# Kafka直播内容

## 内容回顾

## 知识点扩展

1.Kafka是怎么设计的？

Kafka 将消息以 topic 为单位进行管理。

向Kafka topic发布消息的程序成为producer。

订阅topics并消费消息的程序成为consumer。

Kafka以集群的方式运行，可以由一个或多个服务组成，每个服务叫做broker。

producers 通过网络将消息发送到Kafka集群，集群向消费者提供消息。

producer、consumer以及Kafka都可以横向扩展。

2.数据传输的事物定义有哪三种？

数据传输的事务定义通常有以下三种级别：

（1）最多一次: 消息不会被重复发送，最多被传输一次，但也有可能一次不传输

（2）最少一次: 消息不会被漏发送，最少被传输一次，但也有可能被重复传输.

（3）精确的一次（Exactly once）: 不会漏传输也不会重复传输,每个消息都传输被一次而仅仅被传输一次，这是大家所期望的

3.Kafka判断一个节点是否还活着有那两个条件？

（1）节点必须可以维护和ZooKeeper的连接，Zookeeper通过心跳机制检查每个节点的连接。

（2）如果节点是个 follower，必须能及时的同步leader的写操作，延时不能太久。

4.producer是否直接将数据发送到broker的leader(主节点)？

producer直接将数据发送到broker的leader(主节点)，不需要在多个节点进行分发，为了帮助producer做到这点，所有的Kafka节点都可以及时的告知：哪些节点是活动的，目标topic 目标分区的leader在哪。这样producer就可以直接将消息发送到目的地了。

这些元数据保存于zookeeper。

5、Kafa consumer 是否可以消费指定分区消息？

Kafa consumer消费消息时，向broker发出"fetch"请求去消费特定分区的消息，consumer指定消息在日志中的偏移量（offset），就可以消费从这个位置开始的消息，customer 拥有offset的控制权，可以向后回滚去重新消费之前的消息。

6、Kafka消息是采用Pull模式，还是Push模式？

Kafka最初考虑的问题是，customer应该从brokes拉取消息还是brokers将消息推送到consumer，也就是pull还push。在这方面，Kafka 遵循了一种大部分消息系统共同的传统的设计：producer将消息推送到broker，consumer从broker拉取消息

一些消息系统比如 Scribe 和 Apache Flume 采用了push模式，将消息推送到下游的consumer。这样做有好处也有坏处：由broker决定消息推送的速率，对于不同消费速率的consumer就不太好处理了。消息系统都致力于让consumer以最大的速率最快速的消费消息，但不幸的是，push模式下，当broker推送的速率远大于consumer消费的速率时，consumer恐怕就要崩溃了。最终Kafka还是选取了传统的pull模式。

Pull模式的另外一个好处是consumer可以自主决定是否批量的从broker拉取数据。Push模式必须在不知道下游consumer消费能力和消费策略的情况下决定是立即推送每条消息还是缓存之后批量推送。如果为了避免consumer崩溃而采用较低的推送速率，将可能导致一次只推送较少的消息而造成浪费。Pull模式下，consumer就可以根据自己的消费能力去决定这些策略

Pull 有个缺点是，如果broker没有可供消费的消息，将导致consumer不断在循环中轮询，直到新消息到达。为了避免这点，Kafka有个参数可以让consumer阻塞知道新消息到达(当然也可以阻塞知道消息的数量达到某个特定的量这样就可以批量发。

7.Kafka 存储在硬盘上的消息格式是什么？

消息由一个固定长度的头部和可变长度的字节数组组成。头部包含了一个版本号和 CRC32校验码。

消息长度: 4 bytes (value: 1+4+n)

版本号: 1 byte

CRC 校验码: 4 bytes

具体的消息: n bytes

8.Kafka高效文件存储设计特点：

(1).Kafka把topic中一个parition大文件分成多个小文件段，通过多个小文件段，就容易定期清除或删除已经消费完文件，减少磁盘占用。

(2).通过索引信息可以快速定位message和确定response的最大大小。

(3).通过index元数据全部映射到memory，可以避免segment file的IO磁盘操作。

(4).通过索引文件稀疏存储，可以大幅降低index文件元数据占用空间大小。

9.Kafka与传统消息系统之间有三个关键区别

(1).Kafka 持久化日志，这些日志可以被重复读取和无限期保留，默认7天保留。

(2).Kafka 是一个分布式系统：它以集群的方式运行，可以灵活伸缩，在内部通过复制数据提升容错能力和高可用性

(3).Kafka 支持实时的流式处理

10.Kafka创建Topic时如何将分区放置到不同的Broker中

副本因子不能大于Broker的个数；

第一个分区（编号为0）的第一个副本放置位置是随机从brokerList选择的；

其他分区的第一个副本放置位置相对于第0个分区依次往后移。也就是如果我们有5个Broker，5个分区，假设第一个分区放在第四个Broker上，那么第二个分区将会放在第五个 Broker上；第三个分区将会放在第一个Broker上；第四个分区将会放在第二个Broker 上，依次类推；

剩余的副本相对于第一个副本放置位置其实是由nextReplicaShift决定的，而这个数也是随机产生的

11.Kafka新建的分区会在哪个目录下创建

在启动 Kafka 集群之前，我们需要配置好log.dirs参数，其值是Kafka数据的存放目录，这个参数可以配置多个目录，目录之间使用逗号分隔，通常这些目录是分布在不同的磁盘上用于提高读写性能。

当然也可以配置log.dir参数，含义一样。只需要设置其中一个即可。

如果log.dirs参数只配置了一个目录，那么分配到各个Broker上的分区肯定只能在这个目录下创建文件夹用于存放数据。

如果log.dirs参数配置了多个目录，那么Kafka会在哪个文件夹中创建分区目录呢？

答案是：Kafka会在含有分区目录最少的文件夹中创建新的分区目录，分区目录名为Topic名+分区ID。注意，是分区文件夹总数最少的目录，而不是磁盘使用量最少的目录！也就是说，如果你给log.dirs参数新增了一个新的磁盘，新的分区目录肯定是先在这个新的磁盘上创建直到这个新的磁盘目录拥有的分区目录不是最少为止。

12.partition的数据如何保存到硬盘

topic中的多个partition以文件夹的形式保存到broker，每个分区序号从0递增，且消息有序

Partition文件下有多个segment（xxx.index，xxx.log）

segment文件里的大小和配置文件大小一致可以根据要求修改默认为1g

如果大小大于1g时，以第一条消息的偏移量命名，滚动新segment文件。

13.kafka 的 ack 机制

request.required.acks 有三个值 0 1 -1

0:生产者不会等待broker的ack，这个延迟最低但是存储的保证最弱，当server挂掉的时候就会丢数据。

1：服务端会等待ack值leader副本确认接收到消息后发送ack；但是如果leader挂掉后他不确保是否复制完成新leader也会导致数据丢失

-1：同样在1的基础上，服务端会等所有的ISR follower的副本受到数据后才会收到leader发出的ack，这样数据不会丢失。也可以指定最小确认副本的个数。

14.Kafka 的消费者如何消费数据

消费者每次消费数据的时候，消费者都会记录消费的物理偏移量（offset）的位置，等到下次消费时，他会接着上次位置继续消费。偏移量存储于Kafka特定主题：\_\_consumer\_offsets中。

15.消费者负载均衡策略

一个消费者组中的一个分片对应一个消费者成员，他能保证每个消费者成员都能访问，如果组中成员太多会有空闲的成员。

16.数据有序

一个消费者组里它的内部各个消费者是有序的

消费组与消费组之间是无序的

17.kafaka 生产数据时数据的分组策略

1. 生产者如果不指定消息的key，则客户端通过轮询的方式选择该消息所属的分区。

2. 生产者如果指定了key，则以key的hashCode对分区数量取模决定该消息所属的分区。

3. 生产者也可以直接指定该消息所属的分区，此时忽略消息的key。