

1: 安装 sun jdk

2: 安装 ssh (hadoop 使用 ssh 来实现 cluster 中各 node 的登录认证, 免密码 ssh 设置在后文中有介绍)

```
sudo apt-get install ssh
```

3. 安装 rsync (Ubuntu12.10 已自带 rsync)

```
sudo apt-get install rsync
```

下面开始安装 Hadoop

1、创建 hadoop 用户组以及用户:

```
sudo addgroup hadoop
```

```
sudo adduser --ingroup hadoop hadoop
```

在/home/下会有一个新的 hadoop 文件夹, 此时最好切换至新建的 hadoop 用户登陆 Ubuntu。

2. 将下载的 hadoop 拷贝至该新建文件夹下:

```
cp /mnt/hgfs/hadoop-1.0.4-bin.tar.gz /home/hadoop/
```

3. 进入该目录 (cd /home/hadoop/) 之后, 解压该文件:

```
tar xzf hadoop-1.0.4-bin.tar.gz
```

4. 进入 hadoop-env.sh 所在目录 (hadoop-1.0.4/conf/), 对该文件进行如下内容的修改:

```
export JAVA_HOME=/usr/java/jdk1.6.0_07 (/usr/java/jdk1.6.0_07 为 jdk 安装目录)
```

5. 为了方便执行 Hadoop 命令, 修改/etc/profiles, 在最后面加上

```
export JAVA_HOME=/usr/java/jdk1.6.0_07
```

```
export HADOOP_HOME=/home/hadoop/hadoop-1.0.4
```

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

```
export
```

```
CLASSPATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar:$JAVA_HOME/lib/htmlconverter.
```

```
jar:$JAVA_HOME/lib/jconsole.jar:$JAVA_HOME/lib/sa-jdi.jar
```

重新启动, 使得/etc/profiles 生效。

6. hadoop 默认是 Standalone Operation。可以按照官方文档进行测试:

在/home/hadoop 目录下建立 HadoopStandaloneTest 目录

```
$ mkdir HadoopStandaloneTest
```

在/home/hadoop/HadoopStandaloneTest 目录下执行以下命令:

```
$ mkdir input
```

```
$ cp $HADOOP_HOME/conf/*.xml input
```

```
$ hadoop jar $HADOOP_HOME/hadoop-examples-1.0.4.jar grep input output 'dfs[a-z.]+'
```

(注意 jar 前面不要加-)

(bin/hadoop jar (使用 hadoop 运行 jar 包) hadoop-*_examples.jar (jar 包的名字) grep (要使用的类, 后边的是参数) input output 'dfs[a-z.]+')

整个就是运行 hadoop 示例程序中的 grep, 对应的 hdfs 上的输入目录为 input、输出目录为

output。

)

7. 测试 Pseudo-Distributed Operation

7.1 首先查看 ssh 服务器和 ssh 客户端是否启动

```
$ps -e|grep ssh
```

如看到如下二个进程则 OK

```
hadoop@guxiwu-virtual-machine: ~/HadoopStandaloneTest
2455 ?      00:00:00 unity-applicati
2457 ?      00:00:00 unity-files-dae
2461 ?      00:00:00 unity-gwibber-d
2462 ?      00:00:00 unity-music-dae
2463 ?      00:00:00 unity-lens-phot
2466 ?      00:00:00 unity-shopping-
2467 ?      00:00:00 unity-lens-vide
2536 ?      00:00:00 unity-musicstor
2538 ?      00:00:00 unity-scope-gdo
2575 ?      00:00:00 unity-scope-vid
2599 ?      00:00:00 update-notifier
2609 ?      00:00:00 deja-dup-monito
2618 ?      00:00:00 sh
2619 ?      00:00:04 gnome-terminal
2626 ?      00:00:00 gnome-pty-helpe
2627 pts/0   00:00:00 bash
2737 ?      00:00:00 kworker/0:1
2811 ?      00:00:00 kworker/0:0
2814 pts/0   00:00:00 ps
hadoop@guxiwu-virtual-machine:~/HadoopStandaloneTest$ ps -e|grep *ssh*
hadoop@guxiwu-virtual-machine:~/HadoopStandaloneTest$ ps -e|grep ssh
1999 ?      00:00:00 sshd
2098 ?      00:00:00 ssh-agent
```

7.2 在/home/hadoop 目录下建立 HadoopPseudoDistributTest 目录

```
$ mkdir HadoopPseudoDistributTest
```

```
$ cd HadoopPseudoDistributeTest/
```

```
$ mkdir conf
```

```
$ cp $HADOOP_HOME/conf/* conf （复制 conf 目录下的所有文件）
```

编辑 HadoopPseudoDistributeTest/conf/下的配置文件

core-site.xml: 用于配置 Common 组件的属性

hdfs-site.xml: 用于配置 HDFS 的属性

mapred-site.xml: 用于配置 MapReduce 的属性

masters 指定 master 节点

slaves 指定 slave 节点

core-site.xml:

```
<configuration>
  <property>
    <name>fs.default.name</name>
    <!--最好不要用 localhsot, 否则 Eclipse 插件会出问题 -->
    <value>hdfs://192.168.231.111:9000</value>
  </property>
</property>
```

```
    <!--A base for other temporary directories(用来存储其他临时目录的根目录) -->
    <name>hadoop.tmp.dir</name>
    <value>/home/hadoop/HadoopPseudoDistributeTest/tmpdir</value>
</property>
<property>
    <name>dfs.permissions</name>
    <value>>false</value>
</property>
</configuration>
```

hdfs-site.xml

```
<configuration>
    <property>
        <name>dfs.permissions</name>
        <value>>false</value>
    </property>
    <property>
        <name>dfs.replication</name>
        <value>1</value>
    </property>
    <property>
        <!--指定 namenode 存储文件系统元数据的目录 -->
        <name>dfs.name.dir</name>
        <value>/home/hadoop/HadoopPseudoDistributeTest/tmpdir/hdfs/name</value>
    </property>
    <property>
        <!--指定 datanode 存储数据的目录 -->
        <name>dfs.data.dir</name>
        <value>/home/hadoop/HadoopPseudoDistributeTest/tmpdir/hdfs/data</value>
    </property>
</configuration>
```

mapred-site.xml

```
<configuration>
    <property>
        <name>mapred.job.tracker</name>
        <value>192.168.231.111:9001</value>
    </property>
</configuration>
```

masters

localhost

slaves

localhost

7.3 注意不要在 HadoopPseudoDistributeTest 创建以下目录

tmpdir tmpdir/hdfs/name tmpdir/hdfs/data

7.4 测试 ssh 测试可否使用 ssh 登陆 localhost

\$ ssh localhost

发现需要输入密码

7.5 实现免密码输入 ssh 登录

假设 A 为客户机器，B 为目标机；

要达到的目的：

A 机器 ssh 登录 B 机器无需输入密码；

加密方式选 rsa|dsa 均可以，默认 dsa

做法：

1、登录 A 机器

2、ssh-keygen -t [rsa|dsa]，将会生成密钥文件和私钥文件 id_rsa, id_rsa.pub 或 id_dsa, id_dsa.pub

3、将 .pub 文件复制到 B 机器的 .ssh 目录，并 cat id_dsa.pub >> ~/.ssh/authorized_keys

4、大功告成，从 A 机器登录 B 机器的目标账户，不再需要密码了；

\$ ssh-keygen -t dsa -P "" -f ~/.ssh/id_dsa

其中

-t dsa 指定密码算法为 dsa

-P "" 指不需要 passphrase

-f ~/.ssh/id_dsa 指定秘钥输出文件

```
hadoop@guxiwu-virtual-machine: ~
hadoop@guxiwu-virtual-machine:~$ ls -al .ssh
总用量 12
drwx----- 2 hadoop hadoop 4096 5月 2 19:37 .
drwxr-xr-x 22 hadoop hadoop 4096 5月 2 19:37 ..
-rw-r--r-- 1 hadoop hadoop 222 5月 2 19:37 known_hosts
hadoop@guxiwu-virtual-machine:~$ ssh-keygen -t dsa -P '' -f ~/.ssh/id_dsa
Generating public/private dsa key pair.
Your identification has been saved in /home/hadoop/.ssh/id_dsa.
Your public key has been saved in /home/hadoop/.ssh/id_dsa.pub.
The key fingerprint is:
86:15:7a:8f:41:b8:4a:41:c8:5a:12:c7:75:07:ce:85 hadoop@guxiwu-virtual-machine
The key's randomart image is:
+--[ DSA 1024]-----+
|.+.+o o+=|
|..= .+E= .|
|+ .+.+|
|. .+.+|
|. .S .|
|. .|
+-----+
hadoop@guxiwu-virtual-machine:~$
```

可以看到~/.ssh 目录下多了二个文件

```
hadoop@guxiwu-virtual-machine: ~
Your identification has been saved in /home/hadoop/.ssh/id_dsa.
Your public key has been saved in /home/hadoop/.ssh/id_dsa.pub.
The key fingerprint is:
86:15:7a:8f:41:b8:4a:41:c8:5a:12:c7:75:07:ce:85 hadoop@guxiwu-virtual-machine
The key's randomart image is:
+--[ DSA 1024]-----+
|.+.+o o+=|
|..= .+E= .|
|+ .+.+|
|. .+.+|
|. .S .|
|. .|
+-----+
hadoop@guxiwu-virtual-machine:~$ ls -al .ssh
总用量 20
drwx----- 2 hadoop hadoop 4096 5月 2 20:24 .
drwxr-xr-x 22 hadoop hadoop 4096 5月 2 19:37 ..
-rw----- 1 hadoop hadoop 668 5月 2 20:24 id_dsa
-rw-r--r-- 1 hadoop hadoop 619 5月 2 20:24 id_dsa.pub
-rw-r--r-- 1 hadoop hadoop 222 5月 2 19:37 known_hosts
hadoop@guxiwu-virtual-machine:~$
```

将 ssh 公钥追加到 authorized_keys 后面，即可实现免密钥登陆。

```
$ cat ~/.ssh/id_dsa.pub >> ~/.ssh/authorized_keys
```

修改~/.ssh/authorized_keys 的权限，要保证.ssh 和 authorized_keys 都只有用户自己有写权限。否则验证无效。

```
$ chmod 600 ~/.ssh/authorized_keys
```

~/.ssh 目录的权限为 700，因此不用修改。

再利用 ssh 登录，发现不再需要输入密码

```
$ ssh localhost
```

7.6 为了运行例子，将 `hadoop` 加入到 `sudoers` 中

1) 创建 root 用户密码

```
$ sudo passwd root
```

2) 启用 root 用户登录

点击 `System -> Preferences -> Login Window` 菜单，并切换到 `Security` 选项页，然后选中其下的“`Allow local system administrator login`”选项。

批注 [U1]: 这一步可以省略

3) 进入超级用户模式，也就是输入“`su -`”

```
su -
```

系统会让你输入超级用户密码，输入密码后就进入了超级用户模式，也就是 `root` 用户模式。注意这里有“-”，这和 `su` 是不同的，在用命令“`su`”的时候只是切换到 `root`，但没有把 `root` 的环境变量传过去，还是当前用户的环境变量，用“`su -`”命令将环境变量也一起带过去，就象和 `root` 登录一样。

4) 添加文件的写权限，也就是输入命令：

```
chmod u+w /etc/sudoers
```

5) 编辑 `/etc/sudoers` 文件，也就是输入命令：

```
vi /etc/sudoers
```

进入编辑模式，找到这一行：

```
root ALL=(ALL:ALL) ALL
```

在它的下面添加：

```
guxiwu ALL=(ALL:ALL) ALL
```

```
hadoop ALL=(ALL:ALL) ALL
```

这里的 `hadoop` 是你的用户名，然后保存退出。

6) 撤销文件的写权限，也就是输入命令：

```
chmod u-w /etc/sudoers
```

7.7 格式化 HDFS 的 `namenode`（管理元数据）创建一个空的文件系统

```
$ hadoop --config ~/HadoopPseudoDistributeTest/conf namenode -format
```

注意 `--config` 后面一定用绝对路径指定配置文件所在的路径

7.8 运行 `hadoop`（切记：首先使用 `ssh` 登陆 `localhost`）

```
$ ssh localhost
```

```
$ export HADOOP_CONF_DIR=~/HadoopPseudoDistributeTest/conf (这样后面就不用带--config 选项)
```

```
$ $HADOOP_HOME/bin/start-all.sh
```

7.9 打开浏览器

NameNode - <http://localhost:50070/>

JobTracker - <http://localhost:50030/>

7.10 运行 jps 命令看相应服务是否启动

```
hadoop@guxiwu-virtual-machine: ~/HadoopPseudoDistributeTest
conf tmpdir
hadoop@guxiwu-virtual-machine:~/HadoopPseudoDistributeTest$ ls
conf tmpdir
hadoop@guxiwu-virtual-machine:~/HadoopPseudoDistributeTest$ start-all.sh
Warning: $HADOOP_HOME is deprecated.

starting namenode, logging to /home/hadoop/hadoop-1.0.4/libexec/../logs/hadoop-hadoop-namenode-guxiwu-virtual-machine.out
localhost: starting datanode, logging to /home/hadoop/hadoop-1.0.4/libexec/../logs/hadoop-hadoop-datanode-guxiwu-virtual-machine.out
localhost: starting secondarynamenode, logging to /home/hadoop/hadoop-1.0.4/libexec/../logs/hadoop-hadoop-secondarynamenode-guxiwu-virtual-machine.out
localhost: starting jobtracker, logging to /home/hadoop/hadoop-1.0.4/libexec/../logs/hadoop-hadoop-jobtracker-guxiwu-virtual-machine.out
localhost: starting tasktracker, logging to /home/hadoop/hadoop-1.0.4/libexec/../logs/hadoop-hadoop-tasktracker-guxiwu-virtual-machine.out
hadoop@guxiwu-virtual-machine:~/HadoopPseudoDistributeTest$ jps
8566 DataNode
8363 NameNode
9131 Jps
8857 JobTracker
9066 TaskTracker
8777 SecondaryNameNode
hadoop@guxiwu-virtual-machine:~/HadoopPseudoDistributeTest$
```

7.11 在伪分布式的模式下运行前面的例子

1) Copy the input files into the distributed filesystem:

```
$ cd /home/hadoop/HadoopPseudoDistributeTest
```

```
$ $ hadoop fs -put conf input
```

将本地文件系统目录 conf 拷贝到分布式文件系统的 input 下。

```
$ hadoop fs -ls
```

查看分布式文件系统的内容

```
hadoop@guxiwu-virtual-machine: ~/HadoopPseudoDistributeTest
localhost: starting secondarynamenode, logging to /home/hadoop/hadoop-1.0.4/libexec/../logs/hadoop-hadoop-secondarynamenode-guxiwu-virtual-machine.out
localhost: starting jobtracker, logging to /home/hadoop/hadoop-1.0.4/libexec/../logs/hadoop-hadoop-jobtracker-guxiwu-virtual-machine.out
localhost: starting tasktracker, logging to /home/hadoop/hadoop-1.0.4/libexec/../logs/hadoop-hadoop-tasktracker-guxiwu-virtual-machine.out
hadoop@guxiwu-virtual-machine:~/HadoopPseudoDistributeTest$ jps
8566 DataNode
8363 NameNode
9131 Jps
8857 JobTracker
9066 TaskTracker
8777 SecondaryNameNode
hadoop@guxiwu-virtual-machine:~/HadoopPseudoDistributeTest$ ls
conf tmpdir
hadoop@guxiwu-virtual-machine:~/HadoopPseudoDistributeTest$ hadoop fs -put conf input
Warning: $HADOOP_HOME is deprecated.

hadoop@guxiwu-virtual-machine:~/HadoopPseudoDistributeTest$ hadoop fs -ls
Warning: $HADOOP_HOME is deprecated.

Found 1 items
drwxr-xr-x - hadoop supergroup 0 2013-05-03 23:50 /user/hadoop/input
hadoop@guxiwu-virtual-machine:~/HadoopPseudoDistributeTest$
```

2) \$ cd ~/HadoopPseudoDistributeTest/

3) \$ hadoop jar \$HADOOP_HOME/hadoop-examples-1.0.4.jar grep input output 'dfs[a-z.]+'

4) \$ hadoop fs -ls output

7.12 停止 daemon

```
$ stop-all.sh
```