Hadoop2.6.3+HBase1.1.2+ZooKeeper3.4.6 +Scala2.11.4+Spark1.6.1 CentOs6.5 分布式环境搭建说明书

目录

准备工作

1. 软件

JDK	jdk-8u66-linux-x64.rpm
hadoop 安装文件	hadoop-2.6.3.tar.gz
hbase 安装文件	hbase-1.1.2-bin.tar.gz
zookeeper 安装文件	zookeeper-3.4.6.tar.gz
scala 安装文件	scala-2.11.4.tgz
spark 安装文件	spark-1.6.1-bin-hadoop2.6.gz
xshell5	xshell5.exe

2. 硬件

5 台　8T 存储
3 台　2T 存储
内存　32G

3. 系统

CentOs6.5

分布式系统结构

主机名	硬盘	内存	IP
slave1	1.8T	32G	192.168.55.24
slave2	1.8T	32G	192.168.55.25
slave3	1.8T	32G	192.168.55.26
slave4	7.3T	32G	192.168.55.27
slave5	7.3T	32G	192.168.55.28
slave6	7.3T	32G	192.168.55.29
slave7	7.3T	32G	192.168.55.30
master	7.3T	32G	192.168.55.31

部署步骤

1. 用户的建立与权限

本次部署工作的所有操作均要在同一 linux 系统账号下进行，避免出现权限问题。

1. **创建用户**_hadoop

```
useradd _hadoop
```

2. **修改密码**

```
passwd _hadoop
```

3. **赋予** root **权限**

修改/etc/sudoers 文件，找到下面一行，在 root 下面添加一行，如下所示：

```
## Allow root to run any commands anywhere
root ALL=(ALL) ALL
_hadoop ALL=(ALL) ALL
```

修改完毕，现在可以用_hadoop 帐号登录，必要时可使用 sudo 命令获取 root 权限。

2. 修改主机名以及 Host 映射关系

编辑/etc/sysconfig/network 文件来修改 hostname，例如在 master 节点上

```
hostname=master
```

分别在所有节点上添加 host 映射关系

```
vim /etc/hosts
```

添加内容示例如下

```
192.168.55.31 master
```

3. 安装 JAVA

1. **版本**

jdk-8u66-linux-x64

2. **安装**

解压缩.rpm 安装包到一固定统一的路径下

rpm -ivh jdk-8u66-linux-x64.rpm

3. **配置环境变量**

修改/etc/profile 文件，在末尾添加如下内容

```
export JAVA_HOME=/usr/java/jdk-1.8.0_66
export PATH= $JAVA_HOME/bin:$JRE_HOME/bin: $PATH
```

执行 /etc/profile 使设置生效

```
source /etc/profile
```

4. **测试**

根目录下测试 JAVA 安装情况

```
java –version
```

4. 安装 SSH

1. **安装**

在所有机器上全部安装 ssh

```
yum install openssh-server
```

2. **配置无密码登录**

为使得 master 可以控制其余的 slaves，需要 master 和 slave 互相无密码访问，且可自访问。

在所有主机上分别在/home/_hadoop/下创建文件夹.ssh

```
mkdir /home/_hadoop/.ssh
```

并生成密钥

```
ssh-keygen –t rsa
```

按回车直到生成密钥图，此时在.ssh 目录下会生成 id_rsa 和 id_rsa.pub 两个文件。

接下来在 master 上生成 authorized_keys 文件

```
cp id_rsa.pub authorized_keys
```

将上述生成的 authorized_keys 文件复制到各个 slave 上即可，可使用 scp 命令。

```
cat id_rsa.pub | ssh _hadoop@slave1 'cat - >> /home/_hadoop/.ssh/authorized_keys'
```

最后在各个 slave 上生成 authorized_keys，并将其添加到 master 的 authorized_key 下。

3. **测试**

1）master 到 master

```
ssh master
```

2）master 到 slave

```
ssh slave1
```

显示

```
[_hadoop@master hadoop-2.6.3]$ ssh slave2
Last login: Fri Jan 22 23:33:33 2016 from 192.168.55.1
```

则表示登陆成功
注：在第一次登陆时可能需要密码，之后的登陆不再需要；另外登录时需注意在_hadoop 用户状态下。登陆测试结束后注意要 exit，防止以后的操作影响刚刚登陆的主机

5. 安装 hadoop

1. 版本

Hadoop-2.6.3

2. 安装

在/usr 目录下解压安装包并重命名为 hadoop

```
tar -xzvf hadoop-2.6.3.tar.gz
mv hadoop-2.6.3 hadoop
```

3. 配置

在/home/_hadoop/.bashrc 文件中添加如下命令

```
export HADOOP_HOME=/usr/ hadoop-2.6.3
export PATH=$HADOOP_HOME/bin:$PATH
```

立即执行生效

```
source .bashrc
```

在所有主机 hadoop 安装目录下修改 etc/Hadoop/下的配置文件
Core-site.xml

```
<configuration>

<property>
      <name>hadoop.tmp.dir</name>
      <value>/app/hadoop/tmp</value>
      <description>A base for other temporary directories.</description>
</property>

<property>
      <name>fs.default.name</name>
      <value>hdfs://master:54310</value>
</property>

</configuration>
```

hadoop-env.sh 中要修改 JAVA_HOME，注意此处必须直接写路径，写 $JAVA_HOME 可能无效

```
export JAVA_HOME=/usr/java/jdk1.8.0_66
```

hdfs-site.xml

```
<configuration>

<property>
      <name>dfs.replication</name>
      <value>3</value>
      <description>Default block replication.
      The actual number of replications can be specified when the file is created.
      The default is used if replication is not specified in create time.
      </description>
</property>

<property>
    <name>dfs.datanode.address</name>
    <value>0.0.0.0:50009</value>
<!-- DN 的服务监听端口，端口为 0 的话会随机监听端口，通过心跳通知 NN -->
</property>

</configuration>
```

mapred-site.xml

```
<configuration>

<property>
<name>mapreduce.framework.name</name>
                <value>yarn</value>
                <final>true</final>
</property>

<property>
<name>mapreduce.jobtracker.http.address</name>
                <value>master:50030</value>
</property>

<property>
<name>mapreduce.jobhistory.address</name>
                <value>master:10020</value>
</property>

<property>
```

```
<name>mapreduce.jobhistory.webapp.address</name>
                <value>master:19888</value>
</property>

<property>
<name>mapred.job.tracker</name>
      <value>http://master:54311</value>
      <description>The host and port that the MapReduce job tracker runs
      at.    If "local", then jobs are run in-process as a single map
      and reduce task.
      </description>
</property>

</configuration>
```

在主机 master 上修改 hadoop 安装目录下的 slaves 文件，添加 slave 名单。

```
slave1
slave2
…
slave7
```

4. **测试**

格式化文件系统，在 hadoop 安装目录下输入命令

```
bin/hadoop namenode –format
```

启动 hadoop 集群，在 master 上，hadoop 安装目录下输入命令

```
sbin/start-all.sh
```

使用命令 jps 可查看进程启动情况。
正常情况下 master 节点上执行的进程有
namenode/secondarynamenode/resourcemanager
slave 节点上执行的进程有
nodemanager/datanode
停止 hadoop 集群可在 master 上，hadoop 安装目录下

```
sbin/stop-all.sh
```

启动成功后可在浏览器上访问 master 的 ip 地址，端口号 8088，实现集群节点管理。

6. 安装 ZooKeeper

1. **版本**

zookeeper-3.4.6

2. **安装**

解压 zookeeper 安装文件

```
tar -xzvf zookeeper3.4.6.tar.gz
```

3. **配置**

拷贝 zoosample.cfg 文件为 zoo.cfg，并编辑如下

```
dataDir=/home/hadoop/zookeeper/data
server.1=slave1:2888:3888
server.2= slave2:2888:3888
server.3= slave3:2888:3888
server.4= slave4:2888:3888
server.5= slave5:2888:3888
server.6= slave6:2888:3888
server.7= slave7:2888:3888
```

4. **修改** myid

在 dataDir 目录下新建 myid 文件，输入当前主机的 id（1,2,3……）

5. **启动** zookeeper **集群**

在每个 zookeeper 集群的节点上执行启动服务的脚本

```
/home/_hadoop/zookeeper/bin/zkServer.sh start
```

6. **测试**

输入如下命令可显示当前节点角色 fllower 或 leader

```
/home/_hadoop/zookeeper/bin/zkServer.sh status
```

7. 安装 HBase

1. **版本**

hbase-1.1.2

2. **安装**

解压 hbase 安装文件

tar -xzvf hbase1.0.0.tar.gz

3. **配置**

修改配置文件

hbase-env.sh

```
export JAVA_HOME=/usr/java/jdk1.8.0_51
export HBASE_CLASSPATH=/home/hadoop/hadoop/etc/hadoop/
export HBASE_MANAGES_ZK=false
```

hbase-site.xml

```
<configuration>
<property>
```

```
<name>hbase.rootdir</name>
<value>hdfs://master:54310/hbase</value>
</property>

<property>
<name>hbase.master</name>
<value>master</value>
</property>

<property>
<name>hbase.cluster.distributed</name>
<value>true</value>
</property>

<property>
<name>hbase.zookeeper.quorum</name>
<value> slave1,slave2,slave3,slave4,slave5,slave6,slave7</value>
</property>

<property>
<name>hbase.zookeeper.property.clientPort</name>
<value>2181</value>
</property>

<property>
<name>hbase.zookeeper.property.dataDir</name>
<value>/home/_hadoop/zookeeper/store/data</value>
</property>
<property>
<name>hbase.regionserver.ipc.address</name>
<value>0.0.0.0</value>
</property>
<property>
<name>zookeeper.session.timeout</name>
<value>60000000</value>
</property>
<property>
<name>dfs.support.append</name>
<value>true</value>
</property>
</configuration>
```

4. 修改 regionservers

在 regionservers 文件中添加 slaves 列表

```
slave1
slave2
……
```

5. **测试**

使用 jps 命令查看 hbase 进程信息，可见 master 上有进程 HMaser，slave 上有进程 HRegionServer。

8. 启动集群

1. **启动** Zookeeper

```
zookeeper/bin/zkServer.sh start
```

2. **启动** hadoop

```
hadoop/sbin/start-all.sh
```

3. **启动** hbase

```
hbase/bin/start-hbase.sh
```

4. **查看进程信息**

```
jps
```

若启动正常，master 和 slave 进程列表如下

```
SecondaryNameNode       # hadoop 进程
NameNode                # hadoop 进程
ResourceManager         # hadoop 进程
HMaster                 # hbase  进程
```

```
QuorumPeerMain          # zookeeper 进程
DataNode                # hadoop  进程
NodeManager             #hadoop  进程
HRegionServer           # hbase slave 进程
```

9. 安装 Scala

1. **版本**

scala-2.11.4.tgz

2. **安装**

解压安装包到_hadoop/目录下

```
tar -zxf scala-2.11.4.tgz
```

并重命名

```
mv scala-2.11.4 scala
```

3. **配置**

修改/etc/profile 文件，添加如下语句

```
export SCALA_HOME=/home/_hadoop/scala
export PATH=$SCALA_HOME/bin:$PATH
```

保存后退出，并 source 生效

```
source profile
```

在 master 上完成上述安装和配置后使用 scp 命令将整个 spark 目录复制到各个 slave 上即可。

4. **测试**

```
scala –version
```

显示

```
Scala code runner version 2.11.4 -- Copyright 2002-2013, LAMP/EPFL
```

10. 安装 Spark

1. **版本**

spark-1.6.1-bin-hadoop2.6.gz

2. **安装**

解压安装包到_hadoop/目录下

```
tar -zxf spark-1.6.1-bin-hadoop2.6.gz
```

并重命名

```
mv spark-1.6.1-bin-hadoop2.6 spark
```

3. **配置**

修改 spark/conf/spark-env.sh

```
export HADOOP_HOME=/home/_hadoop/hadoop
export JAVA_HOME=/usr/java/jdk1.8.0_66
export HADOOP_CONF_DIR=/home/_hadoop/hadoop/etc/hadoop
export SCALA_HOME=/home/_hadoop/scala
export SPARK_HOME=/home/_hadoop/spark
export SPARK_MASTER_IP=192.168.55.31
export SPARK_MASTER_PORT=7077
export SPARK_MASTER_WEBUI_PORT=8099
export SPARK_WORKER_CORES=3
export SPARK_WORKER_INSTANCES=1
export SPARK_WORKER_MEMORY=8G
```

```
export SPARK_WORKER_WEBUI_PORT=8081
export SPARK_EXECUTOR_CORES=1
export SPARK_EXECUTOR_MEMORY=1G
export
SPARK_CLASSPATH=/home/_hadoop/spark/lib/sequoiadb-driver-1.12.ja
r:/home/_hadoop/spark/lib/spark-sequoiadb_2.11.2-1.12.jar
export
LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:$HADOOP_HOME/lib/native
```

修改 spark/sconf/slaves 文件

```
master
slave1
slave2
……
```

同样地，在 master 上完成上述安装和配置后使用 scp 命令将整个 spark 安装包复制到各个 slave 上。

4. **测试**

在 master 上分别启动 master 和 slaves

master

```
sbin/start-master.sh
```

slaves

```
sbin/start-slaves.sh
```

使用 jps 可查看进程信息

master 上有进程 master

slave 上有进程 worker

注意事项

1. 不同主机上的相同配置文件可以在一个主机上完成后使用 scp 命令复制到其余主机，简化操作。

```
scp [OPTIONS] file_source file_target
```

例：scp /home/_hadoop/hbase　_hadoop@slave1:/home/_hadoop/

2. hadoop 集群中 master 节点若出现 namenode 或 datanode 启动失败问题，可先停止集群，尝试删除 data、name 文件夹，并新建，重新 format，再启动集群。或者手动将 data 目录下的 clusterid 与 name 目录下的 clusterid 统一。

3. 在整个过安装过程中，防火墙应保持关闭状态，防止节点间通讯受到影响。在日后集群的运行过程中应开启防火墙，设置防火墙规则。

4. hadoop 每次修改配置文件后都需要重新格式化，否则进程启动有可能出现问题

5. 从 Windows 上传和下载文件到服务器可使用 sz/rz 命令

若 rz 命令无效，可使用如下命令安装相关服务

```
yum –y install lrzsz
```

6. 一些端口

1. hadoop 相关端口
   192.168.55.31:50070/
   192.168.55.31:8088/
2. hbase 相关端口
   192.168.55.31:60010/
3. spark 相关端口
   192.168.55.31:8099/