

Kafka

为什么Storm需要一个消息队列

答案很简单，解决和其他系统耦合的问题
如：log -> kafka -> spark->kafka->db库

消息队列的特点

生产者消费者模式

- 先进先出（FIFO）顺序保证
- 可靠性保证
 - 自己不丢数据（数据设置时间清除，kafka默认是7天）
 - 消费者不丢数据：“至少一次，严格一次”
- 至少一次就是可能会有两次，会重
- 严格一次机制就会负责一点

Kafka架构

- producer：消息生产者
- consumer：消息消费者
- broker：kafka集群的server，(其实就是一台机器)

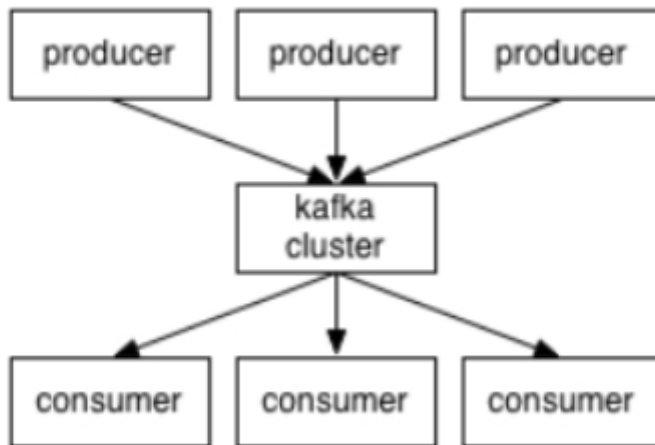
负责处理消息读、写请求，存储消息

- topic：消息队列/分类
- Queue里面有生产者消费者模型
- broker就是代理，在kafka cluster这一层这里，其实里面是有很多个broker（物理机器）
- topic就相当于queue

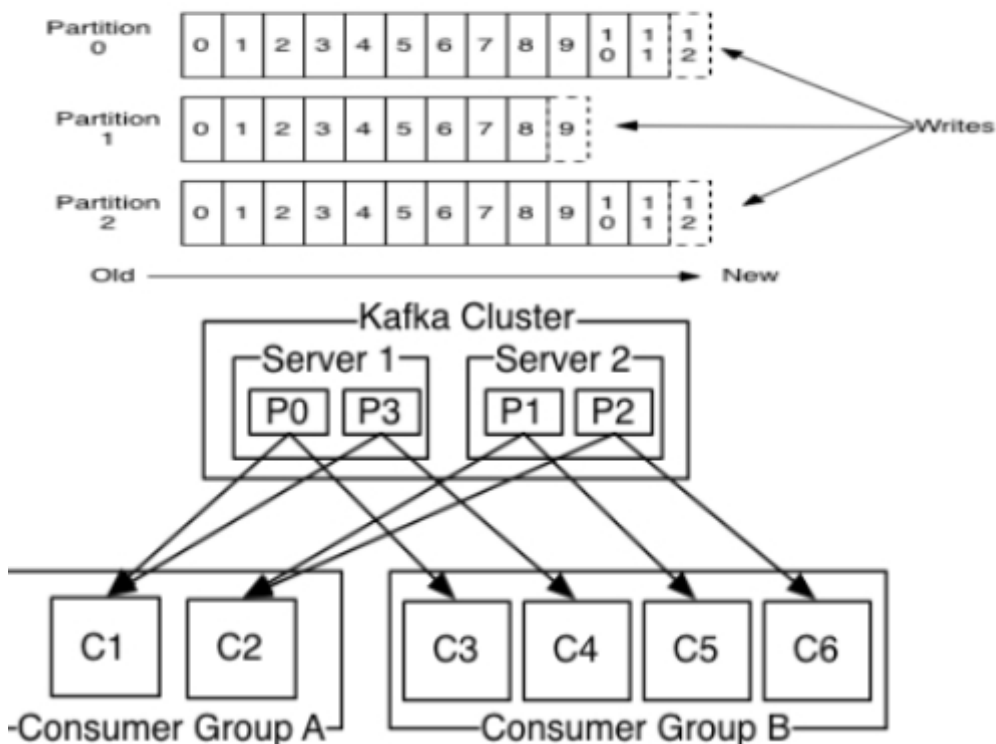
- 图里没有画其实还有zookeeper，这个架构里面有些元信息是存在zookeeper上面的，整个集群的管理也和zookeeper有很大的关系
- 一个topic（抽象概念）分成多个partition（物理概念）
- 每个partition内部消息强有序(一个topic怎么保证强有序，创建一个partition就可以保证)，其中的每个消息都有一个序号叫offset（就是下图的号码）
- 一个partition只对应一个broker，一个broker可以管多个partition
- 消息不经过内存缓冲，直接写入文件
- 根据时间策略删除，而不是消费完就删除
- producer自己决定往哪个partition写消息，可以是轮询的负载均衡，或者是基于hash的partition策略（以key-value写入，如果key是null则为轮训）

怎么提高topic的并行度？增加partition数量

Kafka不允许一个partition被同组里面多个consumer里面读



Anatomy of a Topic



搭建kafka

在kafka的文件夹中，我们写一个这样的脚本，就能在后台运行kafka
运行kafka，加载properties文件，将日志写到log中

2>&1表示将标准错误也写入到log中

```
nohup ./bin/kafka-server-start.sh ./config/server.properties > kafka.log 2>&1 &
```

```
[root@node3 kafka_2.11-1.0.1]# vi startkafka.sh
```

因为这个kafka不是直接安装的，所以要加执行权限在bin下

配置配置文件

```
vim server.properties
```

配置好brokerid，集群中，每台id都不一样

这里说是日志，其实就是kafka的数据

数据默认是7天

配置zookeeper

启动

```
[root@node3 kafka_2.11-1.0.1]# bash startkafka.sh
```

查看topic详细信息

```
./bin/kafka-topics.sh --describe --zookeeper  
node1:2181,node3:2181,node4:2181
```

从信息中可以看出，写和读，都做到了负载均衡

创建topic

```
./bin/kafka-topics.sh --create --  
zookeeper node1:2181,node3:2181,node4:2181 --topic  
20180310 --partitions 3 --replication-factor 1
```

只想看某个topic

消费数据

生产数据

注意这个broker是kafka的地址，不是zookeeper的，之后往里面输入数据就可以了

```
[root@spark001 kafka_2.10-0.8.2.1]# chmod +x ./bin/*  
  
# The id of the broker. This must be set to the id of the  
broker.id=1  
-----  
log.dirs=/home/kafka/kafka-logs  
-----  
# The minimum age of a log file to be eligible for deletion  
log.retention.hours=168  
-----  
# root directory for all kafka znodes.  
zookeeper.connect=node1:2181,node3:2181,node4:2181
```

```
[root@node3 kafka_2.11-1.0.1]# ./bin/kafka-topics.sh --describe --zookeeper node1:2181,node3:2181,node4:2181
Topic:20180310 PartitionCount:3 ReplicationFactor:1 Configs:
  Topic: 20180310 Partition: 0 Leader: 1 Replicas: 1 Isr: 1
  Topic: 20180310 Partition: 1 Leader: 1 Replicas: 1 Isr: 1
  Topic: 20180310 Partition: 2 Leader: 1 Replicas: 1 Isr: 1
[root@node3 kafka_2.11-1.0.1]# ./bin/kafka-topics.sh --create --zookeeper node1:2181,node3:2181,node4:2181 --
topic 20180310 --partitions 3 --replication-factor 1
[root@spark002 kafka_2.10-0.9.2.1]# ./bin/kafka-topics.sh --describe --zookeeper spark001:2181,spark002:2181,spark003:2181 --topic 20160410
Topic:20160410 PartitionCount:3 ReplicationFactor:2 Configs:
  Topic: 20160410 Partition: 0 Leader: 1 Replicas: 1,2 Isr: 1,2
[root@spark002 kafka_2.10-0.9.2.1]# ./bin/kafka-console-consumer.sh --zookeeper spark001:2181,spark002:2181,spark003:2181 --topic 20160410
[...]
```

启动kafka（内置zookeeper）

`./zookeeper-server-start.sh`

`/usr/local/kafka/kafka_2.11-`

`2.0.0/config/zookeeper.properties & ./kafka-`

`server-start.sh /usr/local/kafka/kafka_2.11-`

`2.0.0/config/server.properties &`

启动问题：

Connection to node -1 could not be established.

Broker may not be available.

配置文件的配置ip地址为localhost，外网是无法访问到的

正确的改为：

`advertised.listeners=PLAINTEXT://120.79.51.194:9092`