案例：<https://gitee.com/zx19890628/spring-boot-example/tree/master/lab_091_mq_kafka>

教程：<https://mp.weixin.qq.com/s/una4HQgGpjVd51e_RVRaIQ>

笔记整理：<https://mp.weixin.qq.com/s/3xHUsqEgg7Kb6MZOdW8tnw>

Kafka 原理以及分区分配策略剖析：<https://mp.weixin.qq.com/s/7njhaaLSsY-U3c8TWZ_jhA>

# 基础

## 介绍

Kafka是一种高吞吐量的分布式发布订阅消息系统，它可以处理消费者规模的网站中的所有动作流数据。

这种动作（网页浏览，搜索和其他用户的行动）是在现代网络上的许多社会功能的一个关键因素。

这些数据通常是由于吞吐量的要求而通过处理日志和日志聚合来解决。

对于像Hadoop的一样的日志数据和离线分析系统，但又要求实时处理的限制，这是一个可行的解决方案。

Kafka的目的是通过Hadoop的并行加载机制来统一线上和离线的消息处理，也是为了通过集群机来提供实时的消费。

Kafka 是一种高吞吐量 的分布式发布订阅消息系统，有如下特性：

1.通过O(1)的磁盘数据结构提供消息的持久化，这种结构对于即使数以TB的消息存储也能够保持长时间的稳定性能。

2.高吞吐量：即使是非常普通的硬件Kafka也可以支持每秒数百万 的消息。

3.支持通过Kafka服务器和消费机集群来分区消息。

4.支持Hadoop并行数据加载

Producer ：消息生产者，向broker发消息的客户端。

Consumer ：消息消费者，向broker取消息的客户端

Topic ：一个队列，主题。

## 基本概念

Message：消息是kafka处理的对象，在kafka中，消息是被发布到broker的topic中。而consumer也是从相应的topic中拿数据。也就是说，message是按topic存储的

Consumer Group ：将topic消息的广播发给consumer的手段。一个topic可以有多个CG。

Broker ：一台kafka服务器就是一个broker。一个集群由多个broker组成。一个broker可以容纳多个topic。

Partition：为了实现扩展性，一个非常大的topic可以分布到多个broker（即服务器）上，一个topic可以分为多个partition，每个partition是一个有序的队列。partition中的每条消息都会被分配一个有序的id（offset）。kafka只保证按一个partition中的顺序将消息发给consumer，不保证一个topic的整体（多个partition间）的顺序。

Offset：kafka的存储文件都是按照offset.kafka来命名，用offset做名字的好处是方便查找。例如你想找位于2049的位置，只要找到2048.kafka的文件即可。当然the first offset就是00000000000.kafka

# 安装

## Linux安装

以下案例以kafka\_2.12-1.0.0进行（该版本支持jdk8）

安装过程：

（1）下载解压。官网下载kafka，http://kafka.apache.org/

解压到安装目录下 tar -xcvf

（2）修改配置文件/usr/local/kafka/config/server.properties，修改如下内容

broker.id=0

host.name=localhost

zookeeper.connect=localhost:2181

log.dirs=/data/kafka\_2.12-2.0.1/logs

（3）修改完配置文件即可将整个文件夹传输到其他节点  scp -r 。。。。

（4）传输完之后修改每个节点的broker.id的编号，递增。

（5）启动zookeeper。

这边可以使用kafka自带的zookeeper，也可以使用自己安装的zookeeper。

启动自己安装的zookeeper  ：

/app/zookeeper-3.4.6/bin/zkServer.sh start

各个节点均启动完成之后，可以查看zk的状态  /app/zookeeper-3.4.6/bin/zkServer.sh status

（6）启动kafka

/app/kafka\_2.9.2-0.8.2.1/bin/kafka-server-start.sh /app/kafka\_2.9.2-0.8.2.1/config/server.properties &

尾部加上&的作用是可以启动完之后直接按回车退出，继续下一步操作。也可不加&

（7）创建topic

/app/kafka\_2.9.2-0.8.2.1/bin/kafka-topics.sh --create --topic my\_demo\_topic --replication-factor 1 --partitions 1 --zookeeper localhost:2181

可通过指令查看所有的topic

/app/kafka\_2.9.2-0.8.2.1/bin/kafka-topics.sh --list --zookeeper localhost:2181

（8）发送消息

/app/kafka\_2.9.2-0.8.2.1/bin/kafka-console-producer.sh --broker-list localhost:9092 --sync --topic my\_demo\_topic

（9）消费消息。重新打开一个终端执行指令

/app/kafka\_2.9.2-0.8.2.1/bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic my\_demo\_topic --from-beginning

在原终端中输入消息，新终端中会显示出输入的消息。

关闭kafka指令。

/app/kafka\_2.9.2-0.8.2.1/bin/kafka-server-stop.sh /app/kafka\_2.9.2-0.8.2.1/config/server.properties &

## Window安装

在kafka安装目录中找到bin\windows目录中的kafka-run-class.bat为%CLASSPATH%加上双引号

修改前：

set COMMAND=%JAVA% %KAFKA\_HEAP\_OPTS% %KAFKA\_JVM\_PERFORMANCE\_OPTS% %KAFKA\_JMX\_OPTS%%KAFKA\_LOG4J\_OPTS% -cp%CLASSPATH% %KAFKA\_OPTS% %\*

修改后：

Set COMMAND=%JAVA%%KAFKA\_HEAP\_OPTS% %KAFKA\_JVM\_PERFORMANCE\_OPTS% %KAFKA\_JMX\_OPTS%%KAFKA\_LOG4J\_OPTS% -cp"%CLASSPATH%"%KAFKA\_OPTS% %\*

* 启动自带的zk

zookeeper-server-start.bat ../../config/zookeeper.properties

* 启动kafka

kafka-server-start.bat ../../config/server.properties

## 启动远程访问

<https://blog.csdn.net/u010889616/article/details/80640330>

<https://blog.csdn.net/fyq2016/article/details/103335982>

## Kafka安装及开启SASL\_PLAINTEXT认证（用户名和密码认证）

<https://www.cnblogs.com/iamsach/p/9234624.html>

# 分区

kafka中生产者是如何把消息投递到哪个分区的？消费者又是怎么选择分区的？

<https://mp.weixin.qq.com/s/fHmqZsetdY-cDb-ylrUgIg>

# 消息重试

<https://www.infoq.cn/article/51XSHW2opSmakhHmtth8>

微信备份：<https://mp.weixin.qq.com/s/R6-_N9zhbqbXVuZaJg0OlQ>

# kafka日志保存时间设置

https://www.cnblogs.com/angellst/p/9368493.html

# 副本复制