实时检测共享账户详细部署说明书

## Hadoop的安装部署

1. 在安装hadoop之前需要下载vmware虚拟机，这里就不多做解释，下面开始从linux的配置开始安装部署hadoop
2. Linux系统配置

以下操作步骤需要在 HadoopMaster 和 HadoopSlave 节点上分别完整操作， 都使用 root 用户，从当前用户切换 root 用户的命令如下:

su root

输入密码：

zkpk

1. 时钟同步：就是修改一台Server的时间，让其与世界公用的时间同步, 该项同时需要在 HadoopSlave 节点配置

该命令是 vi 编辑命令，按 i 进入插入模式，按 Esc，然后键入:wq 保存退出

键入下面的一行代码，输入 i，进入插入模式（星号之间和前后都有空格）

0 1 \* \* \* /usr/sbin/ntpdate cn.pool.ntp.org

手动同步时间：

直接在 Terminal 运行下面的命令： /usr/sbin/ntpdate cn.pool.ntp.org

1. 配置主机名：
   * 1. HadoopMaster 节点

使用 gedit 编辑主机名，如果不可以使用 gedit，请直接使用 vi 编辑器（后面用到 gedit 的地方也同此处处理一致） 。

gedit /etc/sysconfig/network

配置信息如下，如果已经存在则不修改，将 HadoopMaster 节点的主机名改为 master，即下面代码的第2 行所示：

NETWORKING=yes #启动网络

HOSTNAME=master #主机名

确实修改生效命令： hostname master

检测主机名是否修改成功命令如下，在操作之前需要关闭当前终端，重新打开一个终端： hostname

* + 1. HadoopSlave 节点 ：

使用 gedit 编辑主机名：

gedit /etc/sysconfig/network

配置信息如下，如果已经存在则不修改，将 Hadoopslave 节点的主机名改为 slave，即下面代码的第 2行所示。

NETWORKING=yes #启动网络

HOSTNAME=slave #主机名

确实修改生效命令： hostname slave

检测主机名是否修改成功命令如下，在操作之前需要关闭当前终端，重新打开一个终端： hostname

* + 1. HadoopSlave1 节点 ：

同Slave一样只需把slave换为slave1即可

1. 使用 setup 命令配置网络环境

在终端中执行下面命令： setup

使用光标键移动选择“Network configuration”，回车进入该项

使用光标键移动选择 eth0，回车进入该项

修改Static IP 与 Netmask为

192.168.150.128

255.255.255.0

相应修改slave与slave1节点的参数(IP要填当与master节点不同的ip)

重启网络服务 /sbin/service network restart

检查是否修改成功： ifconfig 查看IP地址是否与你配置的相同

1. 关闭防火墙

master/slave/slave1 都必须关闭掉防火墙

在集群中节点与节点之间会频繁的通信，如果有防火墙就会导致通信失败，所以要关闭掉

在终端中执行下面命令： setup

光标移动选择“Firewall configuration”选项，回车进入选项

光标移动选择“Firewall configuration”选项，回车进入选项

如果该项前面有“\*”标，则按一下空格键关闭防火墙，如下图所示，然后光标移动选择“OK”保存 修改内容

1. 配置 hosts 列表

该项需要在 HadoopSlave 节点配置。

需要在 root 用户下（使用 su 命令） ，编辑主机名列表的命令： gedit /etc/hosts

将下面两行添加到/etc/hosts 文件中：

192.168.1.100 master

192.168.1.101 slave

注意：这里 master 节点对应 IP 地址是 192.168.1.100，slave 对应的 IP 是 1 92.168.1.101，而自己在做配 置时，需要将这两个 IP 地址改为你的 master 和 slave 对应的 IP 地址。 查看 master 的 IP 地址使用下面的命令： ifconfig

检验：

ping master (or ping ip地址)

ping slave

1. 至此，linux的基础配置到此结束，如果在三个节点上都能互相ping通说明节点搭载完毕

如果想切换节点不输密码的话可以配置免密钥登录，这样就能不输入密码直接使用

ssh slave or ssh slave1 or ssh master来切换节点了

## JDK的安装部署

1. 安装 JDK

将 JDK 文件解压，放到/usr/java 目录下 cd /home/zkpk/resources/software/jdk

mkdir /usr/java

mv jdk-7u71-linux-x64.gz /usr/java/

cd /usr/java

tar -xvf jdk-7u71-linux-x64.gz

使用 vim 配置环境变量 vim /home/zkpk/.bash\_profile

复制粘贴以下内容添加到到上面 vim 打开的文件中：

export JAVA\_HOME=/usr/java/jdk1.7.0\_71

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

使改动生效命令： source /home/zkpk/.bash\_profile

测试配置： java -version 或 java

如果出现下图的信息，表示 JDK 安装成功：

## Eclipse的安装与使用

1. 我们在搞上层开发的时候，都是在Windows下使用Eclipse，那么如果是Linux应用开发，就必须要在Linux中安装Eclipse，用于C/C++开发，当前是要在Linux带界面的系统中运行了，我这里就重新安装了一个带界面的CentOS6.3。



1. 安装JDK，因为刚刚我们已经配置过JDK了，所以这里可以参考第二点，如果想下载更新版本的JDK可以去官网下载tar.gz版本

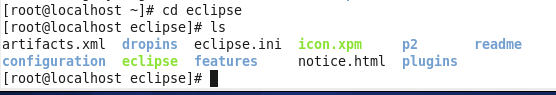
官网截图：



1. 这个下载过程我就不演示了，然后打开Eclipse的官网，下载Eclipse

这里我提供给大家：eclipse-jee-indigo-SR2-linux-gtk-x86\_64.tar

在相关软件文件夹下即可找到，版本比较低，可以去官网下载

1. 下载好，解压即可（解压过程不演示了）
2. 然后配置环境变量/etc/profile(第二点中配置过了)
3. 解压完之后进入cd eclipse
4. 
5. 然后./eclipse启动eclipse



1. 然后new一个Java 项目，编写模拟数据代码，kafka与storm连通的spout代码与bolt代码等……

## Zookeeper的安装部署

1. 用到的安装包:zookeeper-3.4.5.tar.gz

拷贝到家目录下解压:tar –zxvf zookeeper-3.4.5.tar.gz ~./

1. 设置环境变量:

vim ./bash\_profile

export ZOOKEEPER\_HOME=/home/zkpk/zookeeper-3.4.5

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

1. 在zookeeper-3.4.5/conf目录下修改文件名

cp zoo\_sample.cfg zoo.cfg 或者 mv zoo\_sample.cfg zoo.cfg

1. 编辑zoo.cfg : vim zoo.cfg

tickTime=2000 是zooKeeper工作的单位时间

initLimit = 10\*2000 选举与同步的最大时间

syncLimit = 5\*2000 请求与响应确认的最大时间

修改dataDir=/home/zkpk/zookeeper-3.4.5/data

增加dataLogDir=/home/zkpk/zookeeper-3.4.5/dataLog

新增以下内容:

server.1=master:2888:3888 2888:用来选举的 3888：用来同步的

server.2=slave:2888:3888

server.3=slave1:2888:3888

clientPort = 2181

1. 修改myid:

cd zookeeper-3.4.5/data

vim myid

改为所在节点对应的server编号

(比如你在master节点修改为1，在slave节点修改为 2，slave1节点修改为3......)

1. 将zookeeper-3.4.5 拷贝到slave slave1节点 然后修改其上的myid

scp -r zookeeper-3.4.5 zkpk@slave:~/

scp -r zookeeper-3.4.5 zkpk@slave1:~/

1. 启动

分别在master slave slave1

zkSever.sh start

检验:在三个节点上分别执行命令:

zkServer.sh status查看节点是leader 还是 follow

JPS后有如下进程:

QuorumPeerMain

进入zookeeper客户端:

zkCli.sh -server master:2181

执行命令:ls /

有如下显示证明zookeeper安装成功

[zookeeper]

## Flume的安装部署

1. 前期准备:

版本: apache-flume-1.6.0-bin.tar

apache-flume-1.6.0-src.tar

解压文件:

tar -zxvf apache-flume-1.6.0-bin.tar.gz

tar -zxvf apache-flume-1.6.0-src.tar.gz

将apache-flume-1.6.0-src文件夹中的内容全部复制到apache-flume-1.6.0-bin文件中

cp -ri apache-flume-1.6.0-src/\* apache-flume-1.6.0-bin

1. 环境变量配置:

vim /.bash\_profile

JDK环境变量:

export JAVA\_HOME=/usr/java/jdk1.7.0\_71

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

Flume环境变量:

export FLUME\_HOME=/home/zkpk/apache-flume-1.6.0-bin

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

环境变量生效:

source .bash\_profile

1. 配置文件配置:

flume-env.sh

进入到flume.env.sh的存放路径

cd apache-flume-1.6.0-bin/conf

vim flume.env.sh

配置文件中配置:JAVA\_HOME=JAVA\_HOME=/usr/java/jdk1.7.0\_71

建立配置文件example实现Flume监听文件夹 ，数据写入文件夹内时将数据汇聚 到HDFS

首先我们建立一个/home/xdl/apache-flume-1.6.0-bin/conf/example.conf文件

vim example.conf

编写配置文件

对于下面红字部分，记得创建文件夹，并且注意他们的权限一致，这个比较简单的， 就不再书写了。

具体内容如下：

# Name the components on this agent

agent1.sources = source1

agent1.channels = channel1

agent1.sinks = sink1

# Describe/configure the source

agent1.sources.source1.type = spooldir

agent1.sources.source1.spoolDir=/home/zkpk/aboutyunlog

#配置往channel1传输数据

agent1.sources.source1.channels = channel1

agent1.sources.source1.fileHeader=false

# Describe the sink

agent1.sinks.sink1.type = hdfs

agent1.sinks.sink1.hdfs.path=hdfs://master:9000/flume/aboutyunlog

agent1.sinks.sink1.hdfs.fileType=DataStream

agent1.sinks.sink1.writeFormat=TEXT

# hdfs sink间隔多长将临时文件滚动成最终目标文件，单位：秒；

agent1.sinks.sink1.hdfs.rollInterval=4

#配置从channel1接收数据

agent1.sinks.sink1.channel = channel1

# Use a channel which buffers events inmemory

agent1.channels.channel1.type = file

agent1.channels.channel1.checkpointDir=/home/zkpk/aboutyun\_check

agent1.channels.channel1.dataDirs=/home/zkpk/aboutyun\_data

注意：Spooling Directory是监控指定文件夹中新文件的变化，一旦新文件出现，就解析该文件内容，然后写入到channel。写入完成后，标记该文件已完成或者删除该文件。

1. 编写shell脚本

vim avro.bash

#!/bin/bash

in/flume-ng agent -n a1 -c conf -f conf/avro.conf –Dflume.root.logger=INFO,console

//结果信息-->INFO 在控制台打印输出 如果想重定向可以在脚本最后加 >./avro.log 2>&1 &

实时监控日志文件(前提是重定向(>./avro.log 2>&1 &)后)

tail -f ./avro.log

往Flume监控的目录/home/zkpk/aboutyunlog下创建文件

vim a.txt

随便输一些字母保存退出

1. 查看结果:

在控制台最后一行可看到a.txt文件中的内容

在此项目中，因为我们是利用两个子节点做发送，主节点做汇聚，所以相应的配置文件和脚本文件我放在了相应的文件夹中。.conf文件拷贝到/conf目录下， .bash文件拷贝到flume安装包下即可，依次按照master，slave，slave1的顺序启动脚本文件 tail -f XXX.log 来实时监控日志变化

## Kafka的安装部署

Kafka是一个分布式的发布与订阅系统，善于做缓存和数据的分发

1. 安装条件

hadoop已经成功运行

zookeeper正常启动

1. 配置

vim config/server.properties

broker.id=0

host.name=master //主机名

//port=9092 //端口默认为9092 不用配置

zookeeper.connect=master:2181,slave:2181,slave1:2181

1. 将kafka安装程序拷贝到slave slave1 节点

分别在hadoop2和hadoop3节点完成如下操作：

cd ~/kafka\_2.11-1.1.0

vim config/server.properties

修改如下：

broker.id=1

host.name=slave

zookeeper.connect=master:2181,slave:2181,slave1:2181

保存退出

1. 启动kafka

在master,slave,slave1分别启动kafka

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

zkCli.sh 进入zookeeper客户端

ls /

ls /brokers

ls /brokers/topic

1. 验证kafka

1）创建主题

bin/kafka-topic.sh --create --zookeeper master:2181 --replication-factor 1 --partitions 1--topic student

--replication-factor 3 副本因子(一般设为3)

--partitions 1 分区(可设为4)

--topic student 主题名字

2）创建生产者

bin/kafka-console-producer.sh --broker-list master:9092 --topic student

--broker-list master:9092 borker去哪里生产 9092为默认端口

这里的生产者是kafka自带的shell脚本启动的，但是生产者可以是flume，也可以是一段java代码

3）创建消费者

bin/kafka-console-consumer.sh --zookeeper master:2181 --topic student --from-beginning

--from-beginning 从现在开始消费

4) 键入下面的信息

hello word!

在消费者终端能显示上面所示即kafka集群搭建成功

5）删除消费者

bin/kafka-topics.sh --delete --zookeeper master:2181 --topic topicName

1. kafka缓存数据的位置

/tmp/kafka-logs

## Storm的安装部署

1. 安装的条件

Hadoop集群正常启动

ZooKeeper集群正常启动

1. 解压安装

unzip apache-storm-0.9.6.zip

1. 配置

1）环境变量配置

/home/xdl

vim ~/.bash\_profile

export STORM\_HOME=/home/xdl/apache-storm-0.9.6

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

2)配置storm.yaml核心配置文件

storm.zookeeper.servers:（Storm集群使用的Zookeeper集群地址）

- "master"

- "slave"

- "slave1"

nimbus.host=master（Storm集群Nimbus机器地址，负责资源分配和任务调 度）

1. 拷贝

将master节点上的storm安装文件和.bash\_profile 一起拷贝到 slave 和 slave1节 点

scp -r apache-storm-0.9.6 slave:~/

scp -r ~/.bash\_profile slave1:~/

在slave和slave1节点上分别执行

source ~/.bash\_profile

1. 启动

ssh master:

storm nimbus 日志被实时打印出来，当前页面一直处于日志的监测，如果关 掉当前窗口，那我们的nimbus就被kill掉了。

storm nimbus > /dev/null 2>&1 & nimbus ：负责资源分配和任务的调度

storm ui > /dev/null 2>&1 & nimbus

storm ui >/dev/null 2>&1 & core :代表的就是storm的ui界面

ssh slave/slave1

storm supervisor > /dev/null 2>&1 &

1. 验证(在三个节点分别输入JPS)

master

nimbus

core

slave

supervisor

slave1

supervisor

或者再浏览器上访问Storm WEBUI

http://master:8080

1. 案例：向Storm集群提交任务

向Storm集群提交Topology任务只需要运行JAR包中Topology即可

启动Topology（可在storm安装目录下的bin目录下也可在家目录下）

storm jar ./stor....jar storm.starter.ExclamationTopology wordcount

storm kill exclamation-topology 结束Topology

## Tomcat的安装部署

Tomcat需要JDK的支持，所以安装Tomcat前先安装JDK。

1. 首先到JDK官网下载与自己机器相应的JDK。

　　注意机器位数，Linux系统的话可以用uname -a命令查看系统信息，如果是Ubuntu的系统，下载后缀为.tar.gz的就行(文件夹中给到tomcat7.0的安装包)

接下来就是解压，然后配置JDK环境（JDK以及配置过，就不再多做解释）

运行java -version 可以查看相应版本的JDK

1. 下载Tomcat

　 在Tomcat官网找到自己要下载的Tomcat版本，解压后，在Tomcat目录的bin/下运行./startup.sh启动Tomcat

Tomcat托管的默认服务器端口是8080，所以在浏览器中输入http://localhost:8080/index.jsp能看到Tomcat欢迎界面，那么到此Tomcat安装成功了

ps:如果想在80端口启动服务，需要有root权限，1024以下端口都需要root权限；

修改方法：

修改apache-tomcat-7.0.57/conf/server.xml

<Connector port="80" protocol="HTTP/1.1"

connectionTimeout="20000"

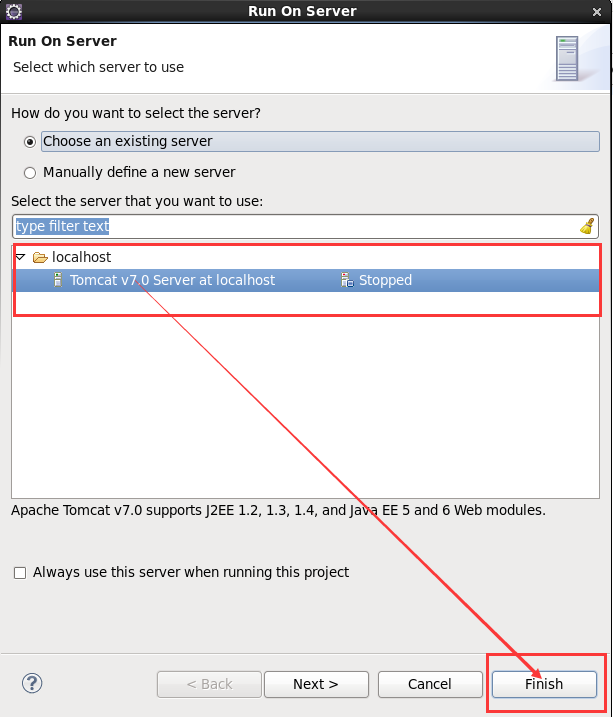
redirectPort="8443" />

如果这些都配置好了还不能访问，用tcpdump查看流量，很有可能是80端口被防火墙封了，需要开启。

iptables -A INPUT -p tcp --dport 80 -j ACCEPT

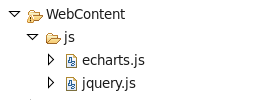
service iptables save

1. eclipse中启动tomcat



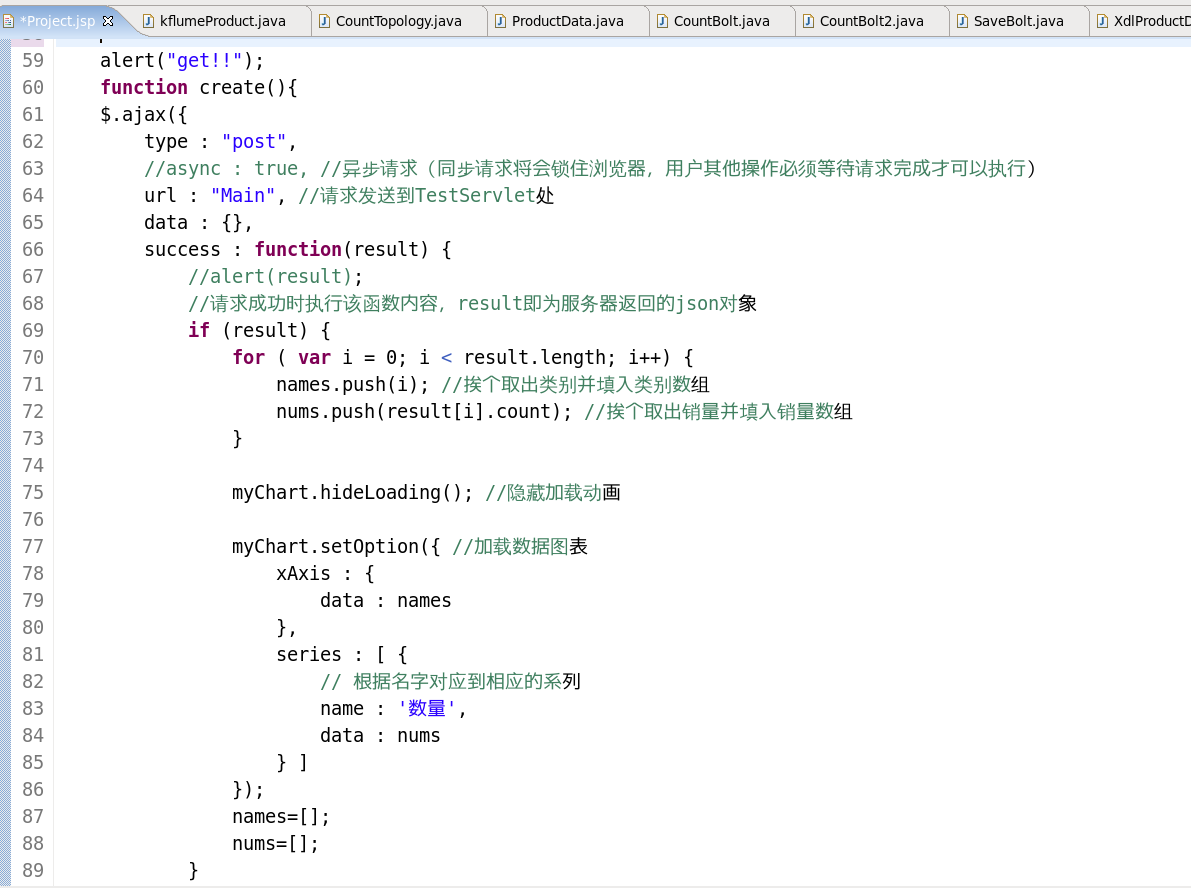
## Echarts的安装

1. echart.js下载、jquery,js下载

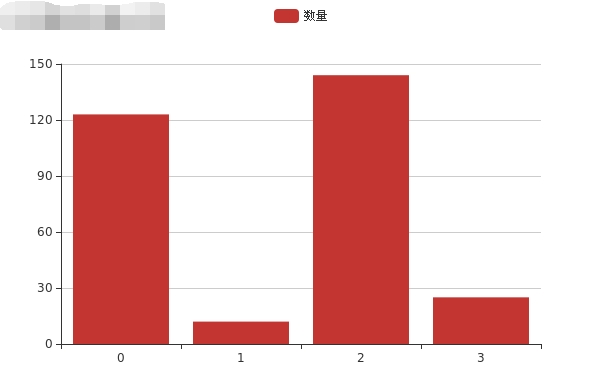


这两个js文件是整个前端的核心，这里也有提供，如果想下载更新版本去官网下载即可

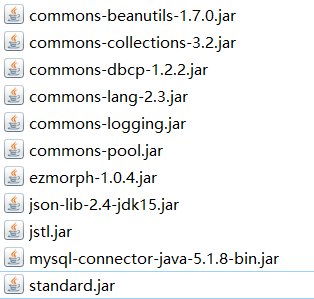
1. 编写前端jsp界面



Ajax从后台抓取数据实时展示如图所示：

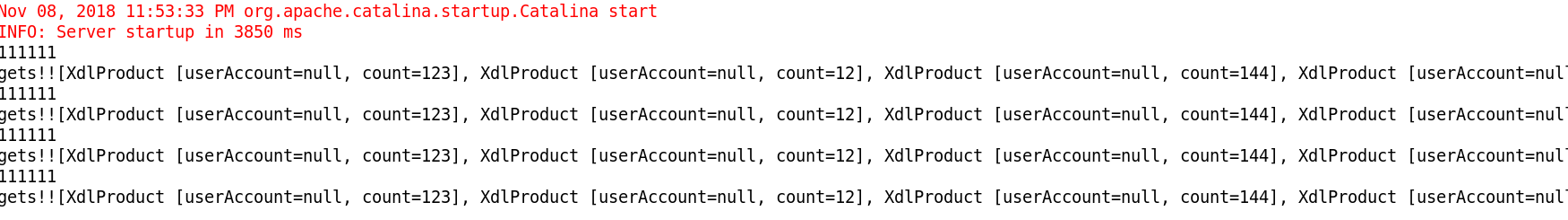


1. 准备所需依赖包做好ajax实时访问数据的准备工作



将这些所需依赖包放到linux目录下创建项目下的WEB-CONTENT下的lib下，bulid path即可

1. 后台代码实现访问数据库控制台显示实施抓取的数据



至此，此项目的详细部署工具已经准备就绪，接下来请观看下一个文档，开始项目的制作