**Kafka 练习题**

# 选择题

1. Kafka服务器默认能接收的最大消息是多大? （单选） A

A：1M

B：10M

C：100M

D：没有大小限制，因为支持大数据

1. Kafka的特性（多选） ABCD

A：高吞吐量、低延迟

B：可扩展性、持久性、可靠性

C：容错性

D：高并发

1. kafka 容错性说话正确的是 ABCD

A：允许集群中节点失败（若副本数量为n,则允许n-1个节点失败）

B：杀死集群中的主节点还是可以使用

C：杀死集群中所有的副节点还是可以使用

D：无论结束的是主还是副broker，都不会影响kafka使用，只要还有一个副本

1. zookeeper在kafka中的作用（多选） AC

A：kafka 将元数据信息维护在zookeeper上

B：kafka的producer、consumer、topic、partition都需要依赖zookeeper

C：kafka使用zookeeper来实现动态的集群扩展

D：发送到topic的数据需要经过zookeeper

1. Kafka相对传统技术有什么优势说法正确的是 ABCD

A：快速:单一的Kafka代理可以处理成千上万的客户端，每秒处理数兆字节的读写操作。

B：可伸缩:在一组机器上对数据进行分区和简化，以支持更大的数据

C：持久:消息是持久性的，并在集群中进行复制，以防止数据丢失。

D：设计:它提供了容错保证和持久性

1. Kafka负载均衡原理说法正确的是 ABCD

A: producer根据用户指定的算法，将消息发送到指定的partition

B:存在多个partiiton，每个partition有自己的replica，每个replica分布在不同的Broker节点上

C:多个partition需要选取出lead partition，lead partition负责读写，并由zookeeper负责fail over

D:通过zookeeper管理broker与consumer的动态加入与离开

1. 关于partition中的数据是如何保持到硬盘上的，说法正确的是 ABC

A:一个topic分为好几个partition，每一个partition是以文件夹的形式保持在broker节点上面的

B:每一个分区的命名是从序号0开始递增

C:每一个partition目录下多个segment文件（包含xx.index，xx.log）,默认是1G

D:每个分区里面的消息是有序的，数据是一样的

# 判断题

1. 一个topic有多个partition，消息会存在leader的partition中，其他partition同步leader的数据？

错

1. kafka 支持热扩展，及扩及用

对

1. kafka有可能在生产消息后发生消息偏移

错

1. 就算首选副本不在ISR中，controller也能将leadership转到首选副本

错

# 练习题（需要自己动手）

1. 搭建并启动一个3个节点的kafka集群，并配置使用3个节点的zookeeper集群。
2. 在搭建的 kafak 集群中创建一个 test 的 topic，指定三个分区三个副本。再次上学会 kafka 的容错扩容等知识点。
3. 自定义一个 producer 拦截器，在消息之前加上“网易云课堂”。
4. 根据课上老师的讲解，尝试自己一个 stream 例子。
5. 运行 kafka manager 监控工具，尝试在 web 端去去分区副本迁移。

# 综合题

1. 为什么要使用消息队列或 kafka

**缓冲和削峰**：上游数据时有突发流量，下游可能扛不住，或者下游没有足够多的机器来保证冗余，kafka在中间可以起到一个缓冲的作用，把消息暂存在kafka中，下游服务就可以按照自己的节奏进行慢慢处理。

**解耦和扩展性**：项目开始的时候，并不能确定具体需求。消息队列可以作为一个接口层，解耦重要的业务流程。只需要遵守约定，针对数据编程即可获取扩展能力。

**冗余**：可以采用一对多的方式，一个生产者发布消息，可以被多个订阅topic的服务消费到，供多个毫无关联的业务使用。

**健壮性**：消息队列可以堆积请求，所以消费端业务即使短时间死掉，也不会影响主要业务的正常进行。

**异步通信**：很多时候，用户不想也不需要立即处理消息。消息队列提供了异步处理机制，允许用户把一个消息放入队列，但并不立即处理它。想向队列中放入多少消息就放多少，然后在需要的时候再去处理它们。

1. **kafka中的broker 是干什么的？**

broker 是消息的代理，Producers往Brokers里面的指定Topic中写消息，Consumers从Brokers里面拉取指定Topic的消息，然后进行业务处理，broker在中间起到一个代理保存消息的中转站。

1. **kafka中的 zookeeper 起到什么作用，可以不用zookeeper么**

zookeeper 是一个分布式的协调组件，早期版本的kafka用zk做meta信息存储，consumer的消费状态，group的管理以及 offset的值。考虑到zk本身的一些因素以及整个架构较大概率存在单点问题，新版本中逐渐弱化了zookeeper的作用。新的consumer使用了kafka内部的group coordination协议，也减少了对zookeeper的依赖，

但是broker依然依赖于ZK，zookeeper 在kafka中还用来选举controller 和 检测broker是否存活等等。

1. **kafka follower如何与leader同步数据？**

Kafka的复制机制既不是完全的同步复制，也不是单纯的异步复制。完全同步复制要求All Alive Follower都复制完，这条消息才会被认为commit，这种复制方式极大的影响了吞吐率。而异步复制方式下，Follower异步的从Leader复制数据，数据只要被Leader写入log就被认为已经commit，这种情况下，如果leader挂掉，会丢失数据，kafka使用ISR的方式很好的均衡了确保数据不丢失以及吞吐率。Follower可以批量的从Leader复制数据，而且Leader充分利用磁盘顺序读以及send file(zero copy)机制，这样极大的提高复制性能，内部批量写磁盘，大幅减少了Follower与Leader的消息量差。

1. **什么情况下一个 broker 会从 isr中踢出去？**

leader会维护一个与其基本保持同步的Replica列表，该列表称为ISR(in-sync Replica)，每个Partition都会有一个ISR，而且是由leader动态维护 ，如果一个follower比一个leader落后太多，或者超过一定时间未发起数据复制请求，则leader将其重ISR中移除。

1. **kafka 比其他消息系统快的原因？**

Cache Filesystem Cache PageCache缓存

顺序写 由于现代的操作系统提供了预读和写技术，磁盘的顺序写大多数情况下比随机写内存还要快。

Zero-copy 零拷⻉技术减少拷贝次数

Batching of Messages 批量量处理。合并小的请求，然后以流的方式进行交互，直顶网络上限。

Pull 拉模式 使用拉模式进行消息的获取消费，与消费端处理能力相符。

1. **kafka producer如何优化打入速度？**

* 增加线程
* 提高 batch.size
* 增加更多 producer 实例
* 增加 partition 数
* 设置 acks=-1 时，如果延迟增大：可以增大 num.replica.fetchers（follower 同步数据的线程数）来调解；
* 跨数据中心的传输：增加 socket 缓冲区设置以及 OS tcp 缓冲区设置。

1. kafka producer 打数据，ack 为 0， 1， -1 的时候代表啥， 设置 -1 的时候，什么情况下，leader 会认为一条消息 commit了？

1（默认） 数据发送到Kafka后，经过leader成功接收消息的的确认，就算是发送成功了。在这种情况下，如果leader宕机了，则会丢失数据。

0 生产者将数据发送出去就不管了，不去等待任何返回。这种情况下数据传输效率最高，但是数据可靠性确是最低的。

-1 producer需要等待ISR中的所有follower都确认接收到数据后才算一次发送完成，可靠性最高。

当ISR中所有Replica都向Leader发送ACK时，leader才commit，这时候producer才能认为一个请求中的消息都commit了。

1. kafka unclean 配置代表啥，会对 spark streaming 消费有什么影响？

unclean.leader.election.enable 为true的话，意味着非ISR集合的broker 也可以参与选举，这样有可能就会丢数据，spark streaming在消费过程中拿到的 end offset 会突然变小，导致 spark streaming job挂掉。如果unclean.leader.election.enable参数设置为true，就有可能发生数据丢失和数据不一致的情况，Kafka的可靠性就会降低；而如果unclean.leader.election.enable参数设置为false，Kafka的可用性就会降低。

1. 如果leader crash时，ISR为空怎么办？

kafka在Broker端提供了一个配置参数：unclean.leader.election,这个参数有两个值：

true（默认）：允许不同步副本成为leader，由于不同步副本的消息较为滞后，此时成为leader，可能会出现消息不一致的情况。

false：不允许不同步副本成为leader，此时如果发生ISR列表为空，会一直等待旧leader恢复，降低了可用性。

1. kafka的message格式是什么样的？

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之间多一个字节的数据：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

1. kafka中consumer group 是什么概念

同样是逻辑上的概念，是Kafka实现单播和广播两种消息模型的手段。同一个topic的数据，会广播给不同的group；同一个group中的worker，只有一个worker能拿到这个数据。换句话说，对于同一个topic，每个group都可以拿到同样的所有数据，但是数据进入group后只能被其中的一个worker消费。group内的worker可以使用多线程或多进程来实现，也可以将进程分散在多台机器上，worker的数量通常不超过partition的数量，且二者最好保持整数倍关系，因为Kafka在设计时假定了一个partition只能被一个worker消费（同一group内）。