什么事kafka？上

Kafka是分布式的发布—订阅消息系统。它最初由LinkedIn(领英)公司发布，使用Scala语言编写，与2010年12月份开源，成为Apache的顶级项目。 Kafka是一个高吞吐量的、持久性的、分布式发布订阅消息系统。 它主要用于处理活跃的数据(登录、浏览、点击、分享、喜欢等用户行为产生的数据)。

特性：

1. 吞吐量：可以满族每秒百万级别消息的生产和消费
2. 持久性：有一套完善的消息存储机制，确保数据的高效安全的持久化
3. 分布式：基于分布式的扩展和容错机制，Kafka的数据都会复制到几台服务器上。当某一台故障失效时，生产者和消费者转而使用其它的机器——整体健壮性。

相关概念：

Topic:主题

Broker:消息代理，kafka集群中的kafka节点。

Partition:topic物理上的分组一个topic可以被分为多个分区，创建topic时指定。在存储层面上是append log文件。新消息直接追加到log文件的尾部，每条消息在log文件中的位置称为offset（偏移量）

Message:

消息，通信的基本单位。三个属性offset(long类型，可以理解为消息在partition中的序号) messageSize、data

Producer: 生产者

Consumer:消费者

Zookeeper:协调kafka

Offset:

Mq的分类：

两类：点对点、topic

Offset机制：

作用；kafka在消费过程中即使挂了或者引发再均衡问题重新分配partition，下次继续消费时任然知道从哪里消费，和标签类似

有两种提交方式：自动提交（默认），手动提交

自动提交：

enable\_auto\_commit、auto\_commit\_interval\_ms控制。enable\_auto\_commit为true时，在消费的过程中会议频率为auto\_commit\_interval\_ms向topic进行偏移量提交，提交分区以算法partation=hash(group\_id)%50来计算的，group\_id=test\_group\_1，则partation=hash("test\_group\_1")%50=28

如果auto\_commit\_interval\_ms值过大，当消费者在提交之前异常退出，将导致偏移量未提交，会大致重复消费的问题，所以改制越小越好

手动提交：

3种方式，同步、异步、同步+异步

同步：失败会一直充实提交，直到无法重试的情况在结束。同步提交时消费者县城在拉取消息是会被阻塞，会限制吞吐量

异步+回调函数：异步提交时不会阻塞消费者线程，提交失败也不会进行重试，可以配合回调函数做出错误记录

异步+同步组合方式提交: 异步提交会存在丢失问题，通过对消费者进行异步批次提交并且在关闭时同步提交的方式，这样即使上一次的异步提交失败，通过同步提交还有补救的机会，同步会一直重试，直至成功

Kafka如何解决消息丢失和消息重复消费问题？

消息丢失如何产生：

Offset设置为自动定时提交时，当正在提交时，数据还在处理中，此时宕机，offset已提交单数据未处理完，导致该部分数据丢失

解决方案：

对kafka限速，启动重试机制，重试时间设置长一些，kafka设置acks=all，即需要相应的所有处理ISR的分区都确认收到消息后，才算发送成功