# Rocketmq

## Tips

Queue messageid commitlog queuelog

阿里出品 java写的

不是同一个queue里面的消息是不能保证顺序消费的

rocketmq可以实现锁库存释放的操作,通过消息延时发送.

金融级别的高可靠的消息中间件.

代码没有注释.

可以实现分布式事务.

分布式主从集群都可以

也有注册中心,类似与kafka的zookeepr,叫做name server,没有主从关系

Kafka只支持部分有序,必须发到一个partination里,其实这个也是必须发到同一个queue

支持定时消息,开源版不能自定义时间,只能用他们的给定是时间.

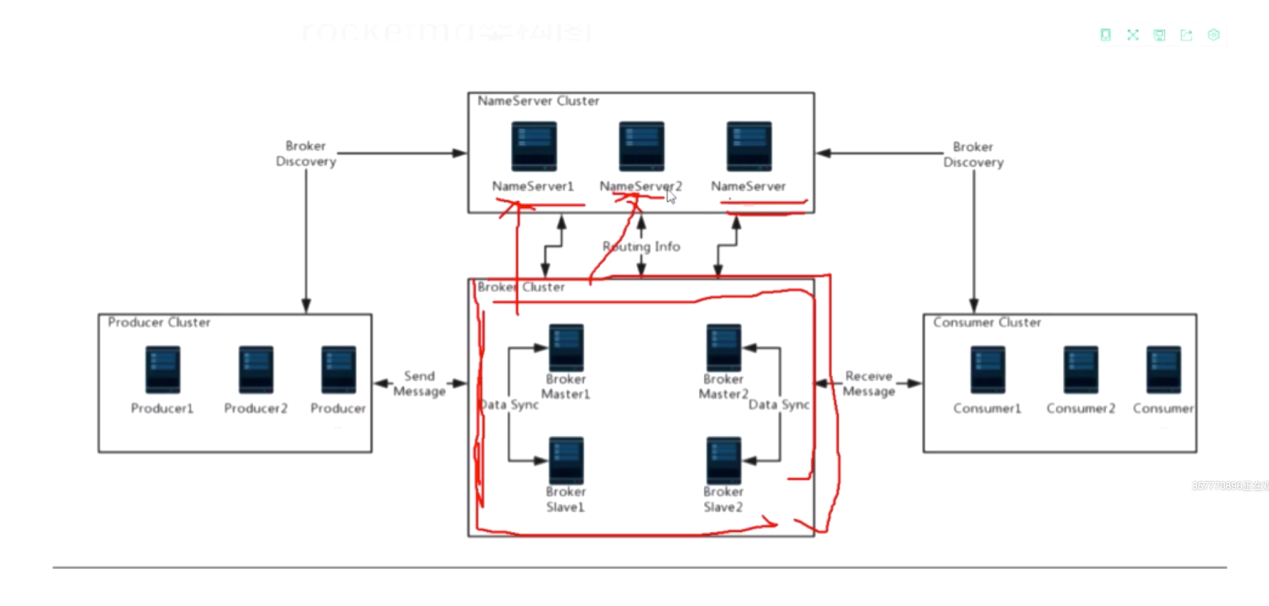
支持顺序消费.

唯一的支持事务消息.

支持messageid查询,有messageid这个东西.

支持失败重试.

支持重新消费.



nameserver之间不是集群关系没有主从关系,就是一个个的副本,每一个都有全量内容.没有选举啊同步啥的.

consumer去nameserver上拉去各种什么broker的信息,topic啊,ip啥的,偏移量啊,然后拿到broker的信息后就可以去broker那里消费了.

所以consumer主要是根据nameserver的指引进行操作的,由于namserver是无状态的,所以consumer是知道所有的nameserver的地址的,哪个可以用就用哪个.

broker和nameserver之间会有心跳消息同步各种数据.

每一个topic都会自动创建四个queue 0 1 2 3每一个queue都会有一个持久化的文件.消息内容会全部存储在commitlog中,queuelog记录每条消息在commitlog中的位置等信息.

对于

commitlog是多个topic的公有财产,queue 里的东西才是topic私有的.

还会有一个consumeroffset.json,用来记录每一个consumer消费到了哪里,格式如下.

格式 topic@consumergroup{

Queue 0 :offset

Queue 1 :offsert

Queue 2 :offset

Queue 3:offset

}

mq也是和mysql一样会有一个刷盘的线程把内存的内容刷到硬盘所以用这两种结合的方式操作.

rockermq支持pull模式和push模式

pull模式就是应用主动调用拉消息的方法,这怎么可以呢,肯定是不行的.

push模式就是consuemr会有个轮询去问broker还有没有消息.

我们肯定要用push模式啊.

producer和consumer和broker都需要把自己注册到nameserver上,producer和broker注册都是为了获取到broker的信息.你得知道哪个topic在哪个的broker.

发送方有同步和异步之说,异步的话可以设置成功或失败的回调函数.

producer和consumer都是有group的.

如果producer和consumer在同一个group里,consumer可以指定偏移量如果不在同一个组里就不行.

默认都是push的方式.

消费的方式分为广播和单播,广播就是一个组里面没个consumer都会收到消息,单播就是组里面只会一个会收到.

发消息也可以批量发送.procker的clinet里面配置的.

消息里面可以加一些tag可以根据这些tag过滤啊 操作啥的.

consumer可以通过bysql功能根据tag过滤消息.

定时消息是指消息发送到broker后,不能立即被consumer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.就是set个level就可以了.

实现方式就是把延迟消息发到一个topic里面,然后定时扫这个topic,时间到了后再转发到真正的topic里面.如果是开源版的话这个扫面线程性能不够的,消息多了定时可能就不准了.

broker也是有失败的retry的.

rocketmq是有事务消息的是比kafka最好的地方,原因就是broker和producer可以互相通信互相发消息.

顺序消费.也就是那样,比如通过订单号模上queue数,然后让所有的消息发到指定的同一个queue里面就可以了,跟kafka发到同一个partination是一个意思.

事务消息是利用了一个half topic.

定时消息是利用了一个schdule topic.

还是回到我们的问题，只要这张小票在，你最终是能拿到炒肝的。同理转账服务也是如此，当支付宝账户扣除1万后，我们只要生成一个凭证（消息）即可，这个凭证（消息）上写着“让余额宝账户增加1万”，只要这个凭证（消息）能可靠保存，我们最终是可以拿着这个凭证（消息）让余额宝账户增加1万的，即我们能依靠这个凭证（消息）完成最终一致性。

这个小票就是我们这里的消息.

我们可以开个事务,这个事务里在支付里扣钱,然后写一条余额宝加钱的消息,然后异步的把消息发到余额宝,余额宝收到消息后也不能直接去操作,而是要先来支付宝这里查询一下这个消息有没有重复啥的,消息状态是不是对的,然后再操作.

这个叫事务型消息,不是事务,就是先发一个prepare消息过去,这个消息现在half topic存着,然后执行本地事务,然后发送commit消息,broker收到后把halftopic里的消息发送到真正的topic里,如果commmit消息失败了不就完了么,broker会检查自己收到的prepare消息,如果发现老久都没有收到commit或者rollback消息就会去producter去问的.