# 1.什么是Kafka

Kafka是一个分布式数据流处理平台，可以实时发布、订阅、存储和处理流数据。

# 2.Kafka相对传统技术有什么优势

传统的消息传递方法包括两种：排队，在队列中，一组用户可以从服务器中读取消息，每条消息都发送给其中一个人。发布-订阅：在这个模型中，消息被广播给所有的用户。

相对于传统技术的优势：

1. 快速：单一的kafka代理可以存储成千上万的客户端，每秒处理数兆字节的读写操作。
2. 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。
3. 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。
4. 设计：它提供了容错保证和持久性。

# 3.Kafka中的broker

一个单独的kafka server就是一个broker，broker主要工作就是接收生产者发过来的消息，分配offset，之后保存到磁盘中。同时，接收消费者、其他broker的请求，根据请求类型进行相应的处理并返回响应，在一般的生产环境中，一个broker独占一台物理服务器。

Broker的意义在于：

1. 接收producer发过来的数据，并且将它持久化，同时提供给consumer去订阅。
2. 组成kafka集群节点，之间没有主从关系，依赖zookeeper来协调，broker负责消息的读取和存储，一个broker可以管理多个partition。

# Kafka与Zookeeper

Zookeeper是一个开放源码的、高性能的协调服务，它用于kafka的分布式应用。

作用：协调kafka broker，存储原数据：consumer的offset+broker信息+topic信息+partition信息。

用户不能越过zookeeper直接联系kafka broker。一旦zookeeper停止了工作，它就不能服务客户端请求。

1.Zookeeper主要用于在集群中的不同节点之间进行通信。

2.在kafka中，它被用于提交偏移量，因此如果节点在任何情况下都失败了，它都可以从之前提交的偏移量中获取。

3.除此之外，它还执行其他活动，如：leader检测、分布式同步、配置管理、识别新节点何时离开或连接、集群、节点实时状态等等。

# 4.如何提高远程用户的吞吐量

如果用户位于与broker不同的数据中心，则可能需要调优套接字缓冲区大小，以对长网络延迟进行摊销。

# 5.在数据制作过程中，如何能从kafka得到准确的信息

1.在数据中，为了准确的获得kafka的信息，你必须遵循两件事：在数据消耗期间避免重复，在数据生产过程中避免重复。

2.在这里有两种方法，可以在数据生成时准确的获得一个语义：

每个分区使用一个单独的读入器，每当你发现一个网络错误，检查该分区中的最后一条信息，以查看您的最后一次写入是否成功。

在消息中包含一个主键，并在用户中进行反复制。

# 6.Kafka主要特征

1.kafka具有近乎实时性的消息处理能力，面对海量数据，高效的存储消息和查询消息。Kafka将消息保存在磁盘中，以顺序读写的方式访问磁盘，从而避免了随机读写磁盘导致的性能瓶颈。

2.kafka支持批量读写数据，并且对消息批量压缩，提高了网络利用率和压缩效率。

3.kafka支持消息分区，每个分区中的消息保证顺序传输，而分区之间可以并发操作，提高了kafka的并发能力。

4.kafka支持在线增加分区，支持在线水平扩展。

5.kafka支持为每个分区创建多个副本，其中只会有一个leader副本负责读写，其他副本只负责与leader副本同步，这种方式提高了数据的容灾能力，kafka会将leader副本均匀的分布在集群中的服务器上，实现性能最大化。

# 7.Kafka的应用场景

1.日志收集：一个公司可以用kafka手机各种服务的log，通过kafka以统一接口服务的方式开放给各种consumer,例如Hadoop、Hbase、Solr等。

2.消息系统：解耦和生产着、消费消息等。

3.用户活动跟踪：kafka经常用来记录web用户或者app用户的各种活动，如浏览网页、搜索、点击等活动，这些活动信息被各个服务器发布到kafka的topic中，然后订阅者通过订阅这些topic来做实时监控信息，或者装载到Hadoop、数据仓库中做离线分析和挖掘。

4.运营指标：kafka也经常用来记录运营监控数据。包括收集各种分布式应用的数据，生产各种操作的集中反馈，比如报警和报告。

5.流式处理：比如spark streaming和strom

6.事件源

# 8.Kafka分区的作用

Kafka的每个topic都可以分为多个partition，每个partition都有多个replica(副本)，每个分区中的消息是不同的，提高了并发读写的能力，而同一个分区的不同副本中保存的是相同的信息。副本之间是一主多从关系，其中leader副本处理读写请求，follower副本只与leader副本进行消息同步，当leader副本出现故障时，则从follower副本中重新选举leader副本对外提供服务。这样，通过提高分区的数量，就可以实现水平扩展，通过提高副本数量，就可以提高容灾能力。

# 9.consumer水平扩展如何实现

Kafka支持consumer水平扩展，可以让多个consumer加入一个consumer group，在一个consumer group中，每个分区只能分配给一个consumer，当kafka服务端增加分区数量进行水平扩展后，可以向consumer group中增加新的consumer来提高整个consumer group的消费能力，当consumer group中的一个consumer出现故障下线时，会通过reblance操作下线consumer，将它负责处理的分区分配给其他consumer。

# 10.消息的顺序

Kafka保证一个partition内消息是有序的，但是并不保证多个partition之间的数据是有序的，每个topic可以划分成多个分区，同一个topic下不同分区包含的消息是不同的，每个消息在被添加到分区时，都会被分配一个offset，它是此消息在分区中的唯一编号，kafka通过offset保证消息在分区内的顺序，offset顺序不能跨区，即kafka只保证在同一个分区内的消息是有序的。

# 11.ISR集合是什么

1.ISR(In-Sync Replica)集合表示的是目前可用并且消息量与leader差不多的副本集合，这是整个副本集合的一个子集。

2.ISR集合的副本必须满足：副本所在节点都必须保持着与zookeeper的连接；副本最后一条消息的offset与leader副本最后一条消息的offset之间的差值不能超出指定的阈值。

3.每个分区的leader副本都会维护此分区ISR集合，写请求首先由leader副本处理，只会follower副本会从leader副本上拉取写入的消息，这个过程会有一定的延迟，导致follower副本中保存的消息略少于leader副本，只要未超出阈值是可以容忍的。

# 12.Kafka设计时是什么样的

1.Kafka将消息以topic为单位进行归纳。

2.将向kafka topic发布消息的程序成为producer。

3.将从kafka消费消息的程序成为consumer。

4.kafka以集群的方式运行，可以由一个或多个服务组成，每个服务称为一个broker。

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

# 13.Kafka怎么判断一个节点还活着

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，他必须能及时的同步leader的写操作，延时不能太久。

# 14.Kafka消息采用Pull模式，还是Push模式

1.Kafka最初考虑的问题是，consumer应该从brokers拉取消息还是brokers将消息推送到consumer，也就是pull还是push。在这方面，kafka遵循了一中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。

2.一些消息系统比如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消费的速率的时候，consumer恐怕就要崩溃了。最终kafka还是选取了传统的pull模式。

3.pull模式的另一种好处是consumer可以自主决定是否批量的从broker拉取消息。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崩溃而采用了较低的推送速率，将可能导致一次只能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。

4.pull模式有个缺点是，如果broker没有可供消费的消息，将导致consumer不断在循环中轮询，知道新消息到达。为了避免这点，kafka有个参数可以让consumer知道新消息到达。

# 15.Kafka存储在硬盘上的消息格式是什么

消息有一个固定长度的头部和可变长度的字节数组组成。头部包含一个版本号和CRC32检验码

消息长度：4 bytes(value:1+4+n)

版本号：1 bytes

CRC校验码：4 bytes

具体的消息：n bytes

# 16.Kafka高效存储文件特点

1.kafka把topic中一个partition的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消费完的文件，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，可以大幅度降低index元数据占用空间大小。

# 17.Kafka与传统消息系统之间的三个区别

1.kafka持久化日志，这些日志可以被重复读取和无限期保留。

2.kafka是一个分布式系统：它已集群的方式运行，可以灵活伸缩，在内部同过复制数据提高容错能力和高可用性。

3.kafka支持实时的流式处理。

# 14.Kafka创建Topic时如何将分区副本放在不同的Broker中

1.副本因子不能大于broker的个数；

2.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。

3.其它分区的第一个副本放置位置相对于第0个分区一次向后移。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4个broker上，那么第二个分区将会放在第5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；

4.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。

# 15.Kafka新建的分区会在哪个目录下创建

1.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，目录之间用逗号分隔，通常这些目录是分布在不同的磁盘上的同于提高读写性能。

2.当然我们也可以配置log.dir参数，含义一样。只需要设置其中一个即可。

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

4.但是如果log.dirs配置了多个目录，那么kafka会在哪个文件夹中创建目录呢？答案是：kafka会在含有分区目录最少的文件中创建新的分区目录，分区目录名为topoc名+分区ID。注意，是分区文件夹总数最少的目录，而不是在磁盘使用率最少的目录！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**。**

# 16.Partition的数据是如何保存到硬盘

1.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。

2.partition文件下有多个segment(xxx.index，xxx.log)。

3.segment文件里的大小和配置文件大小一致根艺根据需要修改默认为1g。

4.如果大小大于1g，会滚动一个新的segment并且以上一个segment最后一条消息的偏移量命名。

# 17.kafka的ack机制

1.reuquest.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挂掉时会丢失数据。

1：服务器会等待ack值，leade副本确认接收到消息后发送ack，但是如果leader挂掉后，他不保证是否复制完，新leader也会导致数据丢失。

-1：在同1的基础上，服务端会等到所有的follower的副本收到数据后，才会收到leader的ack,这样数据不会丢失。

# 18.Kafka消费者如何消费数据

1.消费者每次消费数据的时候，消费者会记录消费的物理偏移量(offset)的位置。

2.等到下次消费的时候，他会接着上次位置继续消费。

# 19.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，如果族中成员太多会有空闲的成员。

# 20.数据有序

一个消费者组里它的内部是有序的。

消费者组和消费者组之间是无序的。

# 21.Kafka生产数据时的数据分组策略

1.生产者决定数据产生到集群中的那个partition。

2.每一条消息都是以(key,value)格式。

3.key是生产者发送数据传入

4.所以生产者key决定了数据产生到集群的哪个partition。

# 22.Kafka的Leader

1.Kafka的leader选举是通过在zookeeper上创建/controller临时节点来实现leader选举，并在该节点上写入当前broker的信息{version:1,brokerid:1,timestamp:1512018424988}，利用zookeeper的强制一致性特性，一个节点只能被一个客户端创建成功，创建成功的broker即为leader，即先到先得原则，leader也就是集群中的领导者，负责集群中所有大小事务。

当leader与zookeeper失去连接时，临时节点会被删除，而其他broker会监听到该节点的变化，当节点删除时，其他broker会收到通知，重新发起leader选举。

2.kafka的leader是什么？

首先kafka会将接收到的消息分区，每个主题的消息有不同的分区。这样一方面消息的存储就不会受到单一服务器存储空间大小的限制，另一方面消息的处理也可以在多个服务器上并行。

其次，为了保证高可用性，每个分区都会有一定数量的副本，这样如果有部分服务器不可用，副本所在服务器就会接替上来保证应用的持续性。

但是，为了保证较高的处理效率，消息的读写都是在固定的一个副本上完成的。这个副本就是所谓的leader，其他副本则是follower。而follower则会定期的到leader上同步数据。