

实验所使用系统环境为 windows10, VMware Workstation 12 Pro, Ubuntu-16.04, jdk1.8_221 Hadoop_2.6.0.

Vmware 和 ubuntu 下载地址：<https://pan.baidu.com/s/1X29KTBNUx71GcqLc9aGB1Q>
提取码：a9d5

Hadoop 下载地址：链接：https://pan.baidu.com/s/1ug00xUXllvN_zyrXsRVnSw
提取码：64kb

JDK 下载地址：链接：<https://pan.baidu.com/s/11Ex2Q5jNgJ4gEkAfO66hVA>
提取码：qcn5

由于实验条件限制，我们使用 VMware 安装了三个虚拟机 来模拟分布式环境，如果有条件的可以在真机上实验。

1. 虚拟机安装

1.1 打开 VMware Workstation 主界面，选择菜单栏「编辑」 - 「虚拟网络编辑器」，打开虚拟网络配置界面，选择配置 VMnet8，下面选择「NAT 模式」，勾选两个复选框，为了统一后续步骤，请尽量设置子网 IP (192.168.142.0) 和子网掩码 (255.255.255.0)，点击确定完成配置：





1.2 用 VMware 安装三个 Ubuntu 系统，分别起名为 master、slave1、slave2，用户名和密码均为 hadoop。安装完成后我们需要固定三个系统的 ip，以 master 为例说明：

先用 ifconfig 查询 ip，第一个显示的是主网卡，这里是 ens32，有一些应该是 eth0：

```
hadoop@master:~$ ifconfig
ens32    Link encap:Ethernet  HWaddr 00:0c:29:f6:ec:e1
          inet addr:192.168.142.128 Bcast:192.168.142.255 Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fef6:ecel/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500 Metric:1
          RX packets:845528 errors:0 dropped:0 overruns:0 frame:0
          TX packets:300496 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:945409210 (945.4 MB) TX bytes:27512707 (27.5 MB)

lo       Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536 Metric:1
          RX packets:329189 errors:0 dropped:0 overruns:0 frame:0
          TX packets:329189 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:45847438 (45.8 MB) TX bytes:45847438 (45.8 MB)
```

然后编辑 Linux 网卡配置文件（sudo vim /etc/network/interfaces），编辑如下：

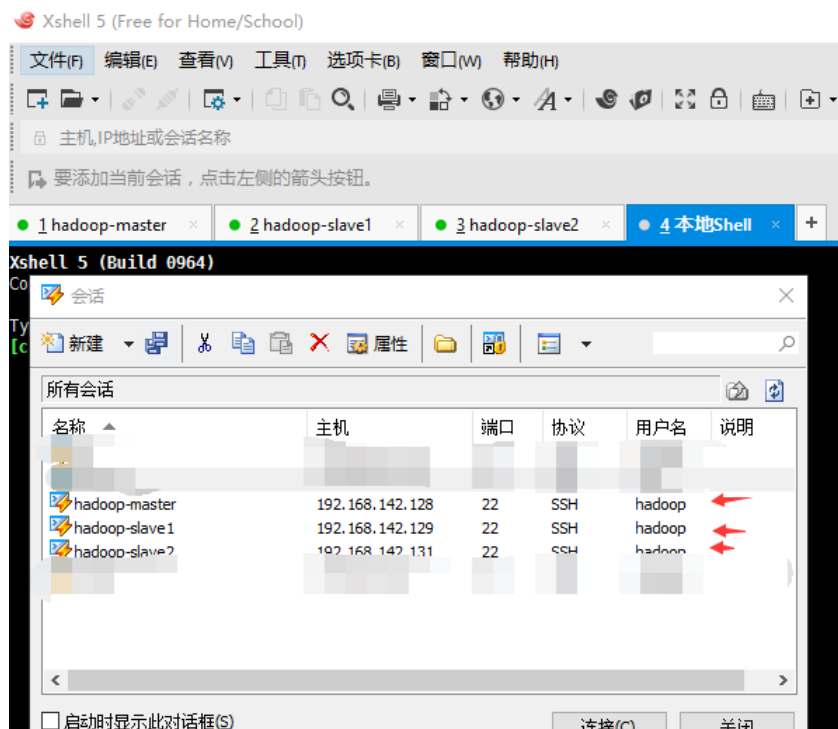
```
interfaces(5) file used by ifup(8) and ifdown(8)
auto lo
iface lo inet loopback

auto ens32
iface ens32 inet static
address 192.168.142.128
gateway 192.168.142.2
netmask 255.255.255.0
```

进一步固话 DNS 服务器 IP：（sudo vim /etc/resolvconf/resolv.conf.d/base）：

```
nameserver 192.168.142.2
```

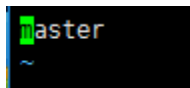
这一步完成后 reboot 重启即可，在 slave1 和 slave2 也进行以上操作，在固定了 IP 之后，我们还可以在虚拟机外通过 xshell 连接它们，这样就避免了键鼠切换的问题，操作起来更方便：



2. 配置免密登陆

2.1 修改各个节点主机器名与 master、slave1、slave2 保持一致：

```
sudo vim /etc/hostname
sudo reboot
```



2.2 修改各个 hosts 文件，在本地植入部分 DNS 映射，将对应的角色名与 IP 匹配起来，然后尝试用角色名相互 ping，相互能 ping 通证明配置成功：

<code>sudo vim /etc/hosts</code>	#编辑 /etc/hosts 文件从而修改主机名
<code>sudo reboot</code>	#重启使新主机名生效

```
127.0.0.1    localhost
127.0.1.1    master

2 192.168.142.128 master
  192.168.142.129 slave1
  192.168.142.131 slave2

# The following lines are desirable for IPv6 capable hosts
::1        ip6-localhost ip6-loopback
fe00::0    ip6-localnet
ff00::0    ip6-mcastprefix
ff02::1    ip6-allnodes
ff02::2    ip6-allrouters
~
~
~
```

2.3 在每一台机器上生成各个节点的 SSH 公私钥：

<code>cd ~/.ssh</code>	# 如果没有该目录，先执行一次 <code>ssh localhost</code>
<code>rm .id_rsa*</code>	# 删除之前生成的公匙（如果有）
<code>ssh-keygen -t rsa</code>	# 一直按回车就可以

```
hadoop@master:~/.ssh$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/hadoop/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
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.
The key fingerprint is:
SHA256:1jNJKNUg+td82sem70JKZ46xyzAv0fdMqjkZiPvtQc4 hadoop@master
The key's randomart image is:
+---[RSA 2048]---+
|  .  oo          |
|  . 0 ..         |
|   . . .         |
|  . . = .        |
| ..So0 .         |
| .00+=*=..       |
| .00E&.=+        |
| . . *0++0       |
| ..00+. +0       |
+---[SHA256]---+
```

2.4 为了免密登录需要每个节点都拥有其他节点的公钥，先在 master 上将 master 的公钥复制到 authorized 里：

<code>cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys</code>	# cat 命令用于提取容，>>输出重定向
--	-----------------------

在 master 上将 slave1 的公钥文件发给 master 再将其追加到 authorized_keys：

<code>scp hadoop@slave1:~/.ssh/id_rsa.pub ~/.ssh</code> <code>cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys</code>

同理也要对 slave2 操作一次：

<code>scp hadoop@slave2:~/.ssh/id_rsa.pub ~/.ssh</code> <code>cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys</code>

然后将 master 的 authorized_keys 发送给 slave1 和 slave2：

<code>scp ~/.ssh/authorized_keys hadoop@slave1:~/.ssh</code> <code>scp ~/.ssh/authorized_keys hadoop@slave2:~/.ssh</code>
--

现在可以在各个节点上通过节点名字免密登陆另外两个节点了。

3. 安装 JDK，此步骤在 3 个节点上都要做一遍

3.1 将上传的 JDK 压缩包（jdk-8u221-linux-x64.tar.gz）放到/home/hadoop/，解压并放到指定的文件夹：

<code>sudo mkdir -p /usr/local/jvm</code> <code>tar -zxvf jdk-8u60-linux-x64.tar.gz -C /usr/local/jvm</code>

3.2：将当前的 PATH 环境变量提取保存到 setenv.sh，然后将其修改为初始化语句，增加 JAVA 的路径：

<code>echo \$PATH >> ~/setenv.sh</code> <code>vi ~/setenv.sh</code>
--

```
export PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin
export JAVA_HOME=/usr/local/jvm/jdk1.8.0_221
export JRE_HOME=${JAVA_HOME}/jre
export CLASSPATH=.:${JAVA_HOME}/lib:${JRE_HOME}/lib
export PATH=$PATH:${JAVA_HOME}/bin
```

3.3 执行 setenv.sh 脚本文件修改当前环境变量 PATH，然后尝试 java 和 javac 指令是否有效：

```
source ~/setenv.sh
java -version
javac -version
```

```
hadoop@master:~$ java -version
java version "1.8.0_221"
Java(TM) SE Runtime Environment (build 1.8.0_221-b11)
Java HotSpot(TM) 64-Bit Server VM (build 25.221-b11, mixed mode)
hadoop@master:~$ javac -version
javac 1.8.0_221
hadoop@master:~$
```

这样只是对当前终端有效，如果想永久有效可以在 ~/.bashrc （对当前用户有效）或者 /etc/profile（对所有用户有效）中修改

4. 安装 hadoop，此步骤在 3 个节点上都要做一遍

4.1 在各个节点上将 hadoop 解压到 /usr/local/ 目录下，改变其所属用户和所属组（让 hadoop 软件用 hadoop 账号登录时对 hadoop 文件夹拥有最高权限）：

```
tar -zxvf hadoop-2.6.0.tar.gz -C /usr/local/
```

```
sudo mv /usr/local/hadoop-2.6.0 /usr/local/hadoop #mv 实现重命名
sudo chown -R hadoop:hadoop /usr/local/hadoop
```

```
hadoop@master:~$ ll /usr/local
total 48
drwxr-xr-x 12 root root 4096 12月 11 15:10 ./
drwxr-xr-x 11 root root 4096 2月 16 2017 ../
drwxr-xr-x 2 root root 4096 2月 16 2017 bin/
drwxr-xr-x 2 root root 4096 2月 16 2017 etc/
drwxr-xr-x 2 root root 4096 2月 16 2017 games/
drwxr-xr-x 12 hadoop hadoop 4096 12月 12 16:58 hadoop/
drwxr-xr-x 2 root root 4096 2月 16 2017 include/
drwxr-xr-x 3 root root 4096 12月 11 14:48 jvm/
drwxr-xr-x 4 root root 4096 2月 16 2017 lib/
lrwxrwxrwx 1 root root    9 12月 11 20:00 man -> share/man/
drwxr-xr-x 2 root root 4096 2月 16 2017 sbin/
drwxr-xr-x 8 root root 4096 2月 16 2017 share/
drwxr-xr-x 2 root root 4096 2月 16 2017 src/
hadoop@master:~$
```

4.2 修改 slaves 文件，让 hadoop 知道自己可以聚合的节点名（保证与 hosts 里的角色名一致）：

```
vim /usr/local/hadoop/etc/hadoop/slaves
```

```
master
slave1
slave2
~
~
```

4.3 修改 core-site.xml 文件如下:

```
vim /usr/local/hadoop/etc/hadoop/core-site.xml
```

```
<configuration>
  <property>
    <name>fs.default.name</name>
    <value>hdfs://master:9000</value>
  </property>
  <property>
    <name>hadoop.tmp.dir</name>
    <value>/usr/local/hadoop/tmp</value>
  </property>
</configuration>
```

4.4 修改 hdfs-site.xml 文件如下 (启用所有节点作为 DataNode, 故 replication=3, 表示有 3 个节点):

```
vim /usr/local/hadoop/etc/hadoop/hdfs-site.xml
```

```
<configuration>

  <property>
    <name>dfs.replication</name>
    <value>3</value>
  </property>
  <property>
    <name>dfs.name.dir</name>
    <value>/usr/local/hadoop/hdfs/name</value>
  </property>
  <property>
    <name>dfs.data.dir</name>
    <value>/usr/local/hadoop/hdfs/data</value>
  </property>

</configuration>
```

4.5 修改 mapred-site.xml 文件如下:

```
vim /usr/local/hadoop/etc/hadoop/mapred-site.xml
```

```
<configuration>

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

</configuration>
```

4.6 修改 yarn-site.xml 文件如下 (启用 yarn 资源管理器):

```
vim /usr/local/hadoop/etc/hadoop/yarn-site.xml
```

```
<configuration>

<!-- Site specific YARN configuration properties -->
  <property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
  </property>

</configuration>
```

4.7 修改 hadoop-env.sh 文件, 将 25 行 JAVA_HOME 的值换成 jdk 所在的路径:

```
vim /usr/local/hadoop/etc/hadoop/hadoop-env.sh
```



```

24 # The java implementation to use.
25 export JAVA_HOME=/usr/local/jvm/jdk1.8.0_221
26
27 # The jsvc implementation to use. Jsvc is required to run secure datanodes
28 # that bind to privileged ports to provide authentication of data transfer
29 # protocol. Jsvc is not required if SASL is configured for authentication of
30 # data transfer protocol using non-privileged ports.
31 #export JSVC_HOME=${JSVC_HOME}

```

说明：上述 Hadoop 的配置操作要在每个节点上做一次，确保每个环节都不出错，我已将改好的文件放在 hadoop setting files 文件夹里，可以直接复制替换

5. 启动 hadoop 并测试，这一步在 master 进行即可

5.1 对 hadoop 进行 NameNode 的格式化：

```
/usr/local/hadoop/bin/hdfs namenode -format
```

```

hadoop@master:~$ hdfs namenode -format
10/12/12 16:57:05 INFO namenode.NameNode: STARTUP_MSG:
/*****
STARTUP_MSG: Starting NameNode
STARTUP_MSG: host = master/127.0.1.1
STARTUP_MSG: args = [-format]
STARTUP_MSG: version = 2.6.0
STARTUP_MSG: classpath = /usr/local/hadoop/etc/hadoop:/usr/local/hadoop/share/hadoop/common/lib/jetty-util-6.1.26.jar:/usr/local/hadoop/share/hadoop/common/lib/servlet-api-2.5.jar:/usr/local/hadoop/share/hadoop/common/lib/jackson-mapper-asl-1.9.13.jar:/usr/local/hadoop/share/hadoop/common/lib/curator-client-2.6.0.jar:/usr/local/hadoop/share/hadoop/common/lib/apached-kerberos-codec-2.0.0-M15.jar:/usr/local/hadoop/share/hadoop/common/lib/protobuf-java-2.5.0.jar:/usr/local/hadoop/share/hadoop/common/lib/hamcrest-core-1.3.jar:/usr/local/hadoop/share/hadoop/common/lib/api-util-1.0.0-M20.jar:/usr/local/hadoop/share/hadoop/common/lib/guava-11.0.2.jar:/usr/local/hadoop/share/hadoop/common/lib/httpclient-4.2.5.jar:/usr/local/hadoop/share/hadoop/common/lib/stax-api-1.0-2.jar:/usr/local/hadoop/share/hadoop/common/lib/asm-3.2.jar:/usr/local/hadoop/share/hadoop/common/lib/jsp-api-2.1.jar:/usr/local/hadoop/share/hadoop/common/lib/commons-beanutils-core-1.8.0.jar:/usr/local/hadoop/share/hadoop/common/lib/commons-configuration-1.6.jar:/usr/local/hadoop/share/hadoop/common/lib/commons-collections-3.2.1.jar:/usr/local/hadoop/share/hadoop/common/lib/commons-compress-1.4.1.jar:/usr/local/hadoop/share/hadoop/common/lib/api-asn1-1.0.0-M20.jar:/usr/local/hadoop/share/hadoop/common/lib/jersey-core-1.9.jar:/usr/local/hadoop/share/hadoop/common/lib/commons-httpclient-3.1.jar:/usr/local/hadoop/share/hadoop/common/lib/jettison-1.1.jar:/usr/local/hadoop/share/hadoop/common/lib/activation-1.1.jar:/usr/local/hadoop/share/hadoop/common/lib/jackson-core-asl-1.9.13.jar:/usr/local/hadoop/share/hadoop/common/lib/jersey-json-1.9.jar:/usr/local/hadoop/share/hadoop/common/lib/jetty-6.1.26.jar:/usr/local/hadoop/share/hadoop/common/lib/bjunit-4.11.jar:/usr/local/hadoop/share/hadoop/common/lib/httpcore-4.2.5.jar:/usr/local/hadoop/share/hadoop/common/lib/curator-recipes-2.6.0.jar:/usr/local/hadoop/share/hadoop/common/lib/log4j-1.2.17.jar:/usr/local/hadoop/share/hadoop/common/lib/commons-beanutils-1.7.0.jar:/usr/local/hadoop/share/hadoop/common/lib/java-xmlbuilder-0.4.jar:/usr/local/hadoop/share/hadoop/common/lib/commons-net-3.1.jar:/usr/local/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.5.jar:/usr/local/hadoop/share/hadoop/common/lib/gson-2.2.4.jar:/usr/local/hadoop/share/hadoop/common/lib/apached-110n-2.0.0-M15.jar:/usr/local/hadoop/share/hadoop/common/lib/jsch-0.1.42.jar:/usr/local/hadoop/share/hadoop/common/lib/commons-io-2.4.jar:/usr/local/hadoop/share/hadoop/common/lib/commons-codec-1.4.jar:/usr/local/hadoop/share/hadoop/common/lib/zokeeper-3.4.6.jar:/usr/local/hadoop/share/hadoop/common/lib/mockito-all-1.8.5.jar:/usr/local/hadoop/share/hadoop/common/lib/jaxrs-1.9.13.jar:/usr/local/hadoop/share/hadoop/common/lib/hadoop-auth-2.6.0.jar:/usr/local/hadoop/share/hadoop/common/lib/commons-el-1.0.jar:/usr/local/hadoop/share/hadoop/common/lib/commons-digester-1.8.jar:/usr/local/hadoop/share/hadoop/common/lib/commons-lang-2.6.jar:/usr/local/hadoop/share/hadoop/common/lib/jersey-server-1.9.jar:/usr/local/hadoop/share/hadoop/common/lib/curator-framework-2.6.0.jar:/usr/local/hadoop/share/hadoop/common/lib/jackson-xc-1.9.13.jar:/usr/local/hadoop/share/hadoop/common/lib/commons-cli-1.2.jar:/usr/local/hadoop/share/hadoop/common/lib/jaxb-impl-2.2.3-1.jar:/usr/local/hadoop/share/hadoop/common/lib/paranamer-2.3.jar:/usr/local/hadoop/share/hadoop/common/lib/jasper-compiler-5.5.23.jar:/usr/local/hadoop/share/hadoop/common/lib/htrace-core-3.0.4.jar:/usr/local/hadoop/share/hadoop/common/lib/jasper-runtime-5.5.23.jar:/usr/local/hadoop/share/hadoop/common/lib/xz-1.0.jar:/usr/local/hadoop/share/hadoop/common/lib/jsr305-1.3.9.jar:/usr/local/hadoop/share/hadoop/common/lib/slf4j-api-1.7.5.jar:/usr/local/hadoop/share/hadoop/common/lib/commons-logging-1.1.3.jar:/usr/local/hadoop/share/hadoop/common/lib/netty-3.6.2.Final.jar:/usr/local/hadoop/share/hadoop/common/hadoop-nfs-2.6.0.jar:/usr/local/hadoop/share/hadoop/common/hadoop-common-2.6.0.jar:/usr/local/hadoop/share/hadoop/common/hadoop-common-2.6.0-tests.jar:/usr/local/hadoop/share/hadoop/hdfs/lib/jetty-util-6.1.26.jar:/usr/local/hadoop/share/hadoop/hdfs/lib/servlet-api-2.5.jar:/usr/local/hadoop/share/hadoop/hdfs/lib/xml-apis-1.3.04.jar:/usr/local/hadoop/share/hadoop/hdfs/lib/jackson-mapper-asl-1.9.13.jar:/usr/local/hadoop/share/hadoop/hdfs/lib/protobuf-java-2.5.0.jar:/usr/local/hadoop/share/hadoop/hdfs/lib/guava-11.0.2.jar:/usr/local/hadoop/share/hadoop/hdfs/lib/asm-3.2.jar:/usr/local/hadoop/share/hadoop/hdfs/lib/jsp-api-2.1.jar:/usr/local/hadoop/share/hadoop/hdfs/lib/xercesImpl-2.9.1.jar:/usr/local/hadoop/share/hadoop/hdfs/lib/commons-daemon-1.0.13.jar:/usr/local/hadoop/share/hadoop/hdfs/lib/jersey-core-1.9.jar:/usr/local/hadoop/share/hadoop/hdfs/lib/jackson-core-asl-1.9.13.jar:/usr/local/hadoop/share/hadoop/hdfs/lib/jetty-6.1.26.jar:/usr/local/hadoop/share/hadoop/hdfs/lib/log4j-1.2.17.jar:/usr/local/hadoop/share/hadoop/hdfs/lib/commons-codec-1.4.jar:/usr/local/hadoop/share/hadoop/hdfs/lib/commons-el-1.0.jar:/usr/local/hadoop/share/hadoop/hdfs/lib/commons-lang-2.6.jar:/usr/local/hadoop/share/hadoop/hdfs/lib/jersey-server-1.9.jar:/usr/local/hadoop/share/hadoop/hdfs/lib/commons-cli-1.2.jar:/usr/local/hadoop/share/hadoop/hdfs/lib/htrace-core-3.0.4.jar:/usr/local/hadoop/share/hadoop/hdfs/lib/jasper-runtime-5.5.23.jar:/usr/local/hadoop/share/hadoop/hdfs/lib/xmlenc-0.52.jar:/usr/local/hadoop/share/hadoop/hdfs/lib/jsr305-1.3.9.jar:/usr/local/hadoop/share/hadoop/hdfs/lib/commons-logging-1.1.3.jar:/usr/local/hadoop/share/hadoop/hdfs/lib/netty-3.6.2.Final.jar:/usr/local/hadoop/share/hadoop/hdfs/hadoop-hdfs-nfs-2.6.0.jar:/usr/local/hadoop/share/hadoop/hdfs/hadoop-hdfs-2.6.0-tests.jar:/usr/local/hadoop/share/hadoop/hdfs/hadoop-hdfs-2.6.0.jar:/usr/local/hadoop/share/hadoop/hdfs/lib/jetty-util-6.1.26.jar:/usr/local/hadoop/share/hadoop/hdfs/lib/servlet-api-2.5.jar:/usr/local/hadoop/share/hadoop/hdfs/lib/jackson-mapper-asl-1.9.13.jar:/usr/local/hadoop/share/hadoop/hdfs/lib/protobuf-java-2.5.0.jar:/usr/local/hadoop/share/hadoop/hdfs/lib/commons-compress-1.4.1.jar:/usr/local/hadoop/share/hadoop/hdfs/lib/jline-0.9.94.jar:/usr/local/hadoop/share/hadoop/hdfs/lib/leveldbjni-all-1.8.jar:/usr/local/hadoop/share/hadoop/hdfs/lib/jersey-core-1.9.jar:/usr/local/hadoop/share/hadoop/hdfs/lib/commons-httpclient-3.1.jar:/usr/local/hadoop/share/hadoop/hdfs/lib/jettison-1.1.jar:/usr/
10/12/12 16:57:07 INFO blockmanagement.BlockManager: shouldCheckForEnoughRacks = false
10/12/12 16:57:07 INFO blockmanagement.BlockManager: replicationRecheckInterval = 3000
10/12/12 16:57:07 INFO blockmanagement.BlockManager: encryptDataTransfer = false
10/12/12 16:57:07 INFO blockmanagement.BlockManager: maxNumBlocksToLog = 1000
10/12/12 16:57:07 INFO namenode.FSNamesystem: fsOwner = hadoop (auth:SIMPLE)
10/12/12 16:57:07 INFO namenode.FSNamesystem: supergroup = supergroup
10/12/12 16:57:07 INFO namenode.FSNamesystem: isPermissionEnabled = true
10/12/12 16:57:07 INFO namenode.FSNamesystem: HA Enabled: false
10/12/12 16:57:07 INFO namenode.FSNamesystem: Append Enabled: true
10/12/12 16:57:12 INFO util.GSet: Computing capacity for map INodeMap
10/12/12 16:57:12 INFO util.GSet: VM type = 64-bit
10/12/12 16:57:12 INFO util.GSet: 1.0% max memory 966.7 MB = 9.7 MB
10/12/12 16:57:12 INFO util.GSet: capacity = 2^20 = 1048576 entries
10/12/12 16:57:12 INFO namenode.NameNode: Caching file names occurring more than 10 times
10/12/12 16:57:12 INFO util.GSet: Computing capacity for map cachedBlocks
10/12/12 16:57:12 INFO util.GSet: VM type = 64-bit
10/12/12 16:57:12 INFO util.GSet: 0.25% max memory 966.7 MB = 2.4 MB
10/12/12 16:57:12 INFO util.GSet: capacity = 2^18 = 262144 entries
10/12/12 16:57:12 INFO namenode.FSNamesystem: dfs.namenode.safemode.threshold-pct = 0.9990000128746033
10/12/12 16:57:12 INFO namenode.FSNamesystem: dfs.namenode.safemode.min.datanodes = 0
10/12/12 16:57:12 INFO namenode.FSNamesystem: dfs.namenode.safemode.extension = 30000
10/12/12 16:57:12 INFO namenode.FSNamesystem: Retry cache on namenode is enabled
10/12/12 16:57:12 INFO namenode.FSNamesystem: Retry cache will use 0.03 of total heap and retry cache entry expiry time is 600000 millis
10/12/12 16:57:12 INFO util.GSet: Computing capacity for map NameNodeRetryCache
10/12/12 16:57:12 INFO util.GSet: VM type = 64-bit
10/12/12 16:57:12 INFO util.GSet: 0.029999999932944774% max memory 966.7 MB = 297.0 KB
10/12/12 16:57:12 INFO util.GSet: capacity = 2^15 = 32768 entries
10/12/12 16:57:12 INFO namenode.NNConf: ACLs enabled? false
10/12/12 16:57:12 INFO namenode.NNConf: XAttrs enabled? true
10/12/12 16:57:12 INFO namenode.NNConf: Maximum size of an xattr: 16384
10/12/12 16:57:13 INFO namenode.FSImage: Allocated new BlockPoolId: BP-1932081557-127.0.1.1-1576141033174
10/12/12 16:57:13 INFO common.Storage: Storage directory /usr/local/hadoop/hdfs/name has been successfully formatted.
10/12/12 16:57:14 INFO util.ExitUtil: Exiting with status 0
10/12/12 16:57:14 INFO namenode.NameNode: SHUTDOWN_MSG:
/*****
SHUTDOWN_MSG: Shutting down NameNode at master/127.0.1.1

```

这一步只要执行一次就可以了，以后重启机器也不用执行了，如果执行两次会造成 5.2 中的 DataNode 服务没有开启，这是只要删掉 hadoop 安装目录下的 hadoop/data 文件夹以及 hadop/name, tmp/data 三个文件夹删掉就可以了。

5.2 启动 hdfs 和 yarn 服务，并在各个节点上输入 jps 查看启动的服务：

```
/usr/local/hadoop/sbin/start-dfs.sh  
/usr/local/hadoop/sbin/start-yarn.sh  
jps
```

```
hadoop@master:~$ jps  
65620 Jps  
65478 ResourceManager  
65335 SecondaryNameNode  
65033 NameNode  
65595 NodeManager  
65149 DataNode
```

```
hadoop@slave1:~$ jps  
62530 Jps  
62435 NodeManager  
62302 DataNode
```

```
hadoop@slave2:~$ jps  
60001 Jps  
59907 NodeManager  
59801 DataNode
```

停止服务命令如下：

```
/usr/local/hadoop/sbin/stop-dfs.sh  
/usr/local/hadoop/sbin/stop-yarn.sh
```

以后开机只要执行启动的两句命令即可。

5.3 尝试在 hdfs 上创建输入文件夹 input，并把 /etc/hadoop 下的所有文本文件放进去：

```
/usr/local/hadoop/bin/hdfs dfs -mkdir /input  
/usr/local/hadoop/bin/hdfs dfs -put /usr/local/hadoop/etc/hadoop/*.xml /input
```

然后我们可在外部浏览器输入 masterIP:50070 查看 hdfs 上的文件：

Namenode information

192.168.142.128:50070/dfshealth.html#tab-overview

应用 百度一下, 你就知道 哔哩哔哩 (゜-゜)つ... GitHub iSEE_Sun Yat-Sen... 主页 - PyTorch中... Conference Mana... 欢迎 - mySYSU 我...

Hadoop Overview Datanodes Snapshot Startup Progress Utilities

Browse the file system
Logs

Overview 'master:9000' (active)

Started:	Fri Dec 13 18:36:53 CST 2019
Version:	2.6.0, re3496499ecb8d220fba99dc5ed4c99c8f9e33bb1
Compiled:	2014-11-13T21:10Z by jenkins from (detached from e349649)
Cluster ID:	CID-cb049473-bb5e-477c-b7d7-54bfa98385e6
Block Pool ID:	BP-295091780-127.0.1.1-1576233352982

Summary

Security is off.
Safemode is off.
11 files and directories, 9 blocks = 20 total filesystem object(s).
Heap Memory used 31.54 MB of 61.66 MB Heap Memory. Max Heap Memory is 966.69 MB.
Non Heap Memory used 46.03 MB of 47.05 MB Committed Non Heap Memory. Max Non Heap Memory is ~1 B.

Configured Capacity:	37.28 GB
DFS Used:	51.06 KB
...	7.54 GB

192.168.142.128:50070/logs

Browsing HDFS

192.168.142.128:50070/explorer.html#/

应用

Hadoop Overview Datanodes Snapshot Startup Progress Utilities

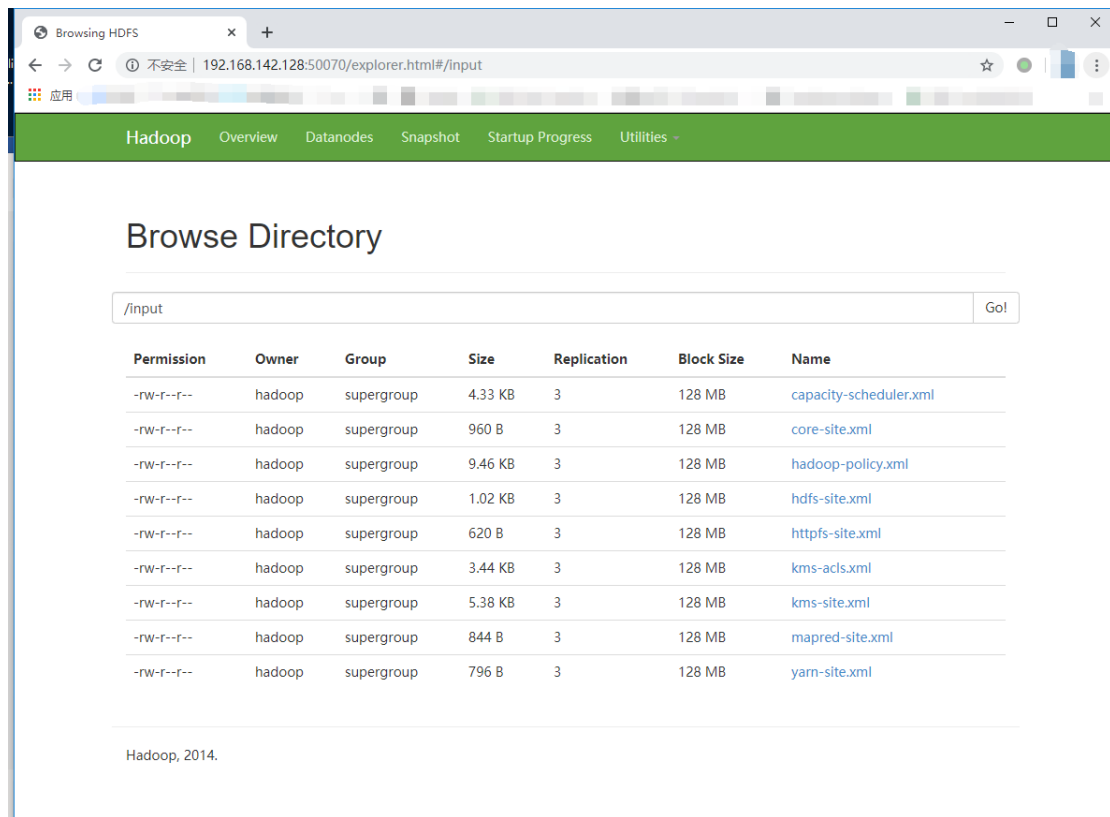
Browse Directory

/

Go!

Permission	Owner	Group	Size	Replication	Block Size	Name
drwxr-xr-x	hadoop	supergroup	0 B	0	0 B	input

Hadoop, 2014.



5.4 尝试用 hadoop 启动自带的 WordCount 样例代码, 统计上面文本文件中每个单词出现的频数：

```
/usr/local/hadoop/bin/hadoop jar
/usr/local/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-
examples-2.6.0.jar wordcount /input /output
/usr/local/hadoop/bin/hdfs dfs -cat /output/*
```

```
hadoop@master:/usr/local/hadoop/share/hadoop/mapreduce$ hadoop jar /usr/local/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.6.0.jar wordcount /input /output
19/12/13 19:01:30 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
19/12/13 19:01:33 INFO input.FileInputFormat: Total input paths to process : 9
19/12/13 19:01:33 INFO mapreduce.JobSubmitter: number of splits:9
19/12/13 19:01:33 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1576233504193_0001
19/12/13 19:01:34 INFO impl.YarnClientImpl: Submitted application application_1576233504193_0001/
19/12/13 19:01:34 INFO mapreduce.Job: The url to track the job: http://master:8088/proxy/application_1576233504193_0001/
19/12/13 19:02:08 INFO mapreduce.Job: Running job: job_1576233504193_0001
19/12/13 19:02:08 INFO mapreduce.Job: Job job_1576233504193_0001 running in uber mode : false
19/12/13 19:04:41 INFO mapreduce.Job: map 0% reduce 0%
19/12/13 19:04:48 INFO mapreduce.Job: map 56% reduce 0%
19/12/13 19:04:48 INFO mapreduce.Job: map 67% reduce 0%
19/12/13 19:05:19 INFO mapreduce.Job: map 100% reduce 0%
19/12/13 19:05:29 INFO mapreduce.Job: map 100% reduce 22%
19/12/13 19:05:22 INFO mapreduce.Job: map 100% reduce 100%
19/12/13 19:05:26 INFO mapreduce.Job: Job job_1576233504193_0001 completed successfully
19/12/13 19:05:26 INFO mapreduce.Job: Counters: 49
  File System Counters
    FILE: Number of bytes read=21756
    FILE: Number of bytes written=1090995
    FILE: Number of read operations=0
    FILE: Number of large read operations=0
    FILE: Number of write operations=0
    HDFS: Number of bytes read=28361
    HDFS: Number of bytes written=10472
    HDFS: Number of read operations=30
    HDFS: Number of large read operations=0
    HDFS: Number of write operations=2
  Job Counters
    Launched map tasks=10
    Launched reduce tasks=1
    Data-local map tasks=10
    Total time spent by all maps in occupied slots (ms)=997027
    Total time spent by all reduces in occupied slots (ms)=27072
    Total time spent by all map tasks (ms)=997027
    Total time spent by all reduce tasks (ms)=27072
    Total vcore-seconds taken by all map tasks=997027
    Total vcore-seconds taken by all reduce tasks=27072
    Total megabyte-seconds taken by all map tasks=1026955648
    Total megabyte-seconds taken by all reduce tasks=27721728
  Map-Reduce Framework
    Map input records=793
    Map output records=2882
    Map output bytes=36632
    Map output materialized bytes=21804
    Input split bytes=942
    Combine input records=2882
    Combine output records=1262
    Reduce input groups=689
    Reduce shuffle bytes=21804
    Reduce input records=1262
```

```

hadoop@master:/usr/local/hadoop/share/hadoop/mapreduce$ hdfs dfs -cat /output/*
"*"      18
"AS"      9
"License");      9
"alice,bob"      18
"kerberos";      1
"simple";      1
'HTTP/' 1
'none' 1
'random'      1
'sasl' 1
'string'      1
'zookeeper'      2
'zookeeper'.      1
(ASF) 1
(Kerberos).      1
(default),      1
(root 1
(specified      1
(the 9
-->      23
0.0      1
1.0.      1
2.0      9
40.      1
<!--      23
</configuration>      9
</description>      42
</property>      70
<?xml      8
<?xml-stYLESHEET      4
<configuration>      9
<description>      41
<description>ACL      21
<description>Default      1
<name>default.key.acl.DECRYPT_EEK</name>      1
<name>default.key.acl.GENERATE_EEK</name>      1
<name>default.key.acl.MANAGEMENT</name>      1
<name>default.key.acl.READ</name>      1
<name>dfs.data.dir</name>      1
<name>dfs.name.dir</name>      1
<name>dfs.replication</name>      1
<name>fs.default.name</name>      1
<name>hadoop.kms.acl.CREATE</name>      1
<name>hadoop.kms.acl.DECRYPT_EEK</name>      1
<name>hadoop.kms.acl.DELETE</name>      1
<name>hadoop.kms.acl.GENERATE_EEK</name>      1
<name>hadoop.kms.acl.GET</name>      1
<name>hadoop.kms.acl.GET_KEYS</name>      1
<name>hadoop.kms.acl.GET_METADATA</name>      1

```

总结，我们先将 hadoop 的 bin 文件夹路径（前面示例中为 `/usr/local/hadoop/bin`）
hadoop 运行 MapReduce 模型的 java 程序需要先将程序打包成 jar 包，然后执行如下命令：

```
hadoop jar xxx.jar [输入文件] [输出文件存放路径]
```

在输出文件存放路径下面找到程序运行结果文件，通过 cat 指令打印文件内容：

```
hdfs dfs -cat 输出文件存放路径/输出文件
```

如果运行 python 程序，则：

```
hadoop jar $HADOOP_HOME/contrib/streaming/hadoop-streaming-2.6.0.jar \
```

```
-mapper 'python mapper.py' -file mapper.py 的路径 \
```

```
-reducer 'python reducer.py' -file reducer.py 的路径 \
```

```
-input 输入文件 -output 输出文件存放路径
```

注意输出文件夹在运行前不能存在的，如果存在可以用 `hdfs dfs -rm -r 输出文件夹` 这个命令删除