## 消息队列的作用

1，解耦合：上游关注通知，而不关注下游的处理

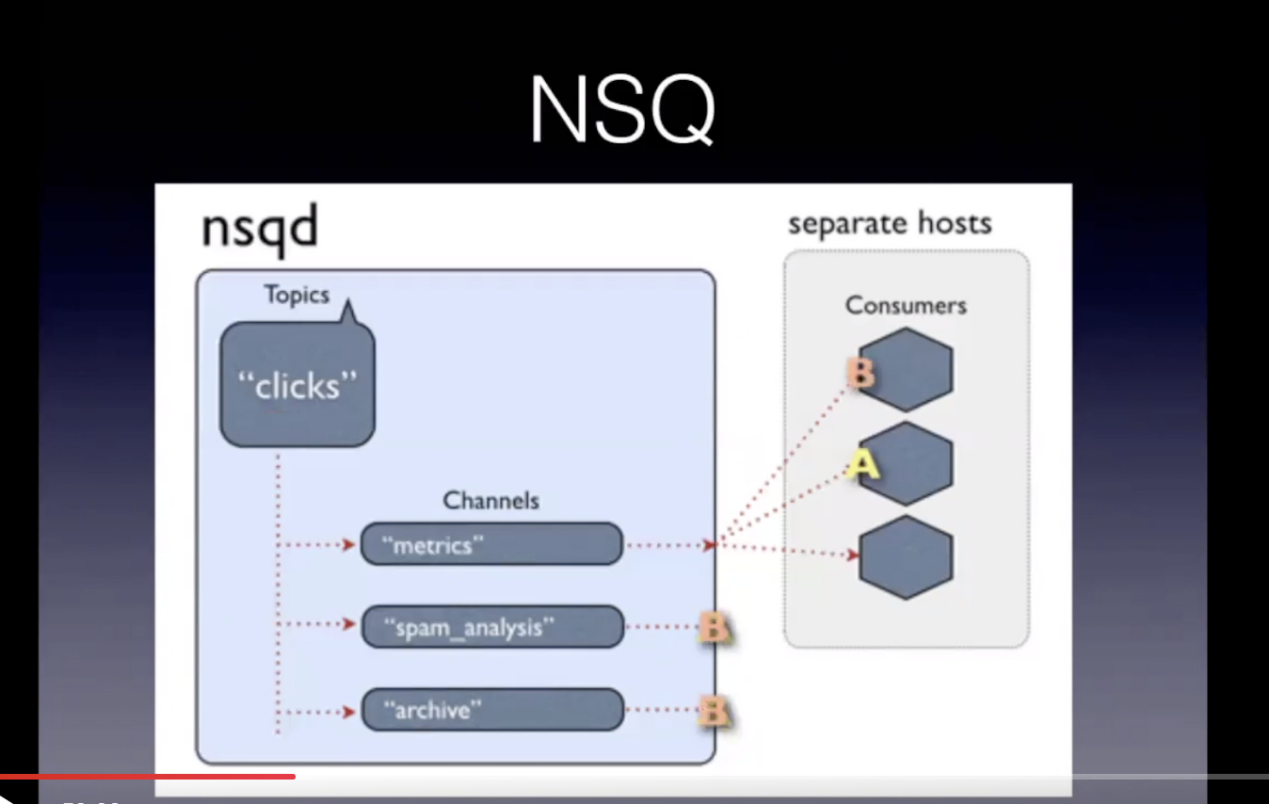
2，缓冲，应对突发流量

3，广播，一条消息可以被多个下游分别处理

4，持久化，到底处理过什么消息可以记住，方便出问题时的回溯和以后的验证等

kafka这种是会把发过的消息存起来的，是可以记录都发过什么消息的。

## NSQ



nsqd相对与kafka的broker。

主要的组件是nsqd，里面有topics，在这里就当消息队列用，一个队列的消息会被发送到所有的channel上，为啥一个消息要发送给多个channels呢，一个消息被处理一次不就行了，正常是这样的，但是呢，头条这边对一个消息不仅仅是处理，还有监控啊，日志归档等需求，监听这个topic的多个channel可能只有一个是处理业务的，别的可能都是处理这些的。

所以啊头条用的都是topic，然后真正的业务cosumer从业务channel里取消息在做业务处理。

对于一个channel也是有多个consumer去监听的。

这个nspd从输入端看是一个topic，从输出端看是多个channel。

nsq的其他功能，requeue，处理失败或超时的消息会重新进队列（amq好像是重发诶）

defer可以让消息在一个指定的时间段之后才被消费。

nsq是没有持久化的，所以nsq挂了会丢消息。

投递保证是at least once 至少被消费一次，消息都不持久化at least once也不能保证。

At most once 最多被消费一次。 还有个正好一次。

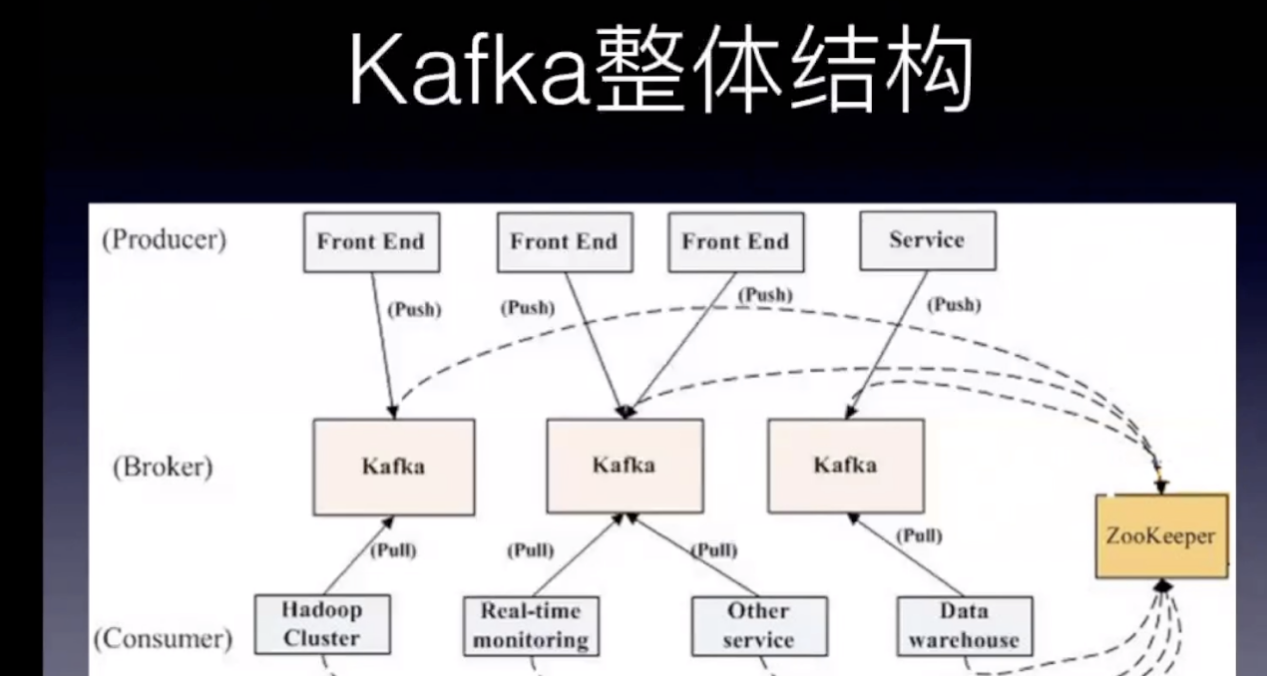
nsq也不能保证消息的顺序，首先requeue和defer这俩机制就会乱序，还有就是nsq发现自己内存不足了，就会吧消息弄到磁盘里，过一会就读出来，这样玩肯定也会导致乱序。

是不能回溯的,因为他没有持久化啊，也就是topic和channel都没有持久化。

nsq是个推模型，也就是nsq主动把消息push到cosumer。

吞吐量也可以，相对于kafka的话小一点。

## Kafka



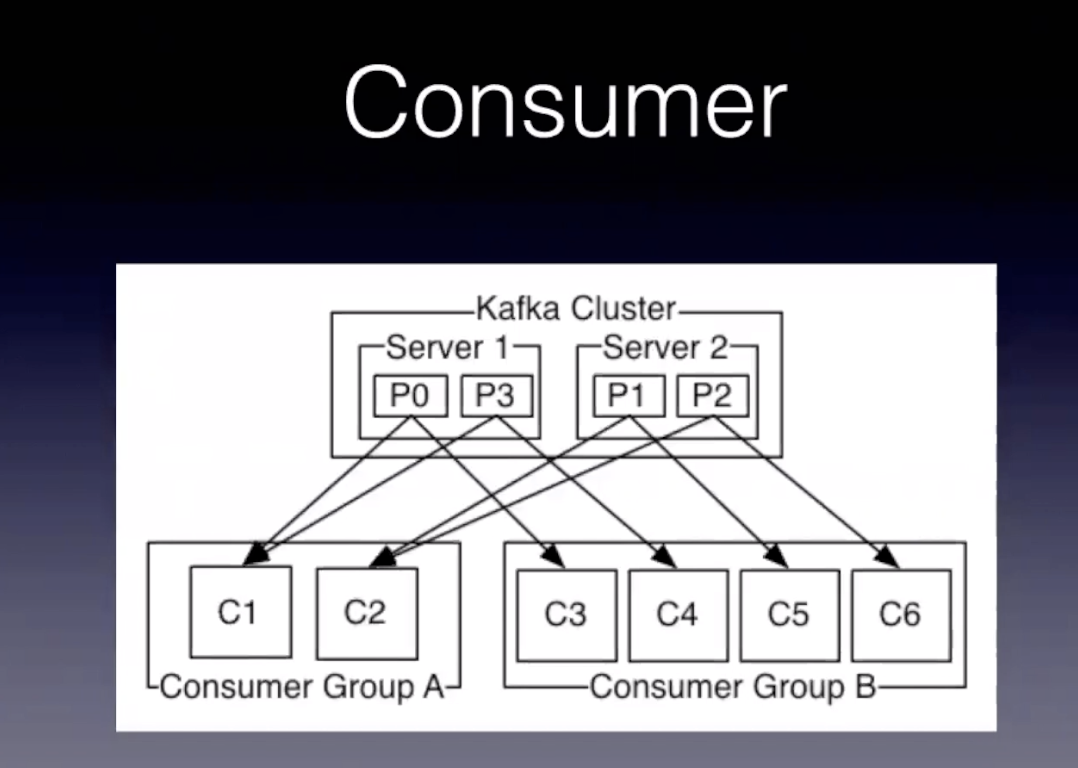
kafka的broker对应的nsq的nsqd

kafka的中间的broker会和zookeeper连着。



每个topic包含多个partition，发送消息的时候要选择一个partition来发，单个partiton内的消息是有时序的。

Consuemrgroup 怎么理解，看下图。



也就是说，我们这里就先不看topic了，而是看partition，每个consumer group都会收到所有的partition的消息，然后多个partion的消息到group里面被分配到group里面的consumer处理，平均分的。

也就是这里获取消息我们是从partition里获取的。

消息到了partition会被持久化。

投递的保证，可以支持 at least once ,at most once . exactly once.

Exactly once的保证需要做接口幂等。

partition内部有序，保证不了topic级别的全局有序。

可回溯。

由于partition内是有序的，所以requeue和defer功能就不能用了。

kafka是个pull模型，是consumer主动去partition拉消息的，这样的话对于consumer的效率情况比较友好，你忙的话自己就不要去拉了，你不忙自己就多来拉。

吞吐很高。

## TMQ

封装了kafka和nsq。