**目录**

[配置前的准备工作 2](#_Toc22224604)

[第一步：准备三台虚拟机 2](#_Toc22224605)

[（第一步：桥接模式下的三台虚拟机配置） 2](#_Toc22224606)

[第二步：实现三台虚拟机之间的ssh免密登陆 3](#_Toc22224607)

[分布式配置 3](#_Toc22224608)

[第一步：修改配置文件 3](#_Toc22224609)

[第二步：启动Hadoop 3](#_Toc22224610)

[观察与思考 6](#_Toc22224611)

[第一题：Summary信息的含义 6](#_Toc22224612)

[第二题：上传文件后的Summary信息变化 6](#_Toc22224613)

[第三题：关闭一台虚拟机后的Summary信息变化 7](#_Toc22224614)

[第四题：上传若干小文件后的Summary信息变化 7](#_Toc22224615)

[第五题：多台计算机的集群搭建 8](#_Toc22224616)

# 配置前的准备工作

## 第一步：准备三台虚拟机

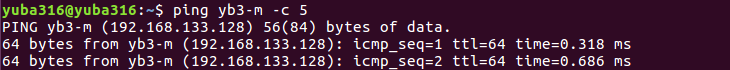
1. 分别配置三台虚拟机，在上面安装好Hadoop-2.9.2和jdk1.8.0\_221
2. 在VMware Workstation上将三者的网络适配器更改为NAT模式，与主机共享IP地址
3. ifconfig 此步可返回每台虚拟机的IP地址，记录下来，方便后续通信
4. vim /etc/hosts 此步可以修改每一台虚拟机的域名，以后需要通过ssh登陆其他虚拟机时就不需要再输入一长串IP地址了

【表1】三台虚拟机的IP地址与域名

|  |  |  |  |
| --- | --- | --- | --- |
| 虚拟机 | 主从关系 | 域名 | IP地址 |
| 1 | master | yb3-m | 192.168.133.128 |
| 2 | slave1 | yb3-s1 | 192.168.133.131 |
| 3 | slave2 | yb3-s2 | 192.168.133.130 |

1. ping yb3-m 此步可以检验三台虚拟机之间是否可以实现通信

【图1】虚拟机间通信成功结果



## （第一步：桥接模式下的三台虚拟机配置）

1. 在VMware Workstation上三台机器的网络适配器更改为桥接模式模式
2. vim /etc/network/interfaces 此步配置网络的四大要素：IP地址、Netmask子网掩码、Getway网关和DNS计算机域名系统，相关信息可以通过输入命令ifconfig查询

【图2】网络配置



1. sudo /etc/init.d/networking restart 此步重启网卡后，三台机器即可互相通信

## 第二步：实现三台虚拟机之间的ssh免密登陆

1. cd ~/.ssh 此步切换至ssh server目录下
2. ssh-keygen -t rsa -P “” 此步生成公钥
3. mv id\_rsa.pub id\_rsa.pub.3 此步对公钥进行重命名
4. scp id\_rsa.pub.3 yuba316@yb3-m:~/.ssh/ 此步将在第三台虚拟机（yb3-s2）上生成的公钥用scp传送到第一台虚拟机（yb3-m）的~/.ssh/文件夹下
5. 在第一、二台虚拟机（yb3-s1）上重复上述操作，直到第一台虚拟机的~/.ssh/文件夹下拥有三台虚拟机的公钥，可以通过输入命令ls检查文件是否传送成功

【图3】三台虚拟机的公钥均已传送成功结果



1. cat id\_rsa.pub.3>>authorized\_keys 重复此步操作可将三台虚拟机的公钥都添加到第一台虚拟机的authorized\_keys文件下
2. scp authorized\_keys yuba316@yb3-s1:~/.ssh/ 重复此步操作可将带有所有虚拟机公钥信息的authorized\_keys文件传送到其他两台虚拟机的~/.ssh/文件夹下实现ssh免密登陆

# 分布式配置

## 第一步：修改配置文件

1. 由于先前已经在虚拟机上搭建过伪分布平台，所以配置文件只需要做稍微的修改即可
2. cd /usr/local/hadoop-2.9.2/etc/hadoop 此步切换目录至配置文件所在文件夹下
3. gedit core-site.xml 此步将core-site.xml文件中的localhost改为yb3-m
4. gedit mapred-site.xml 此步将mapred-site.xml文件中的localhost也同样改为yb3-m

遇到困难：所在目录下不存在mapred-site.xml文件

解决方法：cp mapred-site.xml.template mapred-site.xml 此步将mapred-site.xml.template文件复制为mapred-site.xml文件，再对其进行修改即可

1. gedit hdfs-site.xml 此步将hdfs-site.xml文件中的数据复制份数由1改为3
2. gedit masters 此步在masters文件中输入第一台虚拟机的域名yb3-m
3. gedit slaves 此步在slaves文件中换行输入三台虚拟机的域名yb3-m、yb3-s1、yb3-s2
4. 利用scp将上述修改好的配置文件分别传送到另外两台虚拟机上

## 第二步：启动Hadoop

1. hadoop namenode -format 此步格式化结点
2. start-all.sh 此步启动三台虚拟机上的所有结点

遇到困难：chown: 正在更改’/usr/local/hadoop-2.9.2/logs’ 的所有者: 不允许的操作

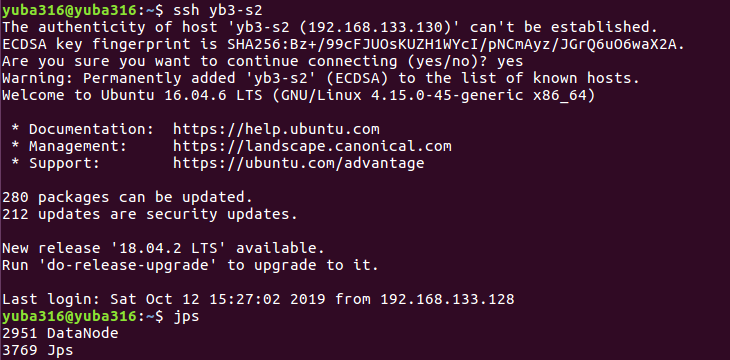
解决方法：sudo chown -R yuba316 /usr/local/hadoop-2.9.2 此步将文件夹下的所有内容的拥有者都更改为普通用户yuba316

（三台机器时遇到困难：head: 无法打开’/usr/local/hadoop-2.9.2/logs/hadoop-hadoop-namenode-yuba316.out’ 读取数据: 没有那个文件或目录

解决方法：此处因为master机器上安装Hadoop的文件夹名与其他两台机器的不同，且三台机器登陆的用户名也不同，需要将文件夹名由hadoop-2.9.2改为hadoop，此时core-site.xml和hdfs-site.xml等配置文件也需要做相应的更改，同时注意通过输入命令vim ~/.bashrc更改环境变量里的Hadoop路径，再输入source ~/.bashrc重新加载环境变量）

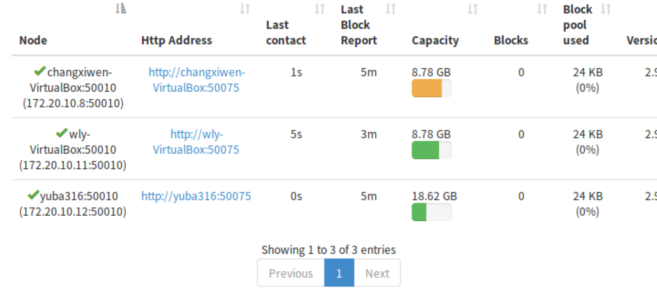
1. ssh yb3-s2 此步通过ssh登陆到第二台虚拟机上输入jps查看结点是否都启动成功

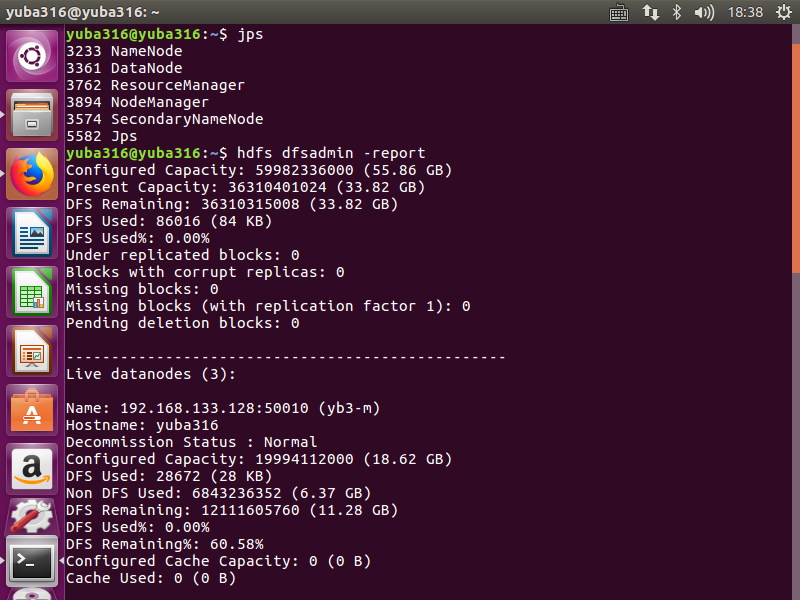
【图4】slave虚拟机结点启动成功结果

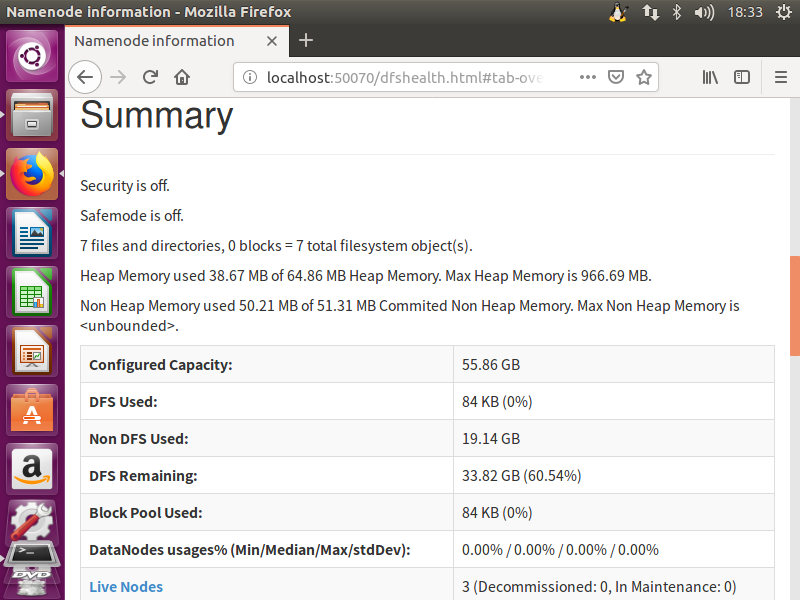


1. mr-jobhistory-daemon.sh start historyserver 此步启动jobhistoryserver进程
2. hdfs dfsadmin -report 此步查看其他datanode的信息
3. 登陆yb3-m:50070可以查看结点信息，可以看到Live Nodes显示的是3个

【图5】所有数据结点启动成功结果







# 观察与思考

## 困难与解决方法

1. 在使用一台机器配置三台虚拟机搭建分布式平台时，即便已确定三台虚拟机的Datanode都已启动，但访问yb3-m:50070网页和用hdfs dfsadmin -report时发现livenode只有一个

**解决方法：**第一次配置时，全部使用了虚拟机的IP地址进行配置，需要通过输入命令vim /etc/hosts为三台虚拟机设置域名后用域名重新配置

1. livenode已经变为三个了，但点击datanode information后却发现只显示一个datanode的信息，怀疑是因为datanode重名导致的，已经修改了三台虚拟机配置文件hdfs-site.xml里存储datanode信息的文件夹位置，问题依旧未得到解决，但可以肯定的是，三个数据结点进程已经成功启动，只是在网页上显示不完全而已

**解决方法：**这是因为三台虚拟机的主机名都一样导致的，对每台虚拟机都输入以下命令：vim /etc/hostname 进行主机名的修改，同时注释掉hosts里原来的主机名地址，重启虚拟机即可

1. 出现Ubuntu网络连接不上的情况

**解决方法：**<https://blog.csdn.net/Azhanzaitianbian123/article/details/88901199>

## 第一题：Summary信息的含义

1. Live Nodes

观察：显示的结果是3 (Decommissioned: 0, In Maintenance: 0)

思考：由于在配置文件的过程中将yb3-m也作为slave虚拟机开启数据结点，故目前活跃的数据结点数理应当有三个；Decommissioned表示退役状态，当数据结点被停用时，该结点就会先转为Decommissioned In Progress状态，待该结点的所有block都复制完成后才会转为Decommissioned状态；In Maintenance表示维护状态，当数据结点被置为维护状态时，该结点会先转为Entering Maintenance状态，待该结点的所有block都被复制到其他位置时才会转为In Maintenance状态

1. DFS Used

观察：显示的结果是84 KB (0%)

思考：目前仅仅只是启动了三个数据结点还未进行任何操作，所以可以知道hdfs文件占用的空间非常的少，而DFS Remaining指的就是dfs可以使用的剩余空间，至于Non DFS Used则指的是原本配置给dfs的空间有多少是被非hdfs文件所占用的

1. DataNodes usages

观察：显示的结果是0.00%

思考：显然因为三个数据结点在启动之后没有进行过任何操作，所以使用空间都为0，而该栏信息还提供数据结点使用空间的最小值、中位数、最大值和标准差等信息

1. Number of Under-Replicated Blocks

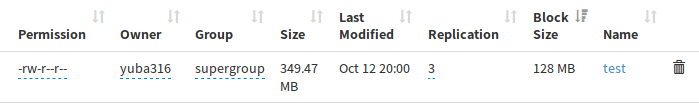
观察：显示的结果是0

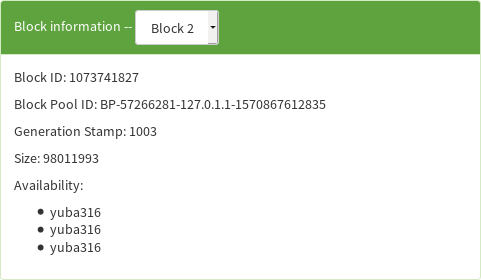
思考：Under-Replicated Block是hdfs中一种记录block优先级的数据结构，此时三个数据结点才刚刚启动还并没有对block进行任何复制操作，即block的副本数量非常少，数据极有可能丢失，所以在Under-Replicated Block里将优先级设为0，即为最高优先级，表示在复制block时应该最先对其进行复制

## 第二题：上传文件后的Summary信息变化

1. hdfs dfs -mkdir -p /user/yuba316/hadoop 此步创建用户文件夹
2. hdfs dfs -put /usr/local/hadoop-2.9.2.tar.gz test 此步将Hadoop压缩包上传到test文件夹
3. DFS Used和Block Pool Used显示的结果都由84 KB (0%)变为1.04 GB (1.87%)，DataNodes usages中的最小值、中位数、最大值和标准差分别为1.85%、1.85%、1.91%和0.03%
4. 点击Utilities中的Browse the file system，进入/user/yuba316文件夹下

【图6】上传一个大文件到hdfs结果





由上图所示可以知道，刚刚上传的Hadoop压缩包被分为了3个block，每个block的大小为128 MB，但是可以看到最后一个block的大小明显比前两个block小一些，具有3个备份，Block Pool ID都是一致的，只有分类标签不同而已

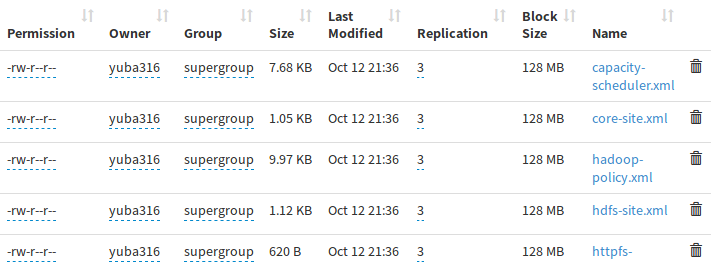
## 第三题：关闭一台虚拟机后的Summary信息变化

1. 事实上除了DataNodes usages的最大值和标准差由1.91%和0.03%变为1.85%和0.00%之外，其他的信息都没有明显的变化，尤其是Live Nodes，结果依旧显示的是3个
2. 仅从DataNodes usages的变化上来看，我们还是可以知道有一个结点已经不再被使用了，通过查看Datanode Information可以发现有一个结点显示状态为Down

## 第四题：上传若干小文件后的Summary信息变化

1. hdfs dfs -put /usr/local/hadoop-2.9.2/etc/hadoop/\*.xml input 此步将Hadoop所有后缀名为xml的配置文件都上传到hdfs上的input文件夹里
2. 点击Utilities中的Browse the file system，进入/user/yuba316/input文件夹下

【图7】上传若干个小文件到hdfs结果



由上图所示可以知道，hdfs为每一个小文件都分配了一个128 MB的block去存储，且每个文件都具有3个备份，区别也只在于分类标签而已

## 第五题：多台计算机的集群搭建

1. 为所有机器都创建用户名一致的用户
2. 安装Java和Hadoop，其中，所有机器的Hadoop安装路径必须一致
3. 确保master机器的用户拥有访问Hadoop所在文件夹下的所有内容的权限
4. 确保master机器与其他slave机器是可以实现相互通信的，方便生成公钥后的传输，以实现ssh的免密登陆
5. 注意修改hdfs-site.xml文件中的备份数目
6. 该解决方案在放到1000台机器的集群搭建时，某些操作，尤其是实现ssh免密登陆前的公钥传输会变得非常繁琐