

1.kafka的message包括哪些信息

2. 怎么解决kafka的数据丢失

1.kafka的message包括哪些信息

一个Kafka的Message由一个固定长度的header和一个变长的消息体body组成header部分由一个字节的magic(文件格式)和四个字节的CRC32(用于判断body消息体是否正常)构成。当magic的值为1的时候，会在magic和crc32之间多一个字节的属性：attributes(保存一些相关属性，比如是否压缩、压缩格式等等)；如果magic的值为0，那么不存在attributes属性。body是由N个字节构成的一个消息体，包含了具体的key/value消息。

2. 怎么解决kafka的数据丢失

1. producer端：

宏观上看保证数据的可靠安全性，肯定是依据分区数做好数据备份，设立副本数。

2. broker端：

topic设置多分区，分区自适应所在机器，为了让各分区均匀分布在所在的broker中，分区数要大于broker数。

分区是kafka进行并行读写的单位，是提升kafka速度的关键。

3. Consumer端

consumer端丢失消息的情形比较简单：如果在消息处理完成前就提交了offset，那么就有可能造成数据的丢失。由于Kafka consumer默认是自动提交位移的，所以在后台提交位移前一定要保证消息被正常处理了，因此不建议采用很重的处理逻辑，如果处理耗时很长，则建议把逻辑放到另一个线程中去做。为了避免数据丢失，现给出两点建议：

- enable.auto.commit=false 关闭自动提交位移
- 在消息被完整处理之后再手动提交位移

3.为什么Kafka不支持读写分离？

也就是说：**kafka对某一个主题的读写是在一个节点完成的**，分区的从节点都是用来同步的

既然是在一个节点, 为什么能做到高并发?(大概是因为: 对主题的读写本质是对主分区的读写, 只要主分区尽可能的分配均匀至每个节点, 就能分散压力)

因为这样有两个明显的缺点:

1. 数据一致性问题。数据从主节点转到从节点必然会有一个延时的时间窗口, 这个时间窗口会导致主从节点之间的数据不一致。
2. 延时问题。数据从写入主节点到同步至从节点中的过程需要经历网络→主节点内存→主节点磁盘→网络→从节点内存→从节点磁盘这几个阶段。
对延时敏感的应用而言, 主写从读的功能并不太适用。

对于Kafka来说, 必要性不是很高, 因为在Kafka集群中, 如果存在多个副本, 经过合理的配置, 可以让leader副本均匀的分布在各个broker上面, 使每个broker 上的读写负载都是一样的。

4. Kafka中的延迟队列

在发送延时消息的时候并不是先投递到要发送的真实主题 (real_topic) 中, 而是先投递到一些 Kafka 内部的主题 (delay_topic) 中, 这些内部主题对用户不可见, 然后通过一个自定义的服务拉取这些内部主题中的消息, 并将满足条件的消息再投递到要发送的真实的主题中, 消费者所订阅的还是真实的主题。