# 服务器环境配置

所有配置均使用**root**用户操作完成

## 资源准备

三台服务器，操作系统为：**CentOS release 6.8**

IP为别为：

10.0.109.99

10.0.109.100

10.0.109.101

## 修改主机名

修改三台服务器主机名，修改**/etc/sysconfig/network**文件

master节点：**jttda-master**

slave节点1：**jttda-slave1**

slave节点2：**jttda-slave2**

每台服务器都修改**/etc/hosts**文件，添加如下域名映射

|  |
| --- |
| 10.0.109.99 jttda-master10.0.109.100 jttda-slave110.0.109.101 jttda-slave2 |

## 安装JDK

### 卸载openJDK

查看centos已安装的jdk

**rpm –qa | grep gcj**

**rpm –qa | grep jdk**

删除查询到的openJDK相关的安装包

**rpm -e –nodeps \*\*\***

### 安装oracle JDK

Oracle官网下载安装包**jdk-8u152-linux-x64.tar.gz**

将安装包解压到指定目录

**tar –zxvf jdk-8u152-linux-x64.tar.gz –C /usr/local/**

修改**/etc/profile**，添加如下配置

|  |
| --- |
| export JAVA\_HOME=/usr/local/jdk1.8.0\_152export PATH=$PATH:$JAVA\_HOME/bin |

## 配置无密码登陆

在master节点上面按照如下顺序执行：

**ssh-keygen**，一路回车到执行结束

**ssh-copy-id root@jttda-slave1**，输入**yes**，回车

**ssh-copy-id root@jttda-slave2**，输入**yes**，回车

## 关闭防火墙

关闭防火墙

**service iptables stop**

设置开机不启动

**chkconfig iptables off**

## NTP时间同步

设置提供ntp同步的服务器，修改各节点**/etc/ntp.conf**文件

开启ntp服务

**service ntpd start**

设置开机启动

**chkconfig ntpd on**

查看ntp同步状态

**ntpstat**

# Hadoop安装配置

## 获取安装包



如上图所示，选择所需版本，下载得到压缩包：**hadoop-2.7.5.tar.gz**

将压缩包解压到选定的目录：**tar -zxvf hadoop-2.7.5.tar.gz -C /home/hadoop/install/**

## 设置环境变量

在**每个节点**的**/etc/profile**文件中添加如下内容

|  |
| --- |
| export HADOOP\_HOME=/home/hadoop/install/hadoop-2.7.5export HADOOP\_COMMON\_LIB\_NATIVE\_DIR=$HADOOP\_HOME/lib/nativeexport HADOOP\_OPTS="-Djava.library.path=$HADOOP\_HOME/lib:$HADOOP\_COMMON\_LIB\_NATIVE\_DIR"export PATH=$PATH:$HADOOP\_HOME/bin:$HADOOP\_HOME/sbin |

## 创建目录

在每台服务器本地创建目录：

**mkdir -p /home/mmds/hadoopdata/dfs/data**

**mkdir -p /home/mmds/hadoopdata/dfs/name**

**mkdir -p /home/mmds/hadoopdata/tmp**

## 修改配置文件

进入**$HADOOP\_HOME/etc/hadoop**目录

### 修改slaves文件

添加DataNode主机名列表，每个主机名一行

|  |
| --- |
| jttda-slave1jttda-slave2 |

### 修改core-site.xml文件

|  |
| --- |
| <configuration> <property> <name>fs.defaultFS</name> <value>hdfs://jttda-master:9000</value> </property> <property> <name>hadoop.tmp.dir</name> <value>file:/data/hadoop/tmp</value> </property> <property> <name>io.file.buffer.size</name> <value>131702</value> </property></configuration> |

### 修改hdfs-site.xml文件

|  |
| --- |
| <configuration> <property> <name>dfs.namenode.name.dir</name> <value>file:/data/hadoop/dfs/name</value> </property> <property> <name>dfs.datanode.data.dir</name> <value>file:/data/hadoop/dfs/data</value> </property> <property> <name>dfs.replication</name> <value>2</value> </property> <property> <name>dfs.namenode.secondary.http-address</name> <value>jttda-master:9001</value> </property> <property> <name>dfs.webhdfs.enabled</name> <value>true</value> </property> <property> <name>dfs.balance.bandwidthPerSec</name> <value>10485760</value> </property></configuration> |

### 修改mapred-site.xml文件

修改模版文件名

**mv mapred-site.xml.template mapred-site.xml**

|  |
| --- |
| <configuration> <property> <name>mapreduce.framework.name</name> <value>yarn</value> </property> <property> <name>mapreduce.jobhistory.address</name> <value>jttda-master:10020</value> </property> <property> <name>mapreduce.jobhistory.webapp.address</name> <value>jttda-master:19888</value> </property> <property> <name>mapreduce.map.memory.mb</name> <value>2048</value> </property> <property> <name>mapreduce.reduce.memory.mb</name> <value>2048</value> </property> <property> <name>mapreduce.map.java.opts</name> <value>-Xmx1536M</value> </property> <property> <name>mapreduce.reduce.java.opts</name> <value>-Xmx1536M</value></property><property> <name>mapred.tasktracker.map.tasks.maximum</name> <value>8</value></property><property> <name>mapred.tasktracker.reduce.tasks.maximum</name> <value>12</value></property></configuration> |

### 修改yarn-site.xml文件

|  |
| --- |
| <configuration> <property> <name>yarn.resourcemanager.hostname</name> <value>jttda-master</value> </property> <property> <name>yarn.nodemanager.aux-services</name> <value>mapreduce\_shuffle</value> </property> <property> <name>yarn.nodemanager.resource.memory-mb</name> <value>51200</value> </property> <property> <name>yarn.scheduler.minimum-allocation-mb</name> <value>2048</value> </property> <property> <name>yarn.scheduler.maximum-allocation-mb</name> <value>10240</value> </property></configuration> |

### 修改hadoop-env.sh文件

修改如下两个配置项

|  |
| --- |
| export JAVA\_HOME=/usr/local/jdk1.8.0\_152export HADOOP\_HEAPSIZE=2048 |

### 修改yarn-env.sh文件

修改如下两个配置项

|  |
| --- |
| export JAVA\_HOME=/usr/local/jdk1.8.0\_152export YARN\_HEAPSIZE=2048 |

配置完成：

**将/usr/local/hadoop-2.7.5整个目录复制到各节点机器上，放在/usr/local/目录下**

 **scp –r /home/hadoop/install/hadoop-2.7.5 jttda-slave50:/home/hadoop/install/hadoop-2.7.5**

## 启动Hadoop

在jttda-master服务器上，进入$HADOOP\_HOME目录

格式化NameNode：**hdfs namenode -format**

启动NameNode和DataNode守护进程：**sbin/start-dfs.sh**

启动ResourceManager和NodeManager守护进程：**sbin/start-yarn.sh**

## 验证Hadoop

执行**jps**命令，有如下进程，说明Hadoop启动正常

|  |
| --- |
| # jps59024 Jps153505 ResourceManager152951 NameNode156206 RunJar153247 SecondaryNameNode |

# Hive安装配置

## 获取安装包



如上图所示，选择所需版本，下载得到压缩包：**apache-hive-2.2.0-bin.tar.gz**

将压缩包解压到选定的目录：

**tar -zxvf apache-hive-2.2.0-bin.tar.gz -C /home/hadoop/install/**

## 设置环境变量

在**master节点jttda-master**的**/etc/profile**文件中添加如下内容

|  |
| --- |
| export HIVE\_HOME=/usr/local/apache-hive-2.2.0-binexport PATH=$HIVE\_HOME/bin:$PATH |

## 创建metastore库

使用Oracle Database 11g作为hive的metastore数据库，创建对应用户并授权

|  |
| --- |
| CREATE USER **hive** IDENTIFIED BY **hive**;GRANT **SELECT\_CATALOG\_ROLE** TO **hive**;GRANT **CONNECT, RESOURCE** TO **hive**; |

在oracle官网下载JDBC驱动：**ojdbc6.jar**，将其放到**$HIVE\_HOME/lib**目录下

## 创建目录

在hdfs 中创建下面的目录 ，并且授权

|  |
| --- |
| hdfs dfs -mkdir -p /user/hive/warehousehdfs dfs -mkdir -p /user/hive/tmphdfs dfs -chmod -R 777 /user/hive/warehousehdfs dfs -chmod -R 777 /user/hive/tmp |

创建本地目录，进入**$HIVE\_HOME**目录

**mkdir log**

**mkdir execlog**

**mkdir tmp**

## 修改配置文件

进入**$HIVE\_HOME/conf**目录，复制默认配置文件并改名

|  |
| --- |
| cp hive-env.sh.template hive-env.shcp hive-default.xml.template hive-site.xmlcp hive-log4j2.properties.template hive-log4j2.propertiescp hive-exec-log4j2.properties.template hive-exec-log4j2.properties |

### 修改hive-env.sh文件

修改以下配置项

|  |
| --- |
| export JAVA\_HOME=/usr/local/jdk1.8.0\_152export HADOOP\_HOME=/usr/local/hadoop-2.7.5export HIVE\_HOME=/usr/local/apache-hive-2.2.0-binexport HIVE\_CONF\_DIR=/usr/local/apache-hive-2.2.0-bin/conf |

### 修改hive-site.xml文件

|  |
| --- |
| <configuration> <property> <name>hive.exec.scratchdir</name> <value>/user/hive/tmp</value> </property> <property> <name>hive.metastore.warehouse.dir</name> <value>/user/hive/warehouse</value> </property> <property> <name>hive.querylog.location</name> <value>/usr/local/apache-hive-2.2.0-bin/tmp/${user.name}</value> </property> <property> <name>javax.jdo.option.ConnectionURL</name> <value>jdbc:oracle:thin:@10.0.109.8:1521:sjzx8</value> </property> <property> <name>javax.jdo.option.ConnectionDriverName</name> <value>oracle.jdbc.driver.OracleDriver</value> </property> <property> <name>javax.jdo.option.ConnectionUserName</name> <value>hive</value> </property> <property> <name>javax.jdo.option.ConnectionPassword</name> <value>hive</value> </property></configuration> |

并在**hive-site.xml**中

所有**${system:java.io.tmpdir}**替换为**/usr/local/apache-hive-2.2.0-bin/tmp/**

所有**{system:user.name}**替换为**{user.name}**

### 修改hive-log4j2.properties文件

|  |
| --- |
| property.hive.log.dir = /usr/local/apache-hive-2.2.0-bin/log/${sys:user.name} |

### 修改hive-exec-log4j2.properties文件

|  |
| --- |
| property.hive.log.dir = /usr/local/apache-hive-2.2.0-bin/execlog/${sys:user.name} |

### 修改core-site.xml文件

进入**$HADOOP\_HOME/etc/hadoop**目录给beeline登录时的用户root配置权限，增加如下的配置项

|  |
| --- |
| <configuration> <property> <name>hadoop.proxyuser.root.groups</name> <value>root</value> </property> <property> <name>hadoop.proxyuser.root.hosts</name> <value>\*</value> </property></configuration> |

## 启动Hive

### 初始化

**schematool -dbType mysql -initSchema**

### CLI启动Hive

在命令行输入命令：**hive**

### 使用beeline

启动hiveserver2：**nohup hive --service hiveserver2** >/dev/null 2>&1 **&**

查看hiveserver2启动状态：**netstat -nptl | grep 10000**

启动beeline

在命令行输入命令：**beeline**

连接Hive：**!connect jdbc:hive2://localhost:10000 root root**

# Tez安装配置

## 获取安装包



如上图所示，选择所需版本，下载得到压缩包：apache-tez-0.9.0-bin.tar.gz

将压缩包解压到选定的目录：

**tar –zxvf apache- tez-0.9.0-bin.tar.gz –C /usr/local/**

## 上传运行包

在**apache-tez-0.9.0-bin/share**下有个**tez.tar.gz**压缩包，把它上传到hdfs上面

**hadoop fs -mkdir -p /user/tez**

**hadoop fs -put tez.tar.gz /user/tez**

## 修改配置文件

进入**$HADOOP\_HOME/etc/hadoop**目录

### 新建tez-site.xml文件

|  |
| --- |
| <configuration> <property> <name>tez.lib.uris</name> <value>/user/tez/tez.tar.gz</value> </property></configuration> |

### 修改mapred-site.xml文件

|  |
| --- |
| <configuration> <property> <name>mapreduce.framework.name</name> <value>yarn-tez</value> </property></configuration |

### 修改hadoop-env.sh文件

|  |
| --- |
| export TEZ\_HOME=/usr/local/apache-tez-0.9.0-binfor jar in `ls $TEZ\_HOME |grep jar`; do export HADOOP\_CLASSPATH=$HADOOP\_CLASSPATH:$TEZ\_HOME/$jardonefor jar in `ls $TEZ\_HOME/lib`; do export HADOOP\_CLASSPATH=$HADOOP\_CLASSPATH:$TEZ\_HOME/lib/$jardone |

## 测试Tez

在hdfs上创建**/tinput**和**/toutput**目录

准备测试文件

**echo "Hello World Hello Tez" > file01**

**echo "Hello World Goodbye Tez" > file02**

将测试文件file01和file02上传到hdfs上的**/tinput**目录

**hdfs dfs -mkdir /tinput**

**hdfs dfs -put file0\* /tinput**

执行如下命令测试tez是否配置成功，进去**$TEZ\_HOME**目录

**hadoop jar tez-examples-0.9.0.jar orderedwordcount /tinput /toutput**

# 测试用例

# 备注

将$HIVE\_HOME/lib/log4j-slf4j-impl-2.6.2.jar和$TEZ\_HOME/lib/slf4j-log4j12-1.7.10.jar删除，否则会与$HADOOP\_HOME/LIB/slf4j-log4j12-1.7.10.jar重复