# 知识回顾

略

# 实时分析

## Kafka简介

他是一个MQ，作用：削峰平谷

ActiveQM（6k）、RabbitMQ(1.2w) 适用与业务系统（对事务要求极高）

ZeroMQ(25-50w)、Kafka(25-50w) 适用于大数据（对速度和吞吐量要求高，海量数据中，丢了一两条，对结果影响微乎其微）

Kafka中模式，只有一种，topic。主题模式。

Kafka已经成为大数据业界主流。

## Storm简介

他是一个分布式的实时数据分析系统，地层基于ZeroMQ做数据传输，通过clojure（语言）开发核心模块。

Storm速度非常快，能达到亚秒级（200sm）。

SparkStreaming也是做实时分析的，但是速度不及Storm，大概在1-5秒，但是吞吐量远远超过Storm。Storm相当于一条一条处理，它相当于批量处理。

问题：

1. flume整合kafka。把数据输出到kafka中
2. kafka整合Storm。把数据输出到Storm中
3. Storm如何处理数据。

## Storm详解

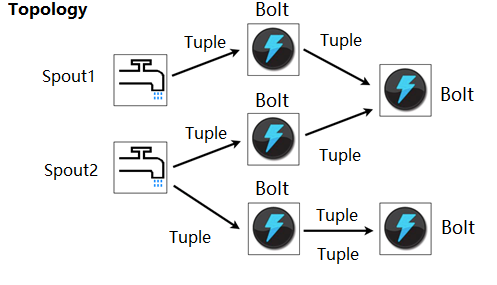
### 组件

Nimbus：老大，storm的发号施令者。（相当于hadoop中的namenode）集群中只能有一个。

Supervisor：小弟，具体业务执行者。（相当于hadoop中的datanode）

Zookeeper：集群资源管理者，监听整个集群的健康状态。

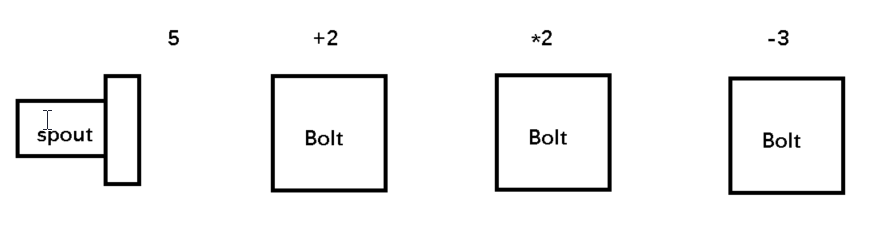
### 核心四大要素



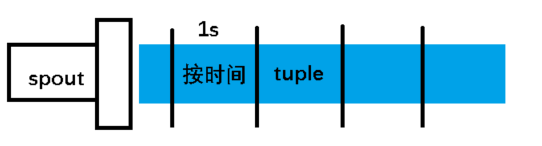
Topology：拓扑，就代表Storm整个数据处理过程。

Spout：喷嘴，就是数据源。

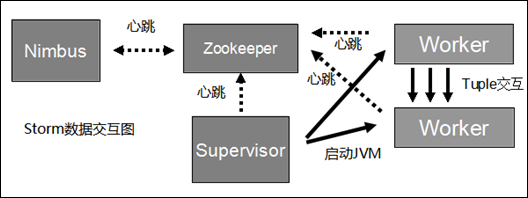
Bolt：闪电（算子），内部需要我们自己定义一些处理逻辑，而且一个topology中 可以有多个bolt。并且一个bolt可以接受多个数据来源，有多个数据出处。类似flume中的扇入扇出。



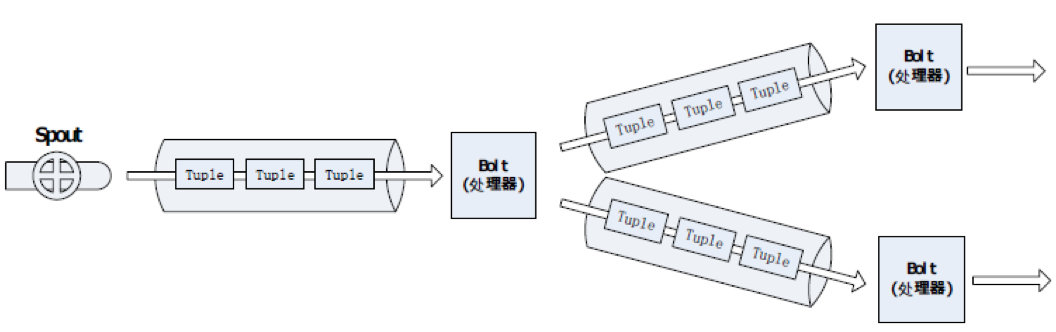
Tuple：数据的载体。内部类似map（key,value），value类型一旦定义，不能存放其他类型的数据。Tuple不太一样，key一般是String类型，value类型不定，数据结构不定。



补充：



Jvm？java虚拟机，随着技术的发展，java有了很多衍生语言，scala、clojure同样编译.class，通过JVM来执行。



Tuple的切割时间不是任意确定的，需要经过多次的尝试，没有最好的只有最合适的。

## Storm的安装

### 准备环境

启动Zookeeper

### 创建文件夹

Mkdir storm

### 上传安装包

### 解压安装包

### 修改配置文件

cd /usr/local/src/storm/apache-storm-0.9.3/conf

vi storm.yaml

storm.zookeeper.servers:

- "hadoop01"

- "hadoop02"

- "hadoop03"

#

nimbus.host: "hadoop01"

storm.local.dir: "/usr/local/src/storm/apache-storm-0.9.3/tmp"

参数说明：

strom.zookeeper.servers 配置zk集群

nimbus.host 配置numbus所在服务器

storm.local.dir 配置临时文件所在路径

注意：yaml格式文件非常差，稍微格式错误就无法正常使用。在:后面都有空格。不要自己敲入，最好复制下发的storm.yaml进行修改覆盖linux中的文件。

### 配置storm环境变量

vi /etc/profile

#set storm env

export STORM\_HOME=/usr/local/src/storm/apache-storm-0.9.3

export PATH=$STORM\_HOME/bin:$PATH

### 复制到其他节点，包括环境变量

## Storm启动

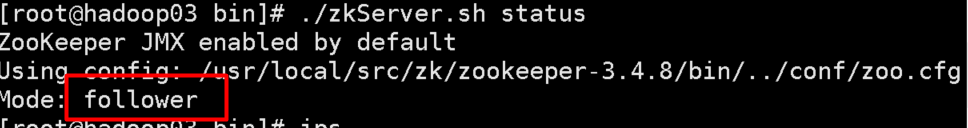
### 启动zookeeper

在hadoop01 02 03 三台节点上分别执行

Bin> ./zkServer.sh start

执行下边命令检测zk集群是否启动成功

Bin> ./zkServer.sh status



此时，zk启动成功

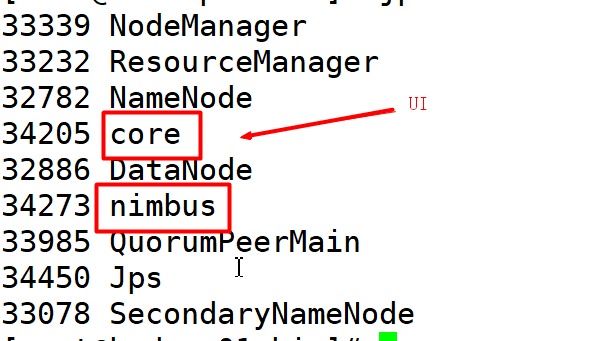
### 启动storm

1.在hadoop01上执行，启动ui 及nimbus

cd /usr/local/src/storm/apache-storm-0.9.3/bin

./storm ui >/dev/null 2>&1 & #启动UI Web界面，访问端口8080

./storm nimbus >/dev/null 2>&1 & #启动nimbus服务

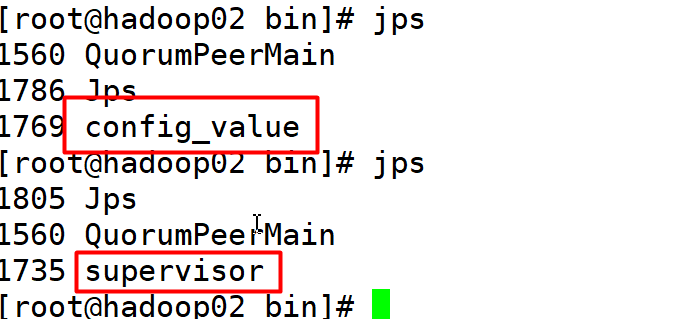


注意：ui占用8080端口，与tomcat默认端口冲突，如有tomcat，修改端口为其他，为Storm让路。

在hadoop02和hadoop03上执行，启动supervisor

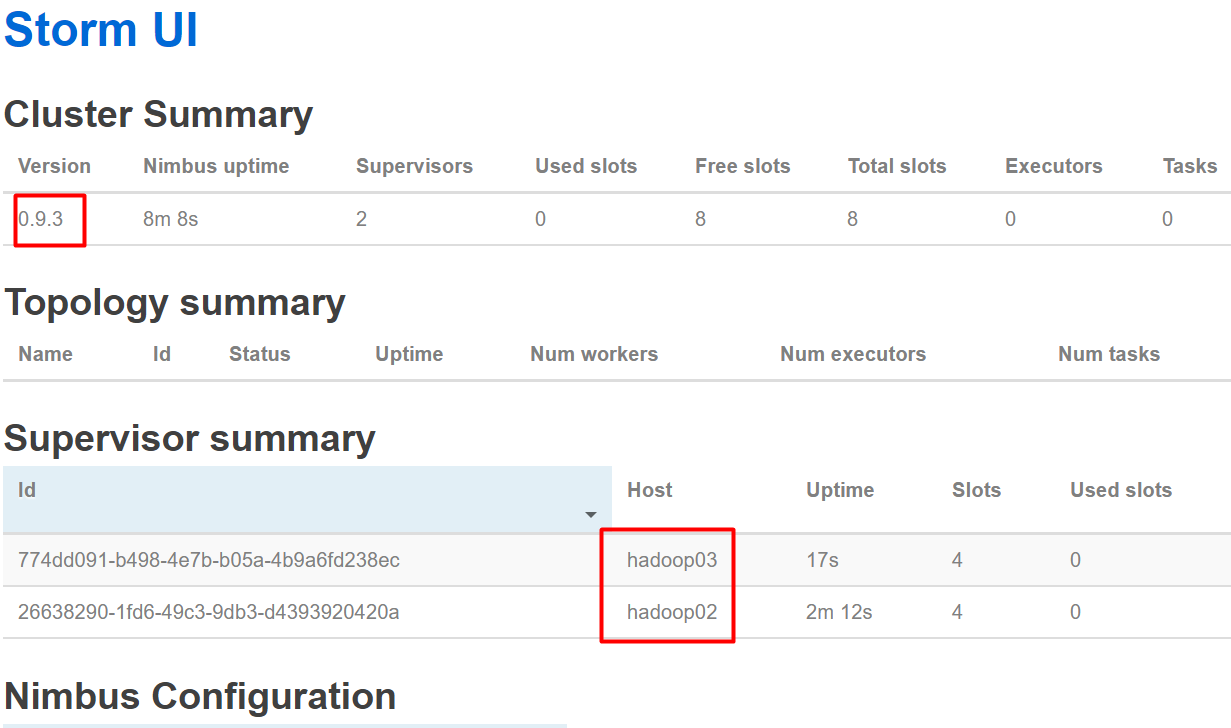
cd /usr/local/src/storm/apache-storm-0.9.3/bin

./storm supervisor >/dev/null 2>&1 & #启动supervisor服务



在Storm进程启动过程中我们发现有config\_value的出现，这是启动的中间过程，成功后会变为对应的进程。

### 测试



## Kafka安装

### 创建目录

### 上传解压

### 修改配置文件

cd /usr/local/src/kafka/kafka\_2.10-0.10.0.1/config

vi server.properties

broker.id=0 #当前server编号

log.dirs=/usr/local/src/kafka/kafka\_2.10-0.10.0.1/tmp/kafka-logs #日志存储目录

zookeeper.connect=hadoop01:2181,hadoop02:2181,hadoop03:2181

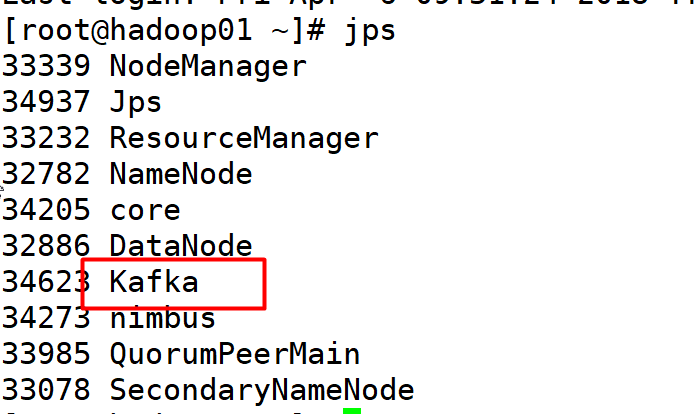
advertised.host.name=hadoop01

advertised.port=9092

### 启动

cd /usr/local/src/kafka/kafka\_2.10-0.10.0.1

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



### 创建主题

创建主题flux

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

参数说明：

--create: 要创建主题

--zookeeper : zk的ip和端口

--replication-factor：副本个数

--partitions：主题分区存储，分区的个数

--topic：主题名(唯一，不能重复)

可以使用 --list 命令查看所有topic

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

### 生产者

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

--broker-list kafka的ip和端口

--topic 要生产的消息属于哪个主题

### 消费者

bin/kafka-console-consumer.sh --zookeeper localhost:2181 --topic flux --from-beginning

参数说明：

--from-beginning 是表示从头开始消费消息

If the consumer does not already have an established offset to consume from, start with the earliest message present in the log rather than the latest message.

消息已经永久保存在kafka集群了，从头消费或是从当前时间消费都可以。

## Flume整合kafka

### 修改flume配置文件

[root@hadoop01 conf]# vi flume-jt.properties

#命名Agent a1的组件

a1.sources = r1

a1.sinks = k1 k2

a1.channels = c1 c2

#描述/配置Source

a1.sources.r1.type = avro

a1.sources.r1.bind = hadoop01

a1.sources.r1.port = 22222

a1.sources.r1.interceptors = t1

a1.sources.r1.interceptors.t1.type = timestamp

#描述Sink

a1.sinks.k1.type = hdfs

a1.sinks.k1.hdfs.path = hdfs://hadoop01:9000/flux/reportTime=%Y-%m-%d

a1.sinks.k1.hdfs.fileType=DataStream

a1.sinks.k2.type = org.apache.flume.sink.kafka.KafkaSink

a1.sinks.k2.brokerList = localhost:9092

a1.sinks.k2.topic = flux

#描述内存Channel

a1.channels.c1.type = memory

a1.channels.c1.capacity = 1000

a1.channels.c1.transactionCapacity = 100

a1.channels.c2.type = memory

a1.channels.c2.capacity = 1000

a1.channels.c2.transactionCapacity = 100

#为Channle绑定Source和Sink

a1.sources.r1.channels = c1 c2

a1.sinks.k1.channel = c1

a1.sinks.k2.channel = c2

注意：添加或者修改黄色的部分，缺少将无法正常运行。

### 启动flume

[root@localhost conf]# ../bin/flume-ng agent -c ./ -f ./flume-jt.properties -n a1 -Dflume.root.logger=INFO,console &

#注意在conf配置文件目录下运行

启动jtlogserver测试

## Kafka整合Storm

创建java工程

导入相关jar包

创建PrintBolt类

创建FluxTopology类

测试

添加ClearBolt类

测试

课后作业：

准备六台虚拟机 512M内存。hadoop1~6

配置静态IP 121~126

链接并保存Xshell

拍摄快照