层得到满足,所以消息过多不会因为磁盘读使得性能下降,这里自己的理解是,在64bit机器下,虚存地址空间(vm\_area\_struct)不是问题,所以相关的文件都会被映射到内存中(有定期删除文件的操作),即使此刻不在内存,操作系统也会因为缺页异常进行换入,虽然地址空间不是问题,但是一个进程映射文件的个数(/proc/sys/vm/max\_map\_count)是有限的,所以可能在这里发生OOM。



通过Broker中的存储目录(默认路径是\$HOME/store)也能看到存储的逻辑视图:

```
tree store
store
 abort
   checkpoint
   commitlog
    — 0000000000000000000000
    — 000000000001073741824
   config
      - consumerOffset.json

    consumerOffset.json.bak

    — delayOffset.json

    delayOffset.json.bak

    ─ topics.json
   — topics.json.bak
   consumequeue

    TopicTest

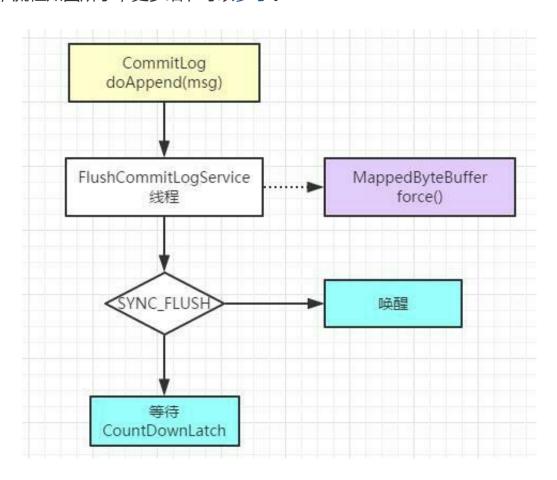
          - 0
            TopicTest2
              - 0000000000000000000000
              - 000000000000000000000
              - 0000000000000000000000
              - 0000000000000000000000
   index
     — 20160822130610979
```

## 顺序消息是如何保证的?

需要业务层自己决定哪些消息应该顺序到达,然后发送的时候通过规则(hash)映射到同一个队列,因为没有谁比业务自己更加知道关于消息顺序的特点。这样的顺序是相对顺序,局部顺序,因为发送方只保证把这些消息顺序的发送到broker上的同一队列,但是不保证其他Producer也会发送消息到那个队列,所以需要Consumer在拉到消息后做一些过滤。

## RocketMQ 刷盘实现

Broker 在消息的存取时直接操作的是内存(内存映射文件),这可以提供系统的吞吐量,但是无法避免机器掉电时数据丢失,所以需要持久化到磁盘中。刷盘的最终实现都是使用NIO中的 MappedByteBuffer.force() 将映射区的数据写入到磁盘,如果是同步刷盘的话,在Broker把消息写到CommitLog映射区后,就会等待写入完成。异步而言,只是唤醒对应的线程,不保证执行的时机,流程如图所示,更多细节可以参考。



## 消息过滤

类似于重复数据删除技术(Data Deduplication),可以在源端做,也可以在目的端实现,就是网络和存储的权衡,如果在Broker端做消息过滤就需要逐一比对consume queue 的 tagsCode 字段(hashcode),如果符合则传输给消费者,因为是 hashcode,所以存在误判,需要在 Consumer 接收到消息后进行字符串级别的过滤,确保准确性。

## 小结

这次代码阅读主要着眼于消息的发送过程和Broker上的存储,其他方面的细节有待深入。

消息队列

Java EE

Java Web



①分享 学报











#### 文章被以下专栏收录



编程之路

进入专栏

#### 6条评论



写下你的评论



#### 张当汤

支持作者多做源码分析

2个月前



Chown (作者)回复 张当汤

② 查看对话

共同学习

2个月前 1 赞



和中堂回复 Chown (作者)

② 查看对话

赞,好好学习,天天向上,世界就是你们的。

2个月前



Chown (作者) 回复 和中堂

② 查看对话

/(⊙\_\_\_⊙)b

2个月前



#### 和中堂回复 Chown (作者)

② 查看对话

这什么表情?不满意我夸奖你。楼主你一定能赶上马云的,出任CEO,赢取白富美,走向人生巅峰。谢谢。

2个月前



#### Chown (作者)回复 和中堂

② 杳看对话

我竟无言以对

2个月前

#### 推荐阅读

## Spring源码阅读(2)-bean解析初体验

从一个简单例子开始通过一个简单的bean加载例子来热热身,虽然我们平时不使用这里的XmlBeanFactory,而是用ApplicationContext,但是后面我们看… 查看全文 >

Chown · 2 个月前

发表于 编程之路

#### 你和旗鱼只剩一个钩叔的距离——记马来西亚云冰行

前言:本文写于9月20日—9月26日期间。钩叔受腾讯天天快报康少见主编的推荐,以及马来西亚旅游局的邀请,有幸以海钓新媒体的身份参加于马来西... 查看全文 >

胡海云 · 19 天前 · 编辑精选

#### 【节气手帖】秋分:木樨蒸

木樨是桂花的古名。前人的记载中,带有"桂"字的香料植物极多。如肉桂、菌桂之流,但因桂花的木质纹理似犀角,所以许它"木樨"之名,以示区... 查看全文 >

蔓玫·1个月前·编辑精选

发表于 花事未了

# 小皮 | 30块的便宜酒和1000块的顶级酒到底有哪些不同?一次性的全部告诉你

过节,和朋友聚会,我带上了好酒;他们看到酒居然没有脸部抽搐、抱头痛哭,当我轻描淡写的说出了酒的价格,换来的只是咿呀咿呀的质疑,如果没...查看全文>

小皮老师 · 6 天前 · 编辑精选

发表于 小皮葡萄酒讲堂