**搭建hadoop集群及常用组件**

（初稿）

王海深

银基富力科技股份有限公司

2016-08-10

[配置hosts文件 3](#_Toc458760123)

[建立hadoop账号 3](#_Toc458760124)

[安装Java jdk 3](#_Toc458760125)

[各个结点之间的免密码ssh访问 4](#_Toc458760126)

[下载hadoop版本 4](#_Toc458760127)

[安装hadoop和yarn 4](#_Toc458760128)

[安装Flume 8](#_Toc458760129)

[安装zookeeper 8](#_Toc458760130)

[安装Kafka 10](#_Toc458760131)

[安装spark 11](#_Toc458760132)

[安装hive 12](#_Toc458760133)

[安装presto 14](#_Toc458760134)

# 配置hosts文件

1.修改hosts文件，执行下面命令：

vi /etc/hosts

192.168.100.133 h1

192.168.100.134 h2

2.发送到其他节点，执行下面命令：

scp /etc/hosts 192.168.100.134:/etc/hosts

# 建立hadoop账号

1.创建用户，执行下面命令：

useradd hadoop

2.设置密码，执行下面命令：

passwd hadoop, 然后输入两边密码即可。

3.使用hadoop账户，执行下面命令：

su hadoop

# 安装Java jdk

1.首先下载想要安装的jdk版本，这里使用的文件是jdk-8u92-linux-x64.tar.gz。

2.卸载系统自带jdk版本，执行命令：

rpm -qa | grep jdk

rpm –e 刚才的结果

3.将下载好的jdk拷贝到想要安装的路径，执行下面命令：

tar -zxvf jdk-8u92-linux-x64.tar.gz

4.修改环境变量，在 /etc/profile中通过下面命令添加java环境变量：

JAVA\_HOME=/data0/jdk1.8.0\_92

PATH=$JAVA\_HOME/bin:$PATH

CLASSPATH=.:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar

export JAVA\_HOME

export PATH

export CLASSPATH

5.重启，或者执行 source /etc/profile 让java的环境变量生效。

# 各个结点之间的免密码ssh访问

1. 在每台服务器上都执行ssh-keygen -t rsa生成密钥对:  
#ssh-keygen -t rsa

连续输入3次回车

2. 在每台服务器上生成密钥对后，将公钥复制到需要无密码登陆的服务器上：

举例如192.168.15.240，192.168.15.241，192.168.15.242这三台服务器需要做相互免密码登陆，在每台服务器生成 密钥对后，在每台服务器上执行ssh-copy-id命令，将公钥复制到其它两台服务器上

#ssh-copy-id -i  ~/.ssh/id\_rsa.pub [root@192.168.15.241](mailto:root@192.168.15.241)  
#ssh-copy-id -i  ~/.ssh/id\_rsa.pub [root@192.168.15.242](mailto:root@192.168.15.242)

本例中执行的命令：

#ssh-copy-id -i  ~/.ssh/id\_rsa.pub hadoop@192.168.100.134

3.类似的在其他节点上执行前两个步骤，完成所有节点之间的无密码访问

# 下载hadoop版本

本例中使用的hadoop文件是：hadoop-2.6.0.tar.gz

# 安装hadoop和yarn

1.将hadoop-2.6.0.tar.gz拷贝到要安装的路径下，执行下面命令：

tar –zxvf hadoop-2.6.0.tar.gz

2.建立一个软连接，执行命令：

ln –s hadoop-2.6.0 hadoop

3.进入hadoop目录，执行命令：

cd hadoop-2.6.0 && ls

下面涉及到配置的文件主要有以下7个：

etc/hadoop/hadoop-env.sh

etc/hadoop/yarn-env.sh

etc/hadoop/slaves

etc/hadoop/core-site.xml

etc/hadoop/hdfs-site.xml

etc/hadoop/mapred-site.xml

etc/hadoop/yarn-site.xml

4.进入指定目录，执行下面命令：

cd etc/Hadoop && ls

5.配置hadoop-env.sh文件

增加JAVA\_HOME，在文件最后加入下面一行代码：

export JAVA\_HOME=/data0/jdk1.8.0\_92

6.配置yarn-env.sh文件

增加JAVA\_HOME，在文件最后加入下面一行代码：

export JAVA\_HOME=/data0/jdk1.8.0\_92

7.配置slaves文件

在文件中添加节点，一个节点一行，如：

h1

h2

节点名称同hosts中配置的名称需对应。

8.配置core-site.xml文件

这里为hadoop的核心配置，如hdfs文件系统的端口是9000等，具体配置：

<configuration>

<property>

<name>fs.defaultFS</name>

<value>hdfs://h1:9000</value>

</property>

<property>

<name>io.file.buffer.size</name>

<value>131072</value>

</property>

<property>

<name>hadoop.tmp.dir</name>

<value>file:/hadoop/tmp</value>

<description>Abasefor other temporary directories.</description>

</property>

<property>

<name>hadoop.proxyuser.spark.hosts</name>

<value>\*</value>

</property>

<property>

<name>hadoop.proxyuser.spark.groups</name>

<value>\*</value>

</property>

</configuration>

注：若指定的目录不存在，需手动创建好。

9.配置hdfs-site.xml文件

主要用于增加hdfs配置信息，如namenode、datanode端口和目录位置等，具体配置：

<configuration>

<property>

<name>dfs.namenode.secondary.http-address</name>

<value>h1:9001</value>

</property>

<property>

<name>dfs.namenode.name.dir</name>

<value>file:/hadoop/dfs/name</value>

</property>

<property>

<name>dfs.datanode.data.dir</name>

<value>file:/hadoop/dfs/data</value>

</property>

<property>

<name>dfs.replication</name>

<value>3</value>

</property>

<property>

<name>dfs.webhdfs.enabled</name>

<value>true</value>

</property>

</configuration>

注：指定的文件路径最好先手动建好。

10.配置maprd-site.xml文件

该文件主要用于增加mapreduce配置，如使用yarn框架等，具体配置：

<configuration>

<property>

<name>mapreduce.framework.name</name>

<value>yarn</value>

</property>

<property>

<name>mapreduce.jobhistory.address</name>

<value>h1:10020</value>

</property>

<property>

<name>mapreduce.jobhistory.webapp.address</name>

<value>h1:19888</value>

</property>

</configuration>

11.配置yarn-site.xml文件

该文件主要用于增加yarn功能，具体配置：

<configuration>

<!-- Site specific YARN configuration properties -->

<property>

<name>yarn.nodemanager.aux-services</name>

<value>mapreduce\_shuffle</value>

</property>

<property>

<name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>

<value>org.apache.hadoop.mapred.ShuffleHandler</value>

</property>

<property>

<name>yarn.resourcemanager.address</name>

<value>h1:8032</value>

</property>

<property>

<name>yarn.resourcemanager.scheduler.address</name>

<value>h1:8030</value>

</property>

<property>

<name>yarn.resourcemanager.resource-tracker.address</name>

<value>h1:8035</value>

</property>

<property>

<name>yarn.resourcemanager.admin.address</name>

<value>h1:8033</value>

</property>

<property>

<name>yarn.resourcemanager.webapp.address</name>

<value>h1:8088</value>

</property>

</configuration>

12.将配置好的hadoop文件copy到其他节点。

使用命令：

scp –r hadoop-2.6.0 hadoop@h2:/data0/

13.验证

a.格式化namenode

执行命令：

cd Hadoop-2.6.0

.bin/hdfs namenode –format

b.启动hdfs

执行命令：

./sbin/start-dfs.sh

结果验证：

执行命令：jps

执行结果：结果显示中有NameNode和SecondaryNameNode

c.停止hdfs

执行命令：

./sbin/stop-hdfs.sh

d.启动yarn

执行命令：

./sbin/start-yarn.sh

结果验证：

执行命令：jps

执行结果：结果显示中含有ResourceManager

e.停止yarn

执行命令：

./sbin/stop-yarn.sh

f.查看集群状态

执行命令：

./sbin/hdfs dfsadmin –report

14.url查看

a.查看hdfs：<http://192.168.100.133:50070/>

b.查看RM：<http://192.168.100.133:8088>/

# 安装Flume

1.将flume-ng-1.5.0-cdh5.4.9.tar.gz文件拷贝到指定路径下，执行命令：

cp flume-ng-1.5.0-cdh5.4.9.tar.gz /data0

2.解压安装，执行命令：

tar -zxvf flume-ng-1.5.0-cdh5.4.9.tar.gz

3.进入生成的apache-flume-1.5.0-cdh5.4.9-bin目录，执行下面命令：

cd apache-flume-1.5.0-cdh5.4.9-bin

4.测试安装成功与否，执行命令：

sh /bin/flume-ng version

如出现返回表示flume已经安装成功：

Flume 1.5.0-cdh5.4.9

Source code repository: https://git-wip-us.apache.org/repos/asf/flume.git

Revision: 7499877e237d4a3e1a7e0c23e43d8e55c3f1c870

Compiled by jenkins on Tue Dec 15 18:21:36 PST 2015

From source with checksum 17d148c7ff2d574c5b4d5e5d86952b0f

# 安装zookeeper

1. 将zookeeper-3.4.5-cdh5.7.0.tar.gz文件拷贝到指定路径下，执行命令：

cp zookeeper-3.4.5-cdh5.7.0.tar.gz /data0

2.解压安装，执行命令：

tar -zxvf zookeeper-3.4.5-cdh5.7.0.tar.gz

3.进入解压生成的目录，执行命令：

cd zookeeper-3.4.5-cdh5.7.0/

4.修改配置文件，执行命令：

cd conf && ls

cp zoo\_sample.cfg zoo.cfg

vi zoo.cfg

具体的配置：

# The number of milliseconds of each tick

tickTime=2000

# The number of ticks that the initial

# synchronization phase can take

initLimit=10

# The number of ticks that can pass between

# sending a request and getting an acknowledgement

syncLimit=5

# the directory where the snapshot is stored.

# do not use /tmp for storage, /tmp here is just

# example sakes.

dataDir=/tmp/zookeeper

#The log

dataLogDir=/tmp/zookeeper/logs

# the port at which the clients will connect

clientPort=2181

# Be sure to read the maintenance section of the

# administrator guide before turning on autopurge.

# http://zookeeper.apache.org/doc/current/zookeeperAdmin.html#sc\_maintenance

# The number of snapshots to retain in dataDir

#autopurge.snapRetainCount=3

# Purge task interval in hours

# Set to "0" to disable auto purge feature

#autopurge.purgeInterval=1

server.1=192.168.100.134:2888:3888

server.2=192.168.100.133:2888:3888

5.把zookeeper目录发送到其他节点，执行命令：

scp –r zookeeper-3.4.5-cdh5.7.0 h2:/data0

6.找到zoo.cfg配置的dataDir目录，在目录下面添加一个“myid”的文件，把zoo.cfg配置的server list中的id值写入并保存，执行的命令：

cd /tmp/zookeeper

vi myid

输入1，然后保存退出。

在其他的节点中也执行上面操作，并确保所有节点填写的值不存在冲突。

7.启动每个节点的zookeeper

分别在每个节点上执行命令：

cd zookeeper-3.4.5-cdh5.7.0/

./bin/zkServer.sh start

8.查看集群状态

在每个节点上查看其在zookeeper集群中所扮演的角色，执行命令：

./bin/zkServer.sh status

结果：

Master会显示 Mode：leader，

Slaves 会显示 Mode：follower。

# 安装Kafka

1.将kafka\_2.11-0.10.0.0.tgz文件拷贝到指定路径下，执行命令：

cp kafka\_2.11-0.10.0.0.tgz /data0

2.解压安装，执行命令：

tar -zxvf kafka\_2.11-0.10.0.0.tgz

3.进入解压生成的目录，执行命令：

cd kafka\_2.11-0.10.0.0/

4.修改配置文件，执行命令：

vi config/server.properties

其中需要配置的属性有broker.id(标示当前server在集群中的id，从0开始)，port，host.name(当前的server host name)，zookeeper.connect(连接的zookeeper集群)，log.dirs(log的存储目录，记得对应的去建立这个目录)等。

详细的配置：

broker.id=0

listeners=PLAINTEXT://h2:9092

zookeeper.connect=h1:2181,h2:2181

zookeeper.connection.timeout.ms=1000000

5.发送配置好的kafka文件到其他节点，执行命令：

scp –r kafka\_2.11-0.10.0.0 h2:/data0

6.修改每个节点上的配置文件，主要修改broker.id 和 host.name 属性,其中broker.id从0开始递增，每台server必须唯一。

7.启动kafka集群，执行命令：

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

其中，需要在启动kafka集群前线启动zookeeper集群。

8.执行下面的测试命令：

a.创建topic

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

b.显示所有topic

./bin/kafka-topics.sh --list --zookeeper 192.168.100.134:2181

c.在一个server上创建一个producer，并输入一些测试信息，执行命令：

./bin/kafka-console-producer.sh --broker-list 192.168.100.134:9092,192.168.100.133:9092 -topic foo

d.在另外一个server上创建一个consumer，接受producer产生的消息，执行命令：

./bin/kafka-console-consumer.sh --zookeeper 192.168.100.134:2181,192.168.100.133:2181 --from-beginning -topic foo

e.查看kafka offset

./kafka-run-class kafka.tools.GetOffsetShell --broker-list 172.31.18.12:9092 --topic cisco --time -1

若能正常接收到producer产生的消息表明kafka集群已经能正常工作。

# 安装spark

1.从官方网站下载scala，本例使用的版本是scala-2.11.8.tgz，并将其拷贝到指定目录，执行命令：

cp scala-2.11.8.tgz /data0

2.解压安装scala，执行命令：

tar –zxvf scala-2.11.8.tgz

3.在/etc/profile中添加SCALA\_HOME的环境变量，执行命令：

vi /etc/profile

在文件尾部追加代码：

SCALA\_HOME=/data0/scala-2.11.8

PATH=$SCALA\_HOME/bin:$PATH

export SCALA\_HOME

export PATH

4.将spark-1.6.2-bin-hadoop2.6.tgz文件拷贝到指定路径下，执行命令：

cp spark-1.6.2-bin-hadoop2.6.tgz /data0

5.解压安装，执行命令：

tar -zxvf spark-1.6.2-bin-hadoop2.6.tgz

6.进入解压生成的目录，执行命令：

cd spark-1.6.2-bin-hadoop2.6/

7.建立软连接，执行命令：

ln –s spark-1.6.2-bin-hadoop2.6 spark

8.在/etc/profile中添加SPARK\_HOME的环境变量，执行命令：

vi /etc/profile

在文件尾部追加代码：

SPARK\_HOME=/data0/spark

PATH=$SPARK\_HOME/bin:$SPARK\_HOME/sbin:$PATH

export SPARK\_HOME

export PATH

9.修改配置文件，执行命令：

cp conf/spark-env.sh.template conf/spark-env.sh

vi conf/spark-env.sh

主要修改：

JAVA\_HOME=/data0/jdk1.8.0\_92

HADOOP\_HOME=/data0/hadoop-2.6.0

SCALA\_HOME=/data0/scala-2.11.8

export JAVA\_HOME

export HADOOP\_HOME

export SCALA\_HOME

10.修改slaves配置文件，执行命令：

vi conf/slaves

将计算节点的主机名添加到该文件，一行一个。本例的slaves文件：

h1

h2

11.将profile、scala、spark几个目录或文件发送到其他节点，执行命令：

scp /etc/profile h2:/etc

scp –r /data0/spark h2:/data0

scp –r /data0/ scala-2.11.8 h2:/data0

12.启动集群，通过在master执行脚本程序完成集群的启动，执行命令：

./sbin/start-master.sh

./sbin/start-slaves.sh

如可以正常启动的话，可在Master节点上看到名为Master的进程，在子节点上看到名为Worker的进程。

13.通过web端口访问spark集群，spark集群的默认端口是8080，访问网址：

<http://192.168.100.133:8080/>

# 安装hive

1.准备工作，需提前安装mysql

2.使用root用户登录mysql，执行命令：

mysql -uroot –p\*\*\*\*，其中\*\*\*\*表示设置的密码

3.创建hive用户、数据库等，执行命令：

insert into mysql.user(Host,User,Password) values("localhost","hive",password("\*\*\*\*"));

create database hive;

grant all on hive.\* to hive@'%'  identified by 'hive';

grant all on hive.\* to hive@'localhost'  identified by '\*\*\*\*';

flush privileges;

其中\*\*\*\*表示所设置的密码；

4.退出mysql，并验证hive用户，执行命令：

exit

mysql -uhive –p\*\*\*\*

show databases;

如显示中存在hive条目表示创建成功；

5.拷贝hive文件到指定目录，执行命令：

cp apache-hive-2.1.0-bin.tar.gz /data0

6.解压安装包，执行命令：

tar –zxvf apache-hive-2.1.0-bin.tar.gz

7.建立软连接，执行命令：

ln –s apache-hive-2.1.0-bin hive

8.增加hive的环境变量，执行命令：

vi /etc/profile

在文件末尾增加下面几行代码，

HIVE\_HOME=/data0/hive

PATH=$HIVE\_HOME/bin:$PATH

export HIVE\_HOME

9.修改hive-site.xml文件，进行参数配置，执行命令：

vi conf/hive-site.xml

该文件中内容很多，主要修改的内容如下：

<property>

   <name>javax.jdo.option.ConnectionURL </name>

   <value>jdbc:mysql://localhost:3306/hive </value>

</property>

<property>

   <name>javax.jdo.option.ConnectionDriverName </name>

   <value>com.mysql.jdbc.Driver </value>

</property>

<property>

   <name>javax.jdo.option.ConnectionPassword </name>

   <value>hive </value>

</property>

<property>

   <name>hive.hwi.listen.port </name>

   <value>9999 </value>

   <description>This is the port the Hive Web Interface will listen on </descript ion>

</property>

<property>

   <name>datanucleus.autoCreateSchema </name>

   <value>true</value>

</property>

<property>

   <name>datanucleus.fixedDatastore </name>

   <value>false</value>

</property>

</property>

  <property>

    <name>javax.jdo.option.ConnectionUserName</name>

    <value>hive</value>

    <description>Username to use against metastore database</description>

  </property>

  <property>

    <name>hive.exec.local.scratchdir</name>

    <value>/data0/hive/iotmp</value>

    <description>Local scratch space for Hive jobs</description>

  </property>

  <property>

    <name>hive.downloaded.resources.dir</name>

    <value>/data0/hive/iotmp</value>

    <description>Temporary local directory for added resources in the remote file system.</description>

  </property>

  <property>

    <name>hive.querylog.location</name>

    <value>/data0/hive/iotmp</value>

    <description>Location of Hive run time structured log file</description>

  </property>

10.拷贝mysql-connector-java-5.1.6-bin.jar 到hive 的lib，执行命令：

cp /\*\*\*/mysql-connector-java-5.\*.\*-bin.jar /data0/hive/lib

11.把jline-2.12.jar拷贝到hadoop相应的目录下，替代jline-0.9.94.jar，否则启动会报错，执行命令：

cp /data0/hive/lib/lib/jline-2.12.jar /data0/hadoop-2.6.0/share/hadoop/yarn/lib/

mv /data0/hadoop-2.6.0/share/hadoop/yarn/lib/jline-0.9.94.jar /data0/hadoop-2.6.0/share/hadoop/yarn/lib/jline-0.9.94.jar.bak

12.创建hive临时文件夹，执行命令：

mkdir /data0/hive/iotmp

13.初始化元数据库，执行命令：

cd /data0/hive/bin

./schematool -initSchema –dbType mysql

14.启动hadoop后，对hive进行测试，执行命令：

hive

show databases;

# 安装presto

1.下载presto到指定目录，执行命令：

cp presto-server-0.151.tar.gz /data0/

2.解压安装presto，执行命令：

tar –zxvf presto-server-0.151.tar.gz

3.进入解压后生成的目录，并创建名为etc的文件夹，执行命令：

cd presto-server-0.151

mkdir etc

4.在etc中添加如下几个配置文件，这些文件全部为手动创建：

config.properties

jvm.config

node.properties

catalog/hive.properties

下面是每个配置文件的具体内容：

a. catalog/hive.properties

connector.name=hive-hadoop2

hive.metastore.uri=thrift://192.168.100.134:9083

hive.config.resources=/data0/hadoop/etc/hadoop/core-site.xml,/data0/hadoop/etc/hadoop/hdfs-site.xml

b. node.properties

node.environment=production

node.id=ffffffff-ffff-ffff-ffff-ffffffffffff-node-h2

node.data-dir=/data0/presto/data

c. config.properties

coordinator=true

node-scheduler.include-coordinator=true

http-server.http.port=8089

discovery-server.enabled=true

discovery.uri=http://h2:8089

d. jvm.config

-server

-Xmx12G

-XX:+UseConcMarkSweepGC

-XX:+ExplicitGCInvokesConcurrent

-XX:+CMSClassUnloadingEnabled

-XX:+AggressiveOpts

-XX:+HeapDumpOnOutOfMemoryError

-XX:ReservedCodeCacheSize=150M

-XX:OnOutOfMemoryError=kill -9 %p

5.启动presto，执行命令：

cd /data0/ presto-server-0.151/bin

./launcher run

最后打印显示如下字样表示presto server已经启动成功，

======== SERVER STARTED ========

6.查看presto web界面，访问网址是：

<http://192.168.100.134:8089/>

进去后可查看当前presto工作组的基本状态以及执行的query记录等。