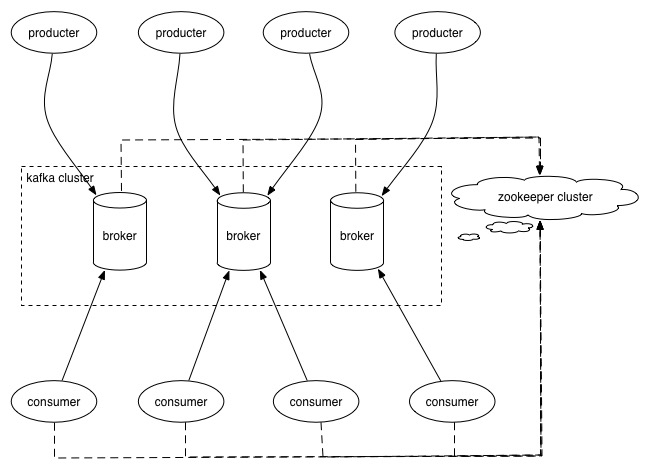
1. Kafka 产生的背景

分布式消息和订阅系统。高性能、高吞吐量（每秒请求处理几十万条数据）。Scala语言编写，非JMS规范。

内置分区、实现集群（容错、数据复制）

1. Kafka的架构



多个broker（消息提供者），采用zookeeper集群实现broker的master-slave集群。Consumer从broker中pull（拉取）消息。

Topic：主题。

Partition：topic中的分区。

Group：消费端的所属。

1. Kafka安装部署及集群部署

安装部署

1. 下载kafka解压：

tar -xzf kafka\_2.11-2.1.0.tgz

1. 启动kafka，后台守护进程运行”-daemon”：

bin/kafka-server-start.sh config/server.properties

1. 创建topic：

bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --topic test

bin/kafka-topics.sh --list --zookeeper localhost:2181

1. 发送消息：

bin/kafka-console-producer.sh --broker-list localhost:9092 --topic test

1. 开启消费者：

bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic test --from-beginning

集群部署：

1. 修改各个节点中/config/server.properties中的如下参数：

|  |
| --- |
| broker.id=0 //集群中唯一  zookeeper.connect=localhost:2181 //zookeeper连接地址，集群下地址逗号分割  listeners=PLAINTEXT://ip:9092 //ip为本机IP地址 |

1. 在zookeeper客户端上查看相关节点如：/brokers/ids、/controller
2. Kafka的基本操作

生产者参数：

1. ProducerConfig.ACKS\_CONFIG = "-1"

0：消息发送给broker后，不需要确认（性能高，但是会出现数据丢失）。

1：只需要获得kafka集群中leader节点的确认即可返回。（leader/follower）

all（-1）：需要ISR中的所有副本进行确认（需要集群中的所有节点确认）。性能低

但数据最安全， 也可能会出现数据丢失（但ISR中的副本只有一个）。

1. ProducerConfig.BATCH\_SIZE\_CONFIG(batch.size)

Producer对于同一个分区来说，会按照batch.size的大小进行统一收集进行批量发送。默认16KB。

1. ProducerConfig.LINGER\_MS\_CONFIG(linger.ms)

每次发送间隔。

与(batch.size)配合使用在指定毫秒时间间隔中延迟(delay)，收集发送的消息，进行批量发送到broker。默认为0。

如batch.size 和 linger.ms都配置只要满足其一就会批量发送。

1. ProducerConfig. MAX\_REQUEST\_SIZE\_CONFIG (max.request.size)

发送请求最大长度。默认为：1MB。设置合理提高发送和接送性能。

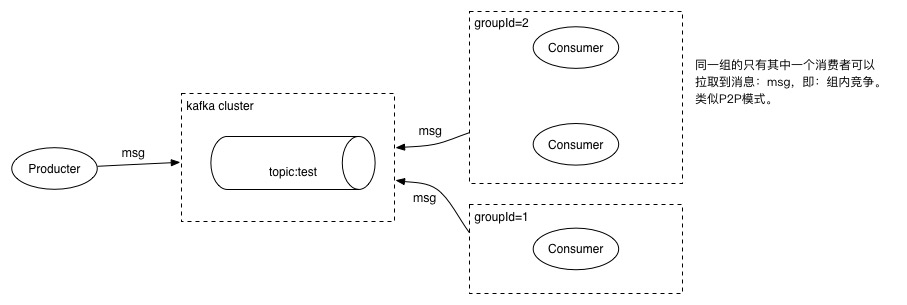
消息同步和异步发送：

Kafka-1.0版本以后默认为异步发送消息。

Kafka持久化消息，即磁盘中。

接收者参数：

1. ConsumerConfig.GROUP\_ID\_CONFIG(group.id)



不同的groupId分组的消费者都可以接受到消息，但同一组中的消费者只有其中一个可以接受到消息。

1. ConsumerConfig. AUTO\_OFFSET\_RESET\_CONFIG

对于新的groupId来说，如果设置为：earliest，那么它会从最早的消息开始消费。对于已消费的groupId来说则从已消费的最大offset开始消费消息。

earliest：对于新的groupId重置offset。

none：如果新的groupId无offset(即没有消费过消息)抛出异常。

latest：对于新的groupId取最近的消息（即：已消费并且提交的最大的offset）。

1. ConsumerConfig.ENABLE\_AUTO\_COMMIT\_CONFIG

自动提交(ACK)。值为：true/false。

ConsumerConfig.AUTO\_COMMIT\_INTERVAL\_MS\_CONFIG

(auto.commit.interval.ms)

自动提交间隔毫秒数。批量确认。

1. ConsumerConfig.MAX\_POLL\_RECORDS\_CONFIG(max.poll.records)

每次poll拉取的消息最大数量。

1. Kafka的应用

行为跟踪、日志搜集

1. Spring整合Kafka