

CentOS 下 Hadoop-0.20.2 集群配置文档

张鹏

一、集群网络环境介绍

集群包含三个节点：1 个 namenode，2 个 datanode，节点之间局域网连接，可以相互 ping 通。节点 IP 地址分布如下：

Namenode: zhangpeng@192.168.90.91

Datanode1: zhangruixue@192.168.90.94

Datanode2: guanxiansun@192.168.90.205

三台节点上均是 CentOS 系统，并且有一个相同的用户 hadoop(具有管理员权限)，在 /home/hadoop/ 目录下均有一个 hadoopinstall 目录，用于存放 hadoop-0.20.2 安装文件，目录结构是 /home/hadoop/hadoopinstall/hadoop-0.20.2。

在 /etc/hosts 上添加主机名和相应的 IP 地址：

```
192.168.90.129  zhangpeng
```

```
192.168.90.94  zhangruixue
```

```
192.168.90.205  guanxiansun
```

二、SSH 无密码验证配置

Hadoop 需要使用 SSH 协议，namenode 使用 SSH 无密码登录并启动 datanode 进程，同样原理，datanode 上也能使用 SSH 无密码登录到 namenode（这样也能在 datanode 上进行启动或关闭 hadoop）。

1、安装和启动 SSH 协议

所有机器上安装 SSH 协议并启动服务，在所有机器上执行以下命令：

```
$ sudo yum install ssh  安装 SSH 协议
```

```
$ sudo yum install rsync  (rsync 是一个远程数据同步工具，可通过 LAN/WAN 快速同步多台主机间的文件)
```

```
$ ssh  sudo /etc/init.d/ssh restart  启动服务
```

命令执行完毕，各台机器之间可以通过密码验证相互登陆。

2、配置 Namenode 无密码登录所有 Datanode

(1) 原理

Namenode 作为客户端，要实现无密码公钥认证，连接到服务端 datanode 上时，需要在 namenode 上生成一个密钥对，包括一个公钥和一个私钥，而后将公钥复制到 datanode 上。当 namenode 通过 ssh 连接 datanode 时，datanode 就会生成一个随机数并用 namenode 的公钥对随机数进行加密，并发送给 namenode。namenode 收到加密数之后再用私钥进行解密，并将解密数回传给 datanode，datanode 确认解密数无误之后就允许 namenode 进行连接了。这就是一个公钥认证过程，其间不需要用户手工输入密码。重要过

程是将客户端 namenode 公钥复制到 datanode 上。

(2) Namenod 机器上生成密码对

在 Namenode 节点上执行以下命令:

```
hadoop@zhangpeng:~$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/hadoop/.ssh/id_rsa): 默认路径
Enter passphrase (empty for no passphrase): 回车, 空密码
Enter same passphrase again:
Your identification has been saved in /home/hadoop/.ssh/id_rsa.
Your public key has been saved in /home/hadoop/.ssh/id_rsa.pub.
```

这将在/home/hadoop/.ssh/目录下生成一个私钥 id_rsa 和一个公钥 id_rsa.pub。

在 namenode 节点上做如下配置

```
hadoop@zhangpeng:~/.ssh$ cp id_rsa.pub authorized_keys (namenode 的公钥)
```

使用 SSH 协议将 namenode 的公钥信息 authorized_keys 复制到所有 DataNode 的.ssh 目录下(.ssh 下最初没有 authorized_keys, 如果有, 则需要追加)。

```
hadoop@zhangpeng:~$ scp authorized_keys hadoop@zhangruixue:/home/hadoop/.ssh
hadoop@zhangpeng:~$ scp authorized_keys hadoop@guanxiansun:/home/hadoop/.ssh
```

这样配置过后, namenode 可以无密码登录所有 datanode, 可以通过命令 “ssh 节点 ip 地址” 来验证。

3、配置 Datanode 无密码登录 Namenode

和 namenode 无密码登录所有 Datanode 原理一样, 把 Datanode 的公钥复制到 Namenode 的.ssh 目录下。

```
hadoop@zhangruixue:~$ scp authorized_keys hadoop@zhangpeng:/home/hadoop/.ssh
hadoop@guanxiansun:~$ scp authorized_keys hadoop@zhangpeng:/home/hadoop/.ssh
```

这样也能在 Datanode 上关闭和启动 Hadoop 服务。

注意: 整个过程中只涉及到创建密钥, 复制公钥, 添加公钥内容, 没有更改配置文件, 实际上须在 Namenode 上编辑配置文件/etc/ssh/sshd_config 开启其公钥验证:

```
RSAAuthentication yes
PubkeyAuthentication yes
```

至此, namenode 能无密码验证登陆每个 datanode。每个 datanode 也能无密码验证 SSH 登陆到 namenode。

三、JDK 安装和 Java 环境变量配置

1、CentOS 下安装 JDK 1.6

所有机器上安装 JDK 1.6, 在所有机器上执行以下命令:

```
$ sudo yum install jdk
```

JDK 默认安装在/usr/java/jdk1.6.0_14 目录里。

2、Java 环境变量配置

在所有机器上执行如下命令：vim /etc/profile。并加入以下内容，配置环境变量

```
# set java environment
export JAVA_HOME="/usr/java/jdk1.6.0_14/"
export JAVA_HOME="/usr/java/jdk1.6.0_14/jre"
export CLASSPATH=.:$JAVA_HOME/lib:$JRE_HOME/lib:$CLASSPATH
export PATH=$JAVA_HOME/bin:$JRE_HOME/bin:$PATH
```

保存并退出，执行以下命令使配置生效：

```
[hadoop@zhangpeng ~]$ source /etc/profile
[hadoop@zhangruixue ~]$ source /etc/profile
[hadoop@guanxiansun ~]$ source /etc/profile
```

配置完毕，在命令行中使用命令“java -version”可以判断是否成功。

四、Hadoop 集群配置

1、下载 *hadoop-0.20.2.tar.gz*

在 **namenode** 上下载 *hadoop-0.20.2.tar.gz*，将其解压到/home/hadoop/hadoopinstall 目录下，使用如下命令：

```
[hadoop@zhangpeng hadoopinstall]$ tar -xvzf hadoop-0.20.2.tar
```

然后在 hadoopinstall 下创建 tmp 文件夹并把 Hadoop 的安装路径添加到/etc/profile 中修改/etc/profile 文件(配置 java 环境变量的文件)，将以下语句添加到末尾，并使其有效(source /etc/profile)：

```
# set hadoop path
export HADOOP_HOME=/home/hadoop/hadoopinstall/hadoop-0.20.1
export PATH=$HADOOP_HOME/bin:$PATH
```

2、编辑 *Hadoop* 的配置文件

(1) 配置 *hadoop-env.sh*

```
[hadoop@zhangpeng conf]$ vim hadoop-env.sh
```

末尾加入以下语句

```
# set java environment
export JAVA_HOME=/usr/java/jdk1.6.0_14/
```

hadoop 配置文件在 conf 目录下，之前的版本 Hadoop 的配置文件主要是 Hadoop-default.xml 和 Hadoop-site.xml。由于 hadoop 发展迅速，代码量急剧增加，代码开发分为了 core,hdfs 和 map/reduce 三部分，配置文件也被分成了三个 core-site.xml、hdfs-site.xml、mapred-site.xml。

(2) 配置 *core-site.xml* 文件

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

```

        <name>hadoop.tmp.dir</name>
        <value>/home/hadoop/hadoopinstall/tmp</value>
        (注意, 请先在 hadoopinstall 目录下建立 tmp 文件夹)
        <description>A base for other temporary directories.</description>
    </property>
<!-- file system properties -->
    <property>
        <name>fs.default.name</name>
        <value>hdfs://192.168.90.91:9000</value>
    </property>
</configuration>

```

(3) 配置 hdfs-site.xml 文件

```

<configuration>
    <property>
        <name>dfs.replication</name>
        <value>1</value>
        (replication 是数据副本数量, 默认为 3, datanode 少于 3 台就会报错)
    </property>
</configuration>

```

(4) 配置 mapred-site.xml 文件

```

<configuration>
    <property>
        <name>mapred.job.tracker</name>
        <value>http://192.168.90.91:9001</value>
    </property>
</configuration>

```

(5) 配置 masters 文件, 加入 namenode 的 ip 地址

```
192.168.90.91
```

(6) 配置 slaves 文件, 加入所有 datanode 的 ip 地址

```
192.168.90.94
192.168.90.205
```

将 namenode 上配置好的 hadoop 所在文件夹 hadoopinstall 复制到 datanode 的 /home/hadoop/ 目录下 (实际上 masters, slaves 文件时不必要的, 复制了也没问题)。

在 datanode 上修改 /etc/profile 文件 (配置 java 环境变量的文件), 将以下语句添加到末尾, 并使其有效 (source /etc/profile):

```

# set hadoop path
export HADOOP_HOME=/home/zk1/hadoopinstall/hadoop-0.20.1
export PATH=$HADOOP_HOME/bin:$PATH

```

五、Hadoop 集群启动

在 Namenode 格式化 hadoop:

```
[hadoop@zhangpeng ~]$ hadoop namenode -format
```

启动 hadoop:

```
[hadoop@zhangpeng ~]$ start-all.sh
```

可以通过以下启动日志看出, 首先启动 namenode 接着启动 datanode1, datanode2, 然后启动 secondarynamenode。再启动 jobtracker, 然后启动 tasktracker1, 最后启动 tasktracker2。

启动 hadoop 成功后, 在 namenode 中的 tmp 文件夹中生成了 dfs 文件夹, 在 datanode 中的 tmp 文件夹中均生成了 dfs 文件夹和 mapred 文件夹。

在 namenode 上用 java 自带的小工具 jps 查看进程

```
[hadoop@zhangpeng ~]$ /usr/java/jdk1.6.0_14/bin/jps
18978 JobTracker
21242 Jps
18899 SecondaryNameNode
18731 NameNode
```

在每个 datanode 上查看进程

```
[hadoop@zhangruixue ~]$ /usr/java/jdk1.6.0_14/bin/jps
17706 TaskTracker
20499 Jps
17592 DataNode
[hadoop@guanxiansun ~]$ /usr/java/jdk1.6.0_14/bin/jps
28550 TaskTracker
28436 DataNode
30798 Jps
```

在 namenode 上查看集群状态:

```
[hadoop@zhangpeng ~]$ hadoop dfsadmin -report
Configured Capacity: 123909840896 (115.4 GB)
Present Capacity: 65765638144 (61.25 GB)
DFS Remaining: 65765257216 (61.25 GB)
DFS Used: 380928 (372 KB)
DFS Used%: 0%
Under replicated blocks: 0
Blocks with corrupt replicas: 0
Missing blocks: 0
```

```
Datanodes available: 2 (2 total, 0 dead)
```

Name: 192.168.90.205:50010
Decommission Status : Normal
Configured Capacity: 61954920448 (57.7 GB)
DFS Used: 258048 (252 KB)
Non DFS Used: 47925145600 (44.63 GB)
DFS Remaining: 14029516800(13.07 GB)
DFS Used%: 0%
DFS Remaining%: 22.64%
Last contact: Thu Jul 22 15:11:28 CST 2010

Name: 192.168.90.94:50010
Decommission Status : Normal
Configured Capacity: 61954920448 (57.7 GB)
DFS Used: 122880 (120 KB)
Non DFS Used: 10219057152 (9.52 GB)
DFS Remaining: 51735740416(48.18 GB)
DFS Used%: 0%
DFS Remaining%: 83.51%
Last contact: Thu Jul 22 15:11:27 CST 2010