**Kafka学习之怎么保证不丢，不重复消费数据**

参考：<http://blog.csdn.net/u010627840/article/details/76435385>

Kafka作为当下流行的高并发消息中间件，大量用于数据采集，实时处理等场景，我们在享受他的高并发，高可靠时，还是不得不面对可能存在的问题，最常见的就是丢包，重发问题。

* 丢包问题：消息推动服务，每天早上，手机上各终端都会给用户推送消息，这时候流量剧增，可能会出现kafka发送数据过快，导致服务器网卡爆满，或者磁盘处于繁忙状态，可能会出现丢包现象。

解决方案：首先对kafka进行限速，其次启用重试机制，重试间隔时间设置长一些，最后Kafka设置acks=all。   
检测方法：使用重发机制，查看问题所在。   
kafka配置如下：

props.put("compression.type", "gzip");

props.put("linger.ms", "50");

props.put("acks", "all");

props.put("retries ", 30);

props.put("reconnect.backoff.ms ", 20000);

props.put("retry.backoff.ms", 20000);

* 重发问题：当消费者重新分配partition的时候，可能出现从头开始消费的情况，导致重发问题。当消费者消费的速度很慢的时候，可能在一个session周期内还未完成，导致心跳机制检测报告出问题。

底层根本原因：已经消费了数据，但是offset没提交。   
配置问题：设置了offset自动提交   
问题场景：

1.设置offset为自动提交，正在消费数据，kill消费者线程；

2.设置offset为自动提交，关闭kafka时，如果在close之前，调用 consumer.unsubscribe() 则有可能部分offset没提交，下次重启会重复消费   
重复消费最常见的原因：re-balance问题,通常会遇到消费的数据，处理很耗时，导致超过了Kafka的session timeout时间（0.10.x版本默认是30秒），那么就会re-balance重平衡，此时有一定几率offset没提交，会导致重平衡后重复消费。   
去重问题：消息可以使用唯一id标识