

Kafka

kafka是什么？ kafka仅仅是属于消息 中间件吗？

kafka在设计之初的时候 开发人员在除了消息中间件以外，还想吧kafka设计为一个能够存储数据的系统，有点像常见的非关系型数据库，比如说NoSql等。除此之外 还希望kafka能支持持续变化，不断增长的数据流，可以发布和订阅数据流，还可以对于这些数据进行保存

也就是说kafka的本质 是一个数据存储平台，流平台， 只是他在做消息发布，消息消费的时候我们可以把他当做消息中间件来用。

而且kafka在设计之初就是采用分布式架构设计的， 基于集群的方式工作，且可以自由伸缩，所以 kafka构建集群非常简单

基本概念：

- Broker : 和AMQP里协议的概念一样， 就是消息中间件所在的服务器
- Topic(主题): 每条发布到Kafka集群的消息都有一个类别，这个类别被称为Topic。（物理上不同Topic的消息分开存储，逻辑上一个Topic的消息虽然保存于一个或多个broker上但用户只需指定消息的Topic即可生产或消费数据而不必关心数据存于何处）
- Partition(分区): Partition是物理上的概念，体现在磁盘上面，每个Topic包含一个或多个Partition.
- Producer : 负责发布消息到Kafka broker
- Consumer : 消息消费者，向Kafka broker读取消息的客户端。
- Consumer Group（消费者群组）：每个Consumer属于一个特定的Consumer Group（可为每个Consumer指定group name，若不指定group name则属于默认group）。
- offset 偏移量： 是kafka用来确定消息是否被消费过的标识，在kafka内部体现就是一个递增的数字

kafka消息发送的时候,考虑到性能 可以采用打包方式发送，也就是说 传统的消息是一条一条发送，现在可以先把需要发送的消息缓存在客户端，等到达一定数值时，再一起打包发送，而且还可以对发送的数据进行压缩处理，减少在数据传输时的开销

kafka优缺点

优点: 基于磁盘的数据存储 高伸缩性 高性能 应用场景：收集指标和日志 提交日志 流处理

缺点:

运维难度大 偶尔有数据混乱的情况 对zookeeper强依赖 多副本模式下对带宽有一定要求

kafka安装&启动

kafka安装的话，直接 从官网下载压缩包下来解压就可以了

注意的是， 启动kafka要先启动zookeeper kafka默认自带了zookeeper 可以启动他自带的 也可以自己另外使用

启动kafka需要执行 kafka-server-start.bat 文件 然后 需要传入一个路径参数 就是你server.config文件的地址 一般情况下传入../../config/server.properties 即可

刚刚提到的zookeeper kafka默认的zookeeper 启动的话启动zookeeper-server-start.bat文件即可 同样 要传入路径参数： ../../config/zookeeper.properties

server 参数解释：

log.dirs: 日志文件存储地址， 可以设置多个

num.recovery.threads.per.data.dir: 用来读取日志文件的线程数量， 对应每一个log.dirs 若此参数为2 log.dirs 为2个目录 那么就会有4个线程来读取

auto.create.topics.enable:是否自动创建topic

num.partitions: 创建topic的时候自动创建多少个分区 (可以在创建topic的时候手动指定)

log.retention.hours: 日志文件保留时间 超时即删除

log.retention.bytes: 日志文件最大大小

log.segment.bytes: 当日志文件达到一定大小时， 开辟新的文件来存储(分片存储)

log.segment.ms: 同上 只是当达到一定时间时 开辟新的文件

message.max.bytes: broker能接收的最大消息大小(单条) 默认1M

kafka基本管理操作命令：

##列出所有主题 kafka-topics.bat --zookeeper localhost:2181/kafka --list

##列出所有主题的详细信息 kafka-topics.bat --zookeeper localhost:2181/kafka --describe

##创建主题 主题名 my-topic, 1副本, 8分区 kafka-topics.bat --zookeeper localhost:2181/kafka --create --replication-factor 1 --partitions 8 --topic my-topic

##增加分区， 注意： 分区无法被删除 kafka-topics.bat --zookeeper localhost:2181/kafka --alter --topic my-topic --partitions 16

##删除主题 kafka-topics.bat --zookeeper localhost:2181/kafka --delete --topic my-topic

##列出消费者群组（仅Linux） kafka-topics.sh --new-consumer --bootstrap-server localhost:9092/kafka --list

##列出消费者群组详细信息（仅Linux） kafka-topics.sh --new-consumer --bootstrap-server
localhost:9092/kafka --describe --group 群组名
