Kafka消息队列或消息中间件：用于系统之间的解耦合、峰值压力缓存和异步通信，生产者消费者模式

Kafka架构：

producer：消息生产者

consumer：消息消费者

broker：Kafka集群的Server节点，负责处理消息读写请求存储消息，broker之间无主从关系

topic：消息队列/分类

ZooKeeper：协调Kafka broker，存储原数据：consumer的offset、broker、topic和partition信息

一个topic分成多个partition，partition内部消息强有序，每个消息都有一个offset序号，broker节点可以管理多个partition，一个partition只能交由一个broker管理，消息不存储在内存中直接写入文件，消息默认时间策略一周删除，producer默认使用基于Hash的partition策略写消息（可以更改策略）到指定的partition中

consumer通过ZooKeeper维护消费到哪个offset消息，consumer都会有对应的group，group是queue消费模式（group内每个consumer消费不同的partition，一个消息在group中只能消费一次，每个group之间互不影响）

Kafka消息系统的特点：partition内部是FIFO，高性能（单节点支持上千个客户端，百MB/s吞吐量），持久性（消息直接持久化到硬盘且性能好），分布式（数据副本冗余、流量负载均衡、可扩展）

Kafka零拷贝Non-Zero Copy方式不需要将数据拷贝到用户空间直接在内核空间存储数据

Kafka的partition的leader宕机，partition会根据副本优先寻找新leader，宕机的leader重启会接管原partition

Kafka集群安装部署：

解压Kafka安装包

修改Kafka解压目录/config/server.properties配置文件：

broker.id=0 Kafka集群的broker标识数值从0开始

log.dirs=/tmp/kafka-logs Kafka数据存储目录，指定到永久目录，暂存目录会被清理数据

zookeeper.connect=hadoop1:2181,hadoop2:2181,hadoop3:2181 ZooKeeper集群的IP地址端口号

启动ZooKeeper集群后使用Kafka安装目录/bin/kafka-server-start.sh ../config/server.properties启动Kafka，jps会有Kafka进程，由于Kafka无主从，所以集群需要每个Kafka节点启动

一般使用脚本后台启动Kafka：

nohup bin/kafka-server-start.sh config/server.properties >kafka.log 2>&1 & 使用nohup命令后台启动Kafka将日志输出到kafka.log

Kafka安装目录/bin/kafka-topics.sh --zookeeper hadoop1:2181,hadoop2:2181,hadoop3:2181 --create --topic ceshi --partition 3 --replication-factor 3 创建topic

Kafka安装目录/bin /kafka-topics.sh --zookeeper hadoop1:2181,hadoop2:2181,hadoop3:2181 –list 查看topic列表

Kafka安装目录/bin/kafka-run-class.sh kafka.tools.GetOffsetShell --broker-list hadoop1:9092 --topic --time -1/-2 查看topic每个分区当前最大消息的位移/最小位移（生成的消息位移/消费的消息位移，最大位移与最小位移之差为集群中topic的消息总数）

Kafka安装目录/bin/kafka-console-producer.sh --topic ceshi --broker-list hadoop1:9092,hadoop2:9092,hadoop3:9092 向指定的topic中生成消息

Kafka安装目录/bin/kafka-console-consumer.sh --zookeeper hadoop1:2181,hadoop2:2181,hadoop3:2181 --topic ceshi 消费指定topic

Kafka删除topic：

需要先进入Kafka所有节点的存储数据目录rm -rf ceshi-\* 删除指定数据文件

./kafka-topics.sh --zookeeper hadoop1:2181,hadoop2:2181,hadoop3:2181 --delete --topic ceshi 标记删除topic，还可以进行读写消息，默认一周后才删除

zkCli.sh 进入ZooKeeper数据目录

rmr /brokers/topics/ceshi 删除ZooKeeper中topic原数据

rmr /admin/delete\_topics/ceshi 删除ZooKeeper中topic标记删除的信息后才会立即删除topic

Kafka两种消费API：

High Level Consumer API：消费者通过ZooKeeper维护消费者offset，数据可能会丢失

Simple Consumer API：消费者自己管理offset，开启WAL（Write Ahead Log）预写日志机制需要设置checkpoint，一般将接收的数据降级存储，将接收的Kafka消息备份到HDFS中再去更新ZooKeeper的offset，会加大Application的处理时间

Simple Consumer API：消费者自己管理offset，开启WAL（Write Ahead Log）预写日志机制，将接收的Kafka消息备份到HDFS中再去更新ZooKeeper的offset，会加大Application的处理时间

Receiver模式的并行度由spark.streaming.blockInterval=200ms决定，接收的数据默认每个200ms持久化生成一个block块，一般设置不小于50ms

Direct模式使用Simple Consumer API直接从Kafka读取数据，生成DStream中RDD的并行度与读取的topic的partition的个数一致，Spark可以不使用ZooKeeper管理offset自己进行管理，也可以程序管理offset，将offset信息存储到ZooKeeper或者MySQL中