

快学Spark 2.0

第30讲、Kafka架构体系和概念

讲师：Cloudy（北风网版权所有）

答疑QQ: 2510337544

Kafka

传统的消息队列产品：

Jboss里的jms ， 单分区（无分区）， 只有一个点进行数据读写。

RecodetMQ， 阿里开源的， 多实例的， 和Kafak区别较大， 支持多分区。前身是MetaQ。

课程采用：

Kafka是LinkedIn开源出来的一个高吞吐的分布式消息系统。

使用scala开发， 支持多语言客户端（c++、java、python等）

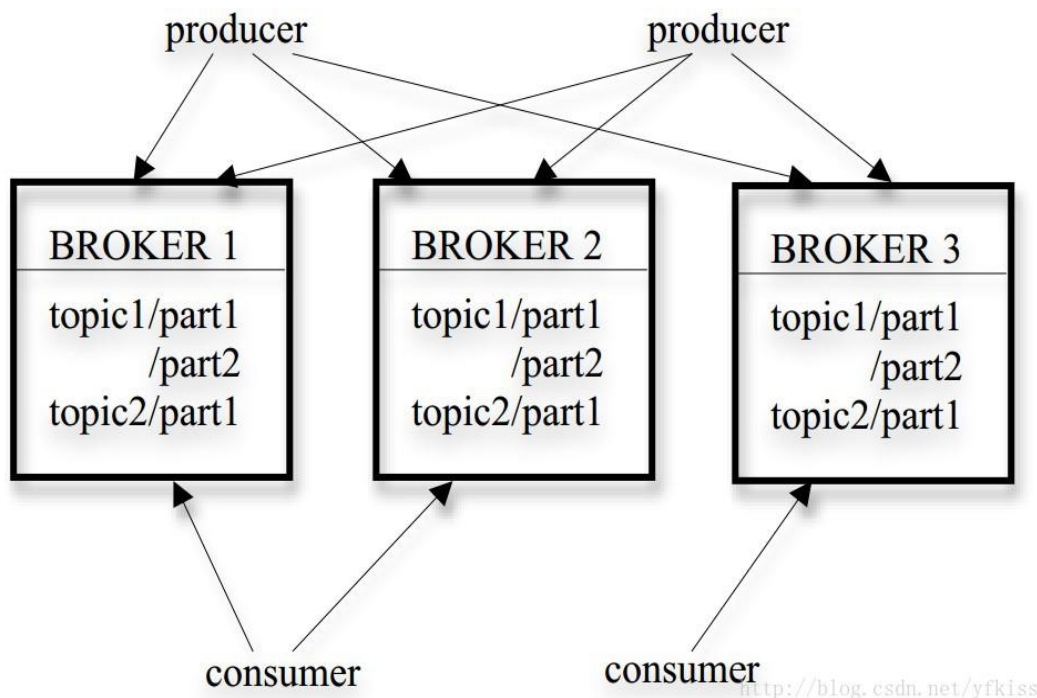
其具有以下特点：

- 1) 支持高Throughput（高吞吐量）的应用， 多分区。
- 2) 无需停机即可扩展机器
- 3) 持久化： 通过将数据持久化到硬盘以及replication（副本）防止数据丢失。
- 4) 支持online和offline的场景， 实时消费和离线消费两种都支持。

Storm支持离线消费Kafka吗？ 不支持。Storm只进行实时计算， 所以只实时消费。

offline消费： 定期消费一次， 比如按小时或天。离线计算。Spark、Mapreduce。

Kafka架构



Broker: 服务端进程，消息中间件处理结点，

一个kafka节点就是一个broker

Producer: 消息发布者

Consumer: 消息订阅者。

Broker就是集群的节点，增加Broker扩容集群。

消息

kafka的消息分几个层次：

1) **Topic**: 一类消息，例如page view日志，click日志等都可以以topic的形式存在，kafka集群能够同时负责多个topic的分发。

2) **Partition**: Topic物理上的分组，一个topic可以分为多个partition(默认是2个)，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 id (offset, 消费位置)。

3) **Message**: 消息，最小订阅单元

消息处理流程

数据流程：

1. **Producer**根据指定的**partition**方法（**round-robin**、**hash**等），将消息发布到指定**topic**的**partition**里面。

写数据时，可以指定写**topic**的哪个分区。

如某**topic**建**24**个分区，每个分区存**1**个小时的数据。消费时指定对应的分区消费数据。

2. **kafka**集群接收到**Producer**发过来的消息后，将其持久化到硬盘，并保留消息指定时长（可配置，默认**7天**），而不关注消息是否被消费。

3. **Consumer**从**kafka**集群消费（**pull**）数据，并控制获取消息的**offset**偏移量。

push：服务端推送给客户端（**consumer**），增加了服务端的压力，服务端需要维护每个**Consumer**的**offset**。**kafka**不支持这个模式，说白了就是淘汰了这种低效的模式。

pull：**Kafka**唯一消费模式。拉，**Consumer**主动去拉取数据，自己维护自己的**offset**。对我们开发来讲，稍微麻烦些，因为需要我们自己维护**offset**。

Kafka如何实现高吞吐量？

High Throughput是kafka需要实现的核心目标之一，为此kafka做了以下一些设计：

- 1) 数据持久化磁盘：消息不在内存中cache，直接写入到磁盘，充分利用磁盘的顺序读写性能，所以broker没有内存压力。
- 2) zero-copy：减少IO操作步骤。
- 3) 数据批量发送
- 4) 数据压缩
- 5) Topic划分为多个partition，提高parallelism（并行度）

负载均衡 load balance&HA

Kafka如何实现负载均衡的？

- 1) producer根据用户指定的算法，将消息发送到指定的partition。
这里说的指定的算法，实际就是分区方式，比如按小时。
- 2) 存在多个partiiton，每个partition有自己的replica（副本），每个replica分布在不同的Broker节点上。
- 3) 多个partition时需要选取出leader partition（通过zk的选举机制），leader partition负责读写，并由zookeeper负责fail over。
- 4) 通过zookeeper管理broker与consumer的动态加入与离开。

扩容

当需要增加broker节点时，新增的broker会向zookeeper注册，而producer及consumer会根据注册在zookeeper上的watcher感知这些变化，并及时作出调整。

集群搭建步骤

1. Kafka下载:

如何选择兼容的Kafka版本? 基本都兼容, 不宜用高版本的。

考虑到Spark1.6用的是kafka_2.11-0.8.2.1, 我们继续采用这个版本即可。

```
wget http://apache.fayea.com/kafka/0.8.2.1/kafka_2.11-0.8.2.1.tgz
```

```
解压 tar zxvf kafka_2.11-0.8.2.1.tgz
```

kafka使用scala编写, 需要下载scala相关的库.

2.11是Scala版本, 0.8.2.1是Kafka版本

2. 下载安装sbt:

下载地址: <http://dl.bintray.com/sbt/rpm/>

选择sbt-0.13.1.rpm 版本及以上

安装yum install -y sbt-0.13.1.rpm 或 rpm -ivh sbt-0.13.1.rpm

集群搭建步骤

3. 更新scala环境:

```
cd kafka_2.11-0.8.2.1
```

```
sbt update
```

```
sbt package
```

更新比较慢，需要等几分钟，可以先配置第4步。

4. 配置config/server.properties

broker.id为依次增长的：0、1、2、3、4，集群中唯一id

log.dirs设置到大硬盘路径下

num.network.threads

num.partitions ，默认分区数

num.io.threads 建议值为机器的核数；

zookeeper.connect 设置为zookeeper Servers 列表，各节点以逗号分开；

Kafka 启动、创建topic

在kafka的部署目录下，**在各个节点上**通过如下命令来启动：

```
$ nohup bin/kafka-server-start.sh config/server.properties &
```

```
[root@master ~]# jps
```

```
3098 ZooKeeperMain
```

```
3037 Kafka
```

创建topic:

```
$ bin/kafka-topics.sh --zookeeper 192.168.1.107:2181,192.168.1.108:2181  
--topic topicName --replication-factor 1 --partitions 1 --create
```

其中， --topic 定义topic名

--replication-factor 定义副本数

--partitions 定义分区数

查看全部Topic:

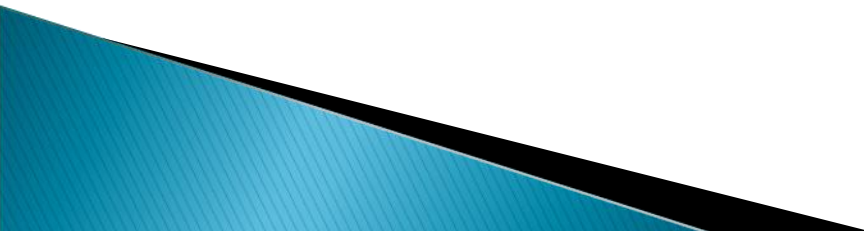
```
bin/kafka-topics.sh --zookeeper 192.168.1.107:2181,192.168.1.107:2181 --list
```

删除Topic:

```
./kafka-topics.sh --topic topicName --delete --zookeeper  
192.168.1.107:2181,192.168.1.107:2181
```

查看Topic明细:

```
./kafka-topics.sh --topic test --describe --zookeeper  
192.168.1.107:2181,192.168.1.107:2181
```



生产和消费

Producer:

```
bin/kafka-console-producer.sh --broker-list  
192.168.1.107:9092,192.168.1.108:9092 --topic topicName
```

Consumer:

```
bin/kafka-console-consumer.sh --zookeeper  
192.168.1.107:2181,192.168.1.107:2181 --topic topicName --from-beginning
```

异常处理

报异常:

Failed to load class "org.slf4j.impl.StaticLoggerBinder"

解决方法:

下载slf4j-1.7.6.zip

```
wget http://www.slf4j.org/dist/slf4j-1.7.6.zip
```

解压

```
unzip slf4j-1.7.6.zip
```

把slf4j-nop-1.7.6.jar 包复制到kafka libs目录下面

配置文件config/server.properties 中配置项:

broker.id	整数, 建议根据ip区分	
log.dirs	kafka存放消息文件的路径,	默认/tmp/kafka-logs
port	broker用于接收producer消息的端口	
zookeeper.connect	zookeeper连接	格式为 ip1:port,ip2:port,ip3:port
message.max.bytes	单条消息的最大长度	
num.network.threads	broker用于处理网络请求的线程数	如不配置默认为3, server.properties默认是2
num.io.threads	broker用于执行网络请求的IO线程数	如不配置默认为8, server.properties默认是2可适当增大,
queued.max.requests	排队等候IO线程执行的requests	默认为500
host.name	broker的hostname	默认null,建议写主机的ip,不然消费端不配置hosts会有麻烦
num.partitions	topic的默认分区数	默认2
log.retention.hours	消息被删除前保存多少小时	默认1周168小时
auto.create.topics.enable	是否可以程序自动创建Topic	默认true,建议false
default.replication.factor	消息备份数目	默认1不做复制, 建议修改
num.replica.fetchers	用于复制leader消息到follower的IO线程数	默认1

本课程版权归北风网所有

欢迎访问我们的官方网站
www.ibeifeng.com