关于topic和groupid的关系

```
topic到group之间，是发布订阅的通信方式
即一条topic会被所有的group消费，属于一对多的模式
group与消费者之间是点对点通信，是一对一的（一个group对应一种消费者，当然一种消费者可以部署多台机

也就是说每一种消费者的groupid通常都是一样的，因为一个消息通常一种消费者中一个消费了就行，没必要都

不使用group的话，启动10个consumer消费一个topic，
这10个consumer都能得到topic的所有数据，相当于这个topic中的任一条消息被消费10次。
```

消费者组group，topic，consumer之间的关系

```
kafka消费端确保一个 Partition 在一个消费者组内只能被一个消费者消费。
    1.在同一个消费者组内，一个 Partition 只能被一个消费者消费。
    2.在同一个消费者组内，所有消费者组合起来必定可以消费一个 Topic 下的所有 Partition。
    3.在同一个消费组内，一个消费者可以消费多个 Partition 的信息。
    4.在不同消费者组内，同一个分区可以被多个消费者消费。
    5.每个消费者组一定会完整消费一个 Topic 下的所有 Partition。
总结:
    group用来管理consumer，保证一个消息只会被本group下的一个消费者消费
    topic的每条消息都会发送给订阅的所有group，也就是一个group的消费者消费的消息加起来就是所有
    一个group的消费者数量上限等于topic的分区数量（上述第一条）----重点
    一个group貌似可以订阅多个topic
```

kafka的作用

这是一种消息队列，主要解决一下三个问题

解耦

异步

削峰

关kafka的原理（https://www.cnblogs.com/liuwei6/p/6900686.html）

```
1. 每个消息都需要指定topic，每个topic会有多个partition，每个partition及之间的消息是不会重复
    这是一种天然的负载均衡提高吞吐量
    同时每个partition的消息会发送给所有的group，也就是说消费者的group不同就是广播，都是相同的
2. offset
    offset指的是消费者group的偏移量，zookeeper会保存每个topic下每个partition在每个group中
    以保证每个消息只会被同一个groupid的一个消费者消费
```

```
2. 消息删除
```

```
8       一般的消息队列会删除已经被消费的消息，但是kafaka集群会保留所有的消息，无论被消费与否
9       但是限于磁盘限制，永久保留不是很正常，因此kafka提供了两种策略删除旧数据，
10          1、基于时间 2、基于parttion大小 可通过配置文件配置删除策略
```

避免消息丢失的机制，三个角度，生产者，消费者，服务端

注意：对于避免消息丢失，也可以加消息处理表，

1、在第一次发送消息时可以判断是否已经插入过，如果插入过则说明是重复的消息，

2、消息在消费时判断消息处理表的状态是否是已经处理，如果已经处理过则无需在处理

3、这时可以扫描消息处理表，如果发现有处理记录长时间未处理，则可以补提交这个消息

生产者：

发送方式：简单发送（不关注是否成功），同步发送（等待发送成功），异步发送（利用线程配置发送成功的回调函数）

配置ack参数：0：只发送，不关注是否成功，1：leader节点收到消息就返回给生产者，all：leader节点收到副本节点也收到也返回生产者

这里可以定义单批次发送的数量以及一个批次等待的时间

消费者

消费端异常保障：关闭自动提交，消息消费完成后再提交

服务端：

消息发送后，消息到了leader，如果在同步给其他副本的时候，leader挂了，再次选中主，新主没有这条消息就会丢失

解决：设置只能这种与leader一定程度同步的副本成为新的leader，而不是之后较多的副本OSR

参考链接：https://www.cnblogs.com/cfas/p/16593290.html

kafka保证高吞吐量

顺序读写

使用顺序io提高磁盘的写入速度，数据顺序插入到文件末尾，消费者通过控制偏移量来读取消息

两种删除策略：基于时间删除，基于分区文件大小

内存文件映射（memory mapped files

零拷贝

正常发送文件是从服务器读取文件时，服务器先将文件复制到内核空间，再复制到用户空间，最后复制到内核空间在通过网卡发送出去

linux kernal2.2出现零拷贝系统调用机制，即跳过“用户缓冲区”的拷贝，建立一个磁盘空间和内存的直接映射，数据不再，数据不再复制到用户态缓冲区，系统切换减少了两次

发送消息时，kafka把消息存放在一个文件中，消费者需要数据时直接将文件发送给消费者，，消费者通过传过来的偏移量利用零拷贝技术读取数据

分区

批量发送

数据压缩

1、点对点消息顺序传递，一个消费者在分区中只有一个位置

2、不标识消息的状态，通过偏移量做标记，offset之前的是已经消费的，之后是未消费的

3、支持点对点消息传送

参考链接：https://blog.csdn.net/weixin_48978790/article/details/123497123