1. kafka：

1.消息分类按不同类别,分成不同的Topic,Topic又拆分成多个partition,每个partition均衡分散到不同的服务器(提高并发访问的能力)

2.消费者按顺序从partition中读取,不支持随机读取数据,但可通过改变保存到zookeeper中的offset位置实现从任意位置开始读取

3.服务器消息定时清除(不管有没有消费)

4.每个partition还可以设置备份到其他服务器上的个数以保证数据的可用性。通过Leader,Follower方式

5.zookeeper保存kafka服务器和客户端的所有状态信息.(确保实际的客户端和服务器轻量级)

6.在kafka中,一个partition中的消息只会被group中的一个consumer消费;每个group中consumer消息消费互相独立;我们可以认为一个group是一个"订阅"者,一个Topic中的每个partions,只会被一个"订阅者"中的一个consumer消费,不过一个consumer可以消费多个partitions中的消息

7.如果所有的consumer都具有相同的group,这种情况和queue模式很像;消息将会在consumers之间负载均衡.

8.如果所有的consumer都具有不同的group,那这就是"发布-订阅";消息将会广播给所有的消费者.

9.持久性,当收到的消息时先buffer起来,等到了一定的阀值再写入磁盘文件,减少磁盘IO.在一定程度上依赖OS的文件系统(对文件系统本身优化几乎不可能)

10.除了磁盘IO,还应考虑网络IO, 批量对消息发送和接收,并对消息进行压缩。

11.在JMS实现中,Topic模型基于push方式,即broker将消息推送给consumer端.不过在kafka中,采用了pull方式,即consumer在和broker建立连接之后,主动去pull(或者说fetch)消息;这种模式有些优点,首先consumer端可以根据自己的消费能力适时的去fetch消息并处理,且可以控制消息消费的进度(offset);此外,消费者可以良好的控制消息消费的数量,batch fetch.

12.kafka无需记录消息是否接收成功,是否要重新发送等,所以kafka的producer是非常轻量级的,consumer端也只需要将fetch后的offset位置注册到zookeeper,所以也是非常轻量级的.

--kafka使用场景

对于一些常规的消息系统,kafka是个不错的选择;partitons/replication和容错,可以使kafka具有良好的扩展性和性能优势.

不过到目前为止,我们应该很清楚认识到,kafka并没有提供JMS中的"事务

性""消息传输担保(消息确认机制)""消息分组"等企业级特性;

kafka只能使用作为"常规"的消息系统,在一定程度上,尚未确保消息的发送与接收绝对可靠(比如,消息重发,消息发送丢失等)

kafka的特性决定它非常适合作为"日志收集中心";application可以将操作日志"批量""异步"的发送到kafka集群中,

而不是保存在本地或者DB中;kafka可以批量提交消息/压缩消息等,这对producer端而言,几乎感觉不到性能的开支.

consumer端采用批量fetch方式,此时consumer端也可以使hadoop等其他系统化的存储和分析系统

2. kafka和RabbitMQ区别:

1、RabbitMQ,遵循AMQP协议,由内在高并发的erlanng语言开发,用在实时的对可靠性要求比较高的消息传递上(有消息确认机制)。

2、kafka是Linkedin于2010年12月份开源的消息发布订阅系统,它主要用于处理活跃的流式数据,大数据量的数据处理上(无消息确认机制,但吞吐量高),kafka用zk做集群负载均衡。

3. 项目中哪里用到了kafka,它有什么特性?

a. 场景:

i. 大数据部门流数据处理;

ii. elk;

b. 特性:

i. 它被设计为一个分布式系统,易于向外扩展;

ii. 它同时为发布和订阅提供高吞吐量;

iii. 它支持多订阅者,当失败时能自动平衡消费者;

iv. 它将消息持久化到磁盘,因此可用于批量消费,例如ETL,以及实时应用程序。

4. kafka数据分区和消费者的关系,kafka的数据offset读取流程,kafka内部如何保证顺序,结合外部组件如何保证消费者的顺序?

1、kafka数据分区和消费者的关系:1个partition只能被同组的一个consumer消费,同组的consumer则起到均衡效果

2、kafka的数据offset读取流程

1.连接ZK集群,从ZK中拿到对应topic的partition信息和partition的Leader的相关信息

2.连接到对应Leader对应的broker

3.consumer将自己保存的offset发送给Leader

4.Leader根据offset等信息定位到segment（索引文件和日志文件）

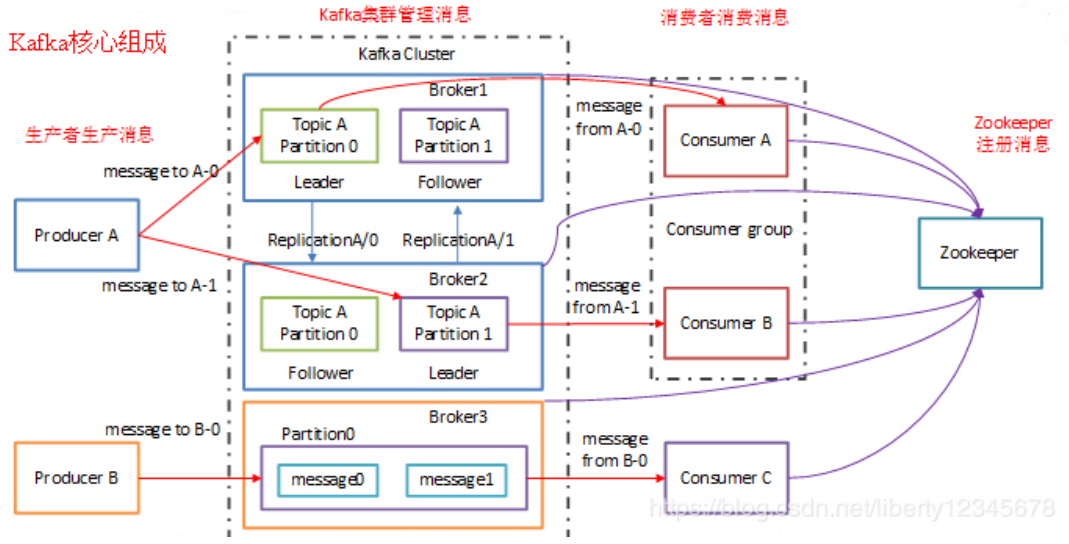
5.根据索引文件中的内容，定位到日志文件中该偏移量对应的开始位置读取相应长度的数据并返回给consumer

3、kafka内部如何保证顺序：

kafka只能保证partition内是有序的，但是partition间的有序是没办法的。爱奇艺的搜索架构，是从业务上把需要有序的打到同一个partition。

5. kafka工作流程?

Apache Kafka是分布式发布-订阅消息系统。它最初由LinkedIn公司开发，之后成为Apache项目的一部分。Kafka是一种快速、可扩展的、设计内在就是分布式的，分区的和可复制的提交日志服务。



a. 生产者定期向主题发送消息。

b. Kafka代理存储为该特定主题配置的分区中的所有消息。 它确保消息在分区之间平等共享。 如果生产者发送两个消息并且有两个分区，Kafka将在第一分区中存储一个消息，在第二分区中存储第二消息。

c. 消费者订阅特定主题。

d. 一旦消费者订阅主题，Kafka将向消费者提供主题的当前偏移，并且还将偏移保存在Zookeeper系综中。

e. 消费者将定期请求Kafka(如100 Ms)新消息。

f. 一旦Kafka收到来自生产者的消息，它将这些消息转发给消费者。

g. 消费者将收到消息并进行处理。

h. 一旦消息被处理，消费者将向Kafka代理发送确认。

i. 一旦Kafka收到确认，它将偏移更改为新值，并在Zookeeper中更新它。 由于偏移在Zookeeper中维护，消费者可以正确地读取下一封邮件，即使在服务器暴力期间。

j. 以上流程将重复，直到消费者停止请求。

k. 消费者可以随时回退/跳到所需的主题偏移量，并阅读所有后续消息。