

柳杰

搭建多节点的HDFS

---

2017年4月25日

## 一、本周所做任务

### 1. 搭建多节点 HDFS 文件系统

#### 1.1环境说明

环境是在本地虚拟机中配置的,Hadoop集群中包括3个节点：1个Master，2个Salve，节点之间局域网连接，可以相互ping通，节点IP地址分布如下：

虚拟机系统	机器名称	IP地址
Ubuntu16.04	Master.Hadoop	192.168.1.141
Ubuntu16.04	Salve1.Hadoop	192.168.1.142
Ubuntu16.04	Salve1.Hadoop	192.168.1.137

Master机器主要配置NameNode和JobTracker的角色，负责总管分布式数据和分解任务的执行；2个Salve机器配置DataNode 和TaskTracker的角色，负责分布式数据存储以及任务的执行。

#### 1.2 环境配置

Hadoop集群要按照上表格所示进行配置，下面介绍如何修改机器名称和配置hosts文件，以方便使用。

##### (1) 修改当前机器名称

修改文件/etc/hostname里的值，修改成功后用hostname命令查看当前主机名是否设置成功。

另外为了能正确解析主机名，最好也修改/etc/hosts文件里对应的主机名

##### (2) 配置hosts文件

## 第二周报告

"/etc/hosts"这个文件是用来配置主机将用的DNS服务器信息，是记载LAN内接续的各主机的对应[HostName IP]用的。当用户在进行网络连接时，首先查找该文件，寻找对应主机名对应的IP地址。

要测试两台机器之间知否连通，一般用"ping 机器的IP"，如果想用"ping 机器的主机名"发现找不见该名称的机器(这也就是为什么在修改主机名的同时最好修改该文件中对应的主机名)，解决的办法就是修改"/etc/hosts"这个文件，通过把LAN内的各主机的IP地址和HostName的一一对应写入这个文件的时候，就可以解决问题。

在进行Hadoop集群配置中，需要在"/etc/hosts"文件中添加集群中所有机器的IP与主机名，这样Master与所有的Slave机器之间不仅可以通过IP进行通信，而且还可以通过主机名进行通信。所以在所有的机器上的"/etc/hosts"文件中都要添加如下内容：

```
192.168.1.141 Master.Hadoop
```

```
192.168.1.142 Slave1.Hadoop
```

```
192.168.1.137 Slave2.Hadoop
```

对机器为"Slave1.Hadoop"的主机名进行ping通测试，看是否能测试成功。

```
$ping Slave1.Hadoop
```

### 1.3 搭建多机分布式HDFS所需软件版本

#### (1) JDK软件

下载地址：<http://www.oracle.com/technetwork/java/javase/index.html>

JDK版本：jdk-7u25-linux-i586.tar.gz

#### (2) Hadoop软件

下载地址：<http://hadoop.apache.org/common/releases.html>

Hadoop版本：hadoop-1.1.2.tar.gz

## 2、SSH无密码验证配置

### 配置原因

Hadoop运行过程中需要管理远端Hadoop守护进程，在Hadoop启动以后，NameNode是通过SSH（Secure Shell）来启动和停止各个DataNode上的各种守护进程的。这就必须在节点之间执行指令的时候是不需要输入密码的形式，故需要配置SSH运用

## 第二周报告

无密码公钥认证的形式，这样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。

如果Linux没有安装SSH，首先安装SSH

Ubuntu下安装ssh: `sudo apt-get install openssh-server`

### 2.1 SSH基本原理和用法

#### 1) SSH基本原理

SSH之所以能够保证安全，原因在于它采用了公钥加密。过程如下：

(1)远程主机收到用户的登录请求，把自己的公钥发给用户。

(2)用户使用这个公钥，将登录密码加密后，发送回来。

(3)远程主机用自己的私钥，解密登录密码，如果密码正确，就同意用户登录。

#### 2) SSH基本用法

假如用户名为java，登录远程主机名为linux，如下命令即可：

```
$ ssh java@linux
```

SSH的默认端口是22，也就是说，你的登录请求会送进远程主机的22端口。使用p参数，可以修改这个端口，例如修改为88端口，命令如下：

```
$ ssh -p 88 java@linux
```

问题：如果出现错误提示：ssh: Could not resolve hostname linux: Name or service not known，则是因为linux主机未添加进本主机的Name Service中，故不能识别。

解决办法：需要在/etc/hosts里添加进该主机及对应的IP即可：

```
linux    192.168.1.107
```

### 2.2 配置Master无密码登录所有Slave

#### 1) SSH无密码原理

Master (NameNode | JobTracker) 作为客户端，要实现无密码公钥认证，连接到服务器Slave (DataNode | Tasktracker) 上时，需要在Master上生成一个密钥对，包括一个公钥和一个私钥，而后将公钥复制到所有的Slave上。当Master通过SSH连接Slave时，Slave就会生成一个随机数并用Master的公钥对随机数进行加密，并发送给Master。

## 第二周报告

Master收到加密数之后再用私钥解密，并将解密数回传给Slave，Slave确认解密数无误之后就允许Master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。

### 2) Master机器上设置无密码登录

a. Master节点利用ssh-keygen命令生成一个无密码密钥对。

在Master节点上执行以下命令：

```
ssh-keygen -t rsa -P ""
```

运行后询问其保存路径时直接回车采用默认路径。生成的密钥对：id\_rsa（私钥）和id\_rsa.pub（公钥），默认存储在"/home/用户名/.ssh"目录下。

查看"/home/用户名/"下是否有".ssh"文件夹，且".ssh"文件下是否有两个刚生产的无密码密钥对。

b. 接着在Master节点上做如下配置，把id\_rsa.pub追加到授权的key里面去。

```
cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
```

查看下authorized\_keys的权限，如果权限不对则利用如下命令设置该文件的权限：

```
chmod 600 authorized_keys
```

c. 用root用户登录修改SSH配置文件"/etc/ssh/sshd\_config"的下列内容。

检查下面几行前面"#"注释是否取消掉：

```
RSAAuthentication yes # 启用 RSA 认证
```

```
PubkeyAuthentication yes # 启用公钥私钥配对认证方式
```

```
AuthorizedKeysFile %h/.ssh/authorized_keys # 公钥文件路径
```

设置完之后重启SSH服务，才能使刚才设置有效。

问题：有时候在测试时可能会出现错误：Agent admitted failure to sign using the key.

解决办法：ssh-add ~/.ssh/id\_rsa，如下所示：

d. 使用ssh-copy-id命令将公钥传送到远程主机上。

## 第二周报告

e. 测试是否无密码登录其它机器成功。

接下来配置所有Slave无密码登录Master，其和Master无密码登录所有Slave原理一样，就是把Slave的公钥追加到Master的".ssh"文件夹下的"authorized\_keys"中。

## 3、Hadoop集群安装

### 3.1 安装hadoop

首先用root用户登录"Master.Hadoop"机器，将下载的"hadoop-1.1.2.tar.gz"复制到/usr目录下。然后进入"/usr"目录下，用下面命令把"hadoop-1.1.2.tar.gz"进行解压，并将其重命名为"hadoop"，把该文件夹的读权限分配给普通用户hadoop，然后删除"hadoop-1.0.0.tar.gz"安装包。

```
cd /usr
```

```
tar -xzf hadoop-1.1.2.tar.gz
```

```
mv hadoop-1.1.2 hadoop
```

```
chown -R hadoop:hadoop hadoop #将文件夹"hadoop"读权限分配给hadoop普通用户
```

```
rm -rf hadoop-1.1.2.tar.gz
```

最后在"/usr/hadoop"下面创建tmp文件夹，并把Hadoop的安装路径添加到"/etc/profile"中，修改"/etc/profile"文件，将以下语句添加到末尾，并使其生效(. /etc/profile):

```
# set hadoop path
```

```
export HADOOP_HOME=/usr/hadoop
```

```
export PATH=PATH:HADOOP_HOME/bin
```

### 3.2 配置hadoop

(1) 配置hadoop-env.sh

该"hadoop-env.sh"文件位于"/usr/hadoop/conf"目录下。

在文件中修改下面内容：

```
export JAVA_HOME=/usr/java/jdk1.7.0_25
```

## 第二周报告

Hadoop配置文件在conf目录下，之前的版本的配置文件主要是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。core-site.xml和hdfs-site.xml是站在HDFS角度上配置文件；core-site.xml和mapred-site.xml是站在MapReduce角度上配置文件。

### (2) 配置core-site.xml文件

修改Hadoop核心配置文件core-site.xml，这里配置的是HDFS master（即namenode）的地址和端口号。

```
<configuration>

  <property>

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

    <value>/usr/hadoop/tmp</value>

    （备注：需要先在 /usr/hadoop 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://192.168.1.141:9000</value>

  </property>

</configuration>
```

备注：如没有配置hadoop.tmp.dir参数，此时系统默认的临时目录为：/tmp/hadoop-hadoop。而这个目录在每次重启后都会被删掉，必须重新执行format才行，否则会出错。

### (3) 配置hdfs-site.xml文件

修改Hadoop中HDFS的配置，配置的备份方式默认为3。

## 第二周报告

```
<configuration>
```

```
  <property>
```

```
    <name>dfs.replication</name>
```

```
    <value>1</value>
```

(备注: replication 是数据副本数量, 默认为3, slave少于3台就会报错)

```
  </property>
```

```
</configuration>
```

clip\_image034[11]

(4) 配置mapred-site.xml文件

修改Hadoop中MapReduce的配置文件, 配置的是JobTracker的地址和端口。

```
<configuration>
```

```
  <property>
```

```
    <name>mapred.job.tracker</name>
```

```
    <value>http://192.168.1.141:9001</value>
```

```
  </property>
```

```
</configuration>
```

(5) 配置masters文件

有两种方案:

(1) 第一种

修改localhost为Master.Hadoop

(2) 第二种

去掉"localhost", 加入Master机器的IP: 192.168.1.141

为保险起见, 启用第二种, 因为万一忘记配置"/etc/hosts"局域网的DNS失效, 这样就会出现意想不到的错误, 但是一旦IP配对, 网络畅通, 就能通过IP找到相应主机。

## 第二周报告

### (6) 配置slaves文件 (Master主机特有)

有两种方案：

#### (1) 第一种

去掉"localhost"，每行添加一个主机名，把剩余的Slave主机名都填上。

例如：添加形式如下：

Slave1.Hadoop

Slave2.Hadoop

#### (2) 第二种

去掉"localhost"，加入集群中所有Slave机器的IP，也是每行一个。

例如：添加形式如下

192.168.1.142

192.168.1.137

原因和添加"masters"文件一样，选择第二种方式。

现在在Master机器上的Hadoop配置就结束了，剩下的就是配置Slave机器上的Hadoop。

最简单的方法是将 Master上配置好的hadoop所在文件夹"/usr/hadoop"复制到所有的Slave的"/usr"目录下（实际上Slave机器上的slavers文件是不必要的，复制了也没问题）。用下面命令格式进行。（备注：此时用户可以为普通用户也可以为root）

```
scp -r /usr/hadoop root@服务器IP:/usr/
```

例如：从"Master.Hadoop"到"Slave1.Hadoop"复制配置Hadoop的文件。

```
scp -r /usr/hadoop root@Slave1.Hadoop:/usr/
```

以root用户进行复制，当然不管是用户root还是普通用户，虽然Master机器上的"/usr/hadoop"文件夹用户hadoop有权限，但是Slave1上的hadoop用户却没有"/usr"权限，所以没有创建文件夹的权限。所以无论是哪个用户进行拷贝，右面都是"root@机器 IP"格式。因为我们只是建立起了普通用户的SSH无密码连接，所以用root进行"scp"时，仍提示让你输入"Slave1.Hadoop" 服务器用户root的密码。



## 第二周报告

查看"Slave1.Hadoop"服务器的"/usr"目录下是否已经存在"hadoop"文件夹，确认已经复制成功。查看结果如下：

给"Slave1.Hadoop"服务器上的用户hadoop添加对"/usr/hadoop"读权限。

以root用户登录"Slave1.Hadoop"，执行下面命令。

```
chown -R hadoop:hadoop (用户名: 用户组) hadoop (文件夹)
```

接着在"Slave1.Hadoop"上修改"/etc/profile"文件，将以下语句添加到末尾，并使其有效 (source /etc/profile)：

```
# set hadoop environment
```

```
export HADOOP_HOME=/usr/hadoop
```

```
export PATH=PATH:PATH:HADOOP_HOME/bin
```

如果不知道怎么设置，可以查看前面"Master.Hadoop"机器的"/etc/profile"文件的配置，到此为止在一台Slave机器上的Hadoop配置就结束了。剩下的事儿就是照葫芦画瓢把剩余的几台Slave机器进行部署Hadoop。

### 4.3 启动及验证

#### (1) 格式化HDFS文件系统

在"Master.Hadoop"上使用普通用户hadoop进行操作。（备注：只需一次，下次启动不再需要格式化，只需 start-all.sh）

```
hadoop namenode -format
```

从上图中知道我们已经成功格式化了，但是美中不足就是出现了一个警告，从网上得知这个警告并不影响hadoop执行，但是也有办法解决，详情看后面的"常见问题FAQ"。

#### (2) 启动hadoop

在启动前关闭集群中所有机器的防火墙，不然会出现datanode开后又自动关闭。使用下面命令启动。

```
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, task-tracker2, ...。

## 第二周报告

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

### (3) 验证hadoop

#### (1) 验证方法一：用"jps"命令

在Master上用 java自带的小工具jps查看进程。

在Slave2上用jps查看进程。

#### (2) 验证方式二：用"hadoop dfsadmin -report"

用这个命令可以查看Hadoop集群的状态。

## 4.4 网页查看集群

#### (1) 访问"http://192.168.1.141:50030"

#### (2) 访问"http://192.168.1.142:50070"

## 二、下周计划

将多节点的HDFS部署到azure平台上