

Hadoop 部署参考文档

2012-12-03

要想深入的学习 hadoop 数据分析技术，首要的任务是必须要将 hadoop 集群环境搭建起来，可以将 hadoop 简化地想象成一个小软件，通过在各个物理节点上安装这个小软件，然后将其运行起来，就是一个 hadoop 分布式集群了。

原文：<http://blog.csdn.net/ab198604/article/details/8250461>

说来简单，但是应该怎么做呢？不急，本文的主要目的就是让新手看了之后也能够亲自动手实施这些过程。由于本人资金不充裕，只能通过虚拟机来实施模拟集群环境，虽然说是虚拟机模拟，但是在虚拟机上的 hadoop 的集群搭建过程也可以使用在实际的物理节点中，思想是一样的。也如果你有充裕的资金，自己不介意烧钱买诸多电脑设备，这是最好不过的了。

也许有人想知道安装 hadoop 集群需要什么样的电脑配置，这里只针对虚拟机环境，下面介绍下我自己的情况：

CPU: Intel 酷睿双核 2.2Ghz

内存: 4G

硬盘: 320G

系统: xp

老实说，我的本本配置显然不够好，原配只有 2G 内存，但是安装 hadoop 集群时实在是让人崩溃，本人亲身体验过后实在无法容忍，所以后来再扩了 2G，虽然说性能还是不够好，但是学习嘛，目前这种配置还勉强可以满足学习要求，如果你的硬件配置比这要高是最好不过的了，如果能达到 8G，甚至 16G 内存，学习 hadoop 表示无任何压力。

说完电脑的硬件配置，下面说说本人安装 hadoop 的准备条件：

1 安装 VMware WorkStation 软件

有些人会问，为何要安装这个软件，这是一个 VM 公司提供的虚拟机工作平台，后面需要在这个平台上安装 linux 操作系统。具体安装过程网上有很多资料，这里不作过多的说明。

2 在虚拟机上安装 linux 操作系统

在前一步的基础之上安装 linux 操作系统，因为 hadoop 一般是运行在 linux 平台之上的，虽然现在也有 windows 版本，但是在 linux 上实施比较稳定，也不易出错，如果在 windows 安装 hadoop 集群，估计在安装过程中面对的各种问题会让人更加崩溃，其实我还没在 windows 上安装过，呵呵~

在虚拟机上安装的 linux 操作系统为 ubuntu10.04，这是我安装的系统版本，为什么我会使用这个版本呢，很简单，因为我用的熟^_^其实用哪个 linux 系统都是可以的，比如，你可以用 centos, redhat, fedora 等均可，完全没有问题。在虚拟机上安装 linux 的过程也在此略过，如果不了解可以在网上搜搜，有许多这方面的资料。

3 准备 3 个虚拟机节点

其实这一步骤非常简单，如果你已经完成了第 2 步，此时你已经准备好了第一个虚拟节点，那第二个和第三个虚拟机节点如何准备？可能你已经想明白了，你可以按第 2 步的方法，再分别安装两遍 linux 系统，就分别实现了第二、三个虚拟机节点。不过这个过程估计会让你很崩溃，其实还有一个更简单的方法，就是复制和粘贴，没错，就是在你刚安装好的第一个虚拟机节点，将虚拟机文件目录进行复制，形成第二和第三个虚拟机节点。简单吧！~~

很多人也许会问，这三个结点有什么用，原理很简单，按照 hadoop 集群的基本要求，其中一个 master 结点，主要是用于运行 hadoop 程序中的 namenode、secondorynamenode 和 jobtracker 任务。用外两个结点均为 slave 结点，其中一个是为了冗余目的，如果没有冗余，就不能称之为 hadoop 了，所以模拟 hadoop 集群至少要有 3 个结点，如果电脑配置非常高，可以考虑增加一些其它的结点。slave 结点主要将运行 hadoop 程序中的 datanode 和 tasktracker 任务。

所以，在准备好这 3 个结点之后，需要分别将 linux 系统的主机名重命名（因为前面是复制和粘贴操作产生另两上结点，此时这 3 个结点的主机名是一样的），重命名主机名的方法：

Vim /etc/hostname

通过修改 hostname 文件即可，这三个点结均要修改，以示区分。

以下是我对三个结点的 ubuntu 系统主机分别命名为：master, node1, node2

```
zhm@master:~$
```

```
zhm@node1:~$
```

```
zhm@node2:~$
```

基本条件准备好了，后面要干实事了，心急了吧，呵呵，别着急，只要跟着本人的思路，一步一个脚印地，一定能成功布署安装好 hadoop 集群的。安装过程主要有以下几个步骤：

- 一、 *配置 hosts 文件*
- 二、 *建立 hadoop 运行帐号*
- 三、 *配置 ssh 免密码连入*
- 四、 *下载并解压 hadoop 安装包*

- 五、配置 *namenode*, 修改 *site* 文件
- 六、配置 *hadoop-env.sh* 文件
- 七、配置 *masters* 和 *slaves* 文件
- 八、向各节点复制 *hadoop*
- 九、格式化 *namenode*
- 十、启动 *hadoop*
- 十一、用 *jps* 检验各后台进程是否成功启动
- 十二、通过网站查看集群情况

下面我们对以上过程，各个击破吧！~~

一、配置 *hosts* 文件

先简单说明下配置 *hosts* 文件的作用，它主要用于确定每个结点的 IP 地址，方便后续 *master* 结点能快速查到并访问各个结点。在上述 3 个虚拟机结点上均需要配置此文件。由于需要确定每个结点的 IP 地址，所以在配置 *hosts* 文件之前需要先查看当前虚拟机结点的 IP 地址是多少，可以通过 *ifconfig* 命令进行查看，如本实验中，*master* 结点的 IP 地址为：

```
zhm@master:~$ ifconfig
eth1      Link encap:以太网  硬件地址 00:0c:29:3c:ce:38
          inet 地址:192.168.1.100  广播:192.168.1.255  掩码:255.255.255.0
          inet6 地址: fe80::20c:29ff:fe3c:ce38/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  跃点数:1
          接收数据包:33 错误:0 丢弃:0 过载:0 帧数:0
          发送数据包:47 错误:0 丢弃:0 过载:0 载波:0
          碰撞:0 发送队列长度:1000
          接收字节:4285 (4.2 KB) 发送字节:7697 (7.6 KB)
          中断:19 基本地址:0x2024

lo        Link encap:本地环回
          inet 地址:127.0.0.1  掩码:255.0.0.0
```

如果 IP 地址不对，可以通过 *ifconfig* 命令更改结点的物理 IP 地址，示例如下：

```
zhm@master:~$ sudo ifconfig eth1 192.168.1.100
```

通过上面命令可以将 IP 改为 192.168.1.100。将每个结点的 IP 地址设置完成后，就可以配置 *hosts* 文件了，*hosts* 文件路径为 */etc/hosts*，我的 *hosts* 文件配置如下，大家可以参考自己的 IP 地址以及相应的主机名完成配置

```

1 127.0.0.1      localhost.localdomain  localhost
2 #127.0.1.1    zhm-desktop
3
4 192.168.1.102  node2
5 192.168.1.101  node1
6 192.168.1.100  master
7
8 # The following lines are desirable for IPv6 capable hosts
9 ::1           localhost ip6-localhost ip6-loopback
10 fe00::0       ip6-localnet
11 ff00::0       ip6-mcastprefix
12 ff02::1       ip6-allnodes
13 ff02::2       ip6-allrouters
14 ff02::3       ip6-allhosts

```

二、建立 hadoop 运行帐号

即为 hadoop 集群专门设置一个用户组及用户，这部分比较简单，参考示例如下：

```
sudo groupadd hadoop //设置 hadoop 用户组
```

```
sudo useradd -s /bin/bash -d /home/zhm -m zhm -g hadoop -G admin //添加一个 zhm 用户，此用户属于 hadoop 用户组，且具有 admin 权限。
```

```
sudo passwd zhm //设置用户 zhm 登录密码
```

```
su zhm //切换到 zhm 用户中
```

上述 3 个虚拟机结点均需要进行以上步骤来完成 hadoop 运行帐号的建立。

三、配置 ssh 免密码连入

这一环节最为重要，而且也最为关键，因为本人在这一步骤栽了不少跟头，走了不少弯路，如果这一步走成功了，后面环节进行的也会比较顺利。

SSH 主要通过 RSA 算法来产生公钥与私钥，在数据传输过程中对数据进行加密来保障数据

据的安全性和可靠性，公钥部分是公共部分，网络上任一结点均可以访问，私钥主要用于对数据进行加密，以防他人盗取数据。总而言之，这是一种非对称算法，想要破解还是非常有难度的。Hadoop 集群的各个结点之间需要进行数据的访问，被访问的结点对于访问用户结点的可靠性必须进行验证，hadoop 采用的是 ssh 的方法通过密钥验证及数据加解密的方式进行远程安全登录操作，当然，如果 hadoop 对每个结点的访问均需要进行验证，其效率将会大大降低，所以才需要配置 SSH 免密码的方法直接远程连入被访问结点，这样将大大提高访问效率。

OK，废话就不说了，下面看看如何配置 SSH 免密码登录吧！~~

(1) 每个结点分别产生公私密钥。

键入命令：

```

zhm@master:~$ ssh-keygen -t dsa -P '' -f ~/.ssh/id_dsa
Generating public/private dsa key pair.
Your identification has been saved in /home/zhm/.ssh/id_dsa.
Your public key has been saved in /home/zhm/.ssh/id_dsa.pub.
The key fingerprint is:
a9:0e:99:a8:e8:62:91:76:58:5a:b8:a9:49:8d:6b:ea zhm@master
The key's randomart image is:
+---[ DSA 1024]-----+
|
|      .
|    . o      .
|   @         S
|  @ + o .
|+. = + .
|==  o
|@E .
+-----+
zhm@master:~$

```

以上命令是产生公私密钥，产生目录在用户主目录下的.ssh 目录中，如下：

```

zhm@master:~$ cd .ssh/
zhm@master:~/.ssh$ ls
id_dsa  id_dsa.pub
zhm@master:~/.ssh$

```

id_dsa.pub 为公钥，id_dsa 为私钥，紧接着将公钥文件复制成 authorized_keys 文件，这个步骤是必须的，过程如下：

```

zhm@master:~/.ssh$ cat id_dsa.pub >> authorized_keys
zhm@master:~/.ssh$ ls
authorized_keys  id_dsa  id_dsa.pub
zhm@master:~/.ssh$

```

用上述同样的方法在剩下的两个结点中如法炮制即可。

(2) 单机回环 ssh 免密码登录测试

即在单机结点上用 ssh 进行登录，看能否登录成功。登录成功后注销退出，过程如下：


```

zhm@master: ~/.ssh$ ssh localhost
The authenticity of host 'localhost (:::1)' can't be established.
RSA key fingerprint is cb:d0:7b:66:11:a6:5c:a6:65:54:5b:26:ae:43:1c:7e.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'localhost' (RSA) to the list of known hosts.
Linux master 2.6.32-21-generic #32-Ubuntu SMP Fri Apr 16 08:10:02 UTC 2010 i
GNU/Linux
Ubuntu 10.04 LTS

Welcome to Ubuntu!
 * Documentation: https://help.ubuntu.com/

597 packages can be updated.
344 updates are security updates.

New release 'precise' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Sat Dec 1 11:17:04 2012 from master
zhm@master: ~$ exit
注销
Connection to localhost closed.

```

注意标红圈的指示，有以上信息表示操作成功，单点回环 SSH 登录及注销成功，这将为后续跨子结点 SSH 远程免密码登录作好准备。

用上述同样的方法在剩下的两个结点中如法炮制即可。

(3) 让主结点(master)能通过 SSH 免密码登录两个子结点 (slave)

为了实现这个功能，两个 slave 结点的公钥文件中必须要包含主结点的公钥信息，这样当 master 就可以顺利安全地访问这两个 slave 结点了。操作过程如下：

```

zhm@node1: ~/.ssh$ ls
authorized_keys id_dsa id_dsa.pub known_hosts
zhm@node1: ~/.ssh$ scp zhm@master: ~/.ssh/id_dsa.pub ./master_dsa.pub
The authenticity of host 'master (192.168.1.100)' can't be established.
RSA key fingerprint is cb:d0:7b:66:11:a6:5c:a6:65:54:5b:26:ae:43:1c:7e.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'master,192.168.1.100' (RSA) to the list of known ho
ts.
zhm@master's password:
id_dsa.pub                               100% 600    0.6KB/s  00:00
zhm@node1: ~/.ssh$ ls
authorized_keys id_dsa id_dsa.pub known_hosts master_dsa.pub
zhm@node1: ~/.ssh$ cat master_dsa.pub >> authorized_keys
zhm@node1: ~/.ssh$

```

如上过程显示了 node1 结点通过 scp 命令远程登录 master 结点，并复制 master 的公钥文件到当前的目录下，这一过程需要密码验证。接着，将 master 结点的公钥文件追加至 authorized_keys 文件中，通过这步操作，如果不出问题，master 结点就可以通过 ssh 远程免密码连接 node1 结点了。在 master 结点中操作如下：

```

zhm@master: ~/.ssh$ ssh node1
The authenticity of host 'node1 (192.168.1.101)' can't be established.
RSA key fingerprint is cb:d0:7b:66:11:a6:5c:a6:65:54:5b:26:ae:43:1c:7e.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'node1,192.168.1.101' (RSA) to the list of known hosts.
Linux node1 2.6.32-21-generic #32-Ubuntu SMP Fri Apr 16 08:10:02 UTC 2010 i686 GNU/Linux
Ubuntu 10.04 LTS

Welcome to Ubuntu!
 * Documentation:  https://help.ubuntu.com/

594 packages can be updated.
344 updates are security updates.

New release 'precise' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Sun Dec  2 21:24:58 2012 from localhost
zhm@node1: ~$ exit
注销
Connection to node1 closed.
zhm@master: ~/.ssh$ ssh node1

```

由上图可以看出，node1 结点首次连接时需要，“YES”确认连接，这意味着 master 结点连接 node1 结点时需要人工询问，无法自动连接，输入 yes 后成功接入，紧接着注销退出至 master 结点。要实现 ssh 免密码连接至其它结点，还差一步，只需要再执行一遍 ssh node1，如果没有要求你输入“yes”，就算成功了，过程如下：

```

注销
Connection to node1 closed.
zhm@master: ~/.ssh$ ssh node1
Linux node1 2.6.32-21-generic #32-Ubuntu SMP Fri Apr 16 08:10:02 UTC 2010 i686 GNU/Linux
Ubuntu 10.04 LTS

Welcome to Ubuntu!
 * Documentation:  https://help.ubuntu.com/

594 packages can be updated.
344 updates are security updates.

New release 'precise' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Sun Dec  2 21:36:34 2012 from master
zhm@node1: ~$ exit
注销
Connection to node1 closed.
zhm@master: ~/.ssh$

```

如上图所示，master 已经可以通过 ssh 免密码登录至 node1 结点了。

对 node2 结点也可以用同样的方法进行，如下图：

Node2 结点复制 master 结点中的公钥文件

```
zhm@node2: ~/.ssh$ scp zhm@master: ~/.ssh/id_dsa.pub ./master_dsa.pub
The authenticity of host 'master (192.168.1.100)' can't be established.
RSA key fingerprint is cb:d0:7b:66:11:a6:5c:a6:65:54:5b:26:ae:43:1c:7e.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'master,192.168.1.100' (RSA) to the list of known hosts.
zhm@master's password:
id_dsa.pub                                100% 600      0.6KB/s   00:00
zhm@node2: ~/.ssh$ ls
authorized_keys id_dsa id_dsa.pub known_hosts master_dsa.pub
zhm@node2: ~/.ssh$ cat master_dsa.pub >> authorized_keys
zhm@node2: ~/.ssh$
```

Master 通过 ssh 免密码登录至 node2 结点测试：

第一次登录时：

```
zhm@master: ~/.ssh$ ssh node2
The authenticity of host 'node2 (192.168.1.102)' can't be established.
RSA key fingerprint is cb:d0:7b:66:11:a6:5c:a6:65:54:5b:26:ae:43:1c:7e.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'node2,192.168.1.102' (RSA) to the list of known hosts.
Linux node2 2.6.32-21-generic #32-Ubuntu SMP Fri Apr 16 08:10:02 UTC 2010 i686 GNU/Linux
Ubuntu 10.04 LTS

Welcome to Ubuntu!
 * Documentation: https://help.ubuntu.com/

597 packages can be updated.
344 updates are security updates.

New release 'precise' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Sun Dec  2 21:25:50 2012 from localhost
zhm@node2: ~$ exit
注销
Connection to node2 closed.
```

第二次登录时：


```

zhm@master:~/.ssh$ ssh node2
Linux node2 2.6.32-21-generic #32-Ubuntu SMP Fri Apr 16 08:10:02 UTC 2010 i6
NU/Linux
Ubuntu 10.04 LTS

Welcome to Ubuntu!
 * Documentation:  https://help.ubuntu.com/

597 packages can be updated.
344 updates are security updates.

New release 'precise' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Sun Dec  2 21:45:53 2012 from master
zhm@node2:~$ exit
注销
Connection to node2 closed.
zhm@master:~/.ssh$

```

表面上看，这两个结点的 ssh 免密码登录已经配置成功，但是我们还需要对主结点 master 也要进行上面的同样工作，这一步有点让人困惑，但是这是有原因的，具体原因现在也说不太好，据说是真实物理结点时需要做这项工作，因为 jobtracker 有可能会分布在其它结点上，jobtracker 有不存在 master 结点上的可能性。

对 master 自身进行 ssh 免密码登录测试工作：

```

zhm@master:~/.ssh$ scp zhm@master:~/.ssh/id_dsa.pub ./master_dsa.pub
The authenticity of host 'master (192.168.1.100)' can't be established.
RSA key fingerprint is cb:d0:7b:66:11:a6:5c:a6:65:54:5b:26:ae:43:1c:7e.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'master,192.168.1.100' (RSA) to the list of known hosts.
id_dsa.pub                               100% 600    0.6KB/s
zhm@master:~/.ssh$ cat master_dsa.pub >> authorized_keys
zhm@master:~/.ssh$

```

```
zhm@master: ~/.ssh$ ssh master
Linux master 2.6.32-21-generic #32-Ubuntu SMP Fri Apr 16 08:10:02 UTC 201
Ubuntu 10.04 LTS

Welcome to Ubuntu!
 * Documentation:  https://help.ubuntu.com/

597 packages can be updated.
344 updates are security updates.

New release 'precise' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Sun Dec  2 21:18:16 2012 from localhost
zhm@master: ~$ exit
注销
Connection to master closed.
zhm@master: ~/.ssh$
```

至此，SSH 免密码登录已经配置成功。

四、下载并解压 hadoop 安装包

关于安装包的下载就不多说了，不过可以提一下目前我使用的版本为 `hadoop-0.20.2`，这个版本不是最新的，不过学习嘛，先入门，后面等熟练了再用其它版本也不急。而且《hadoop 权威指南》这本书也是针对这个版本介绍的。

注：解压后 hadoop 软件目录在 `/home/zhm/hadoop` 下

五、配置 namenode,修改 site 文件

在配置 site 文件之前需要作一些准备工作，下载 java 最新版的 JDK 软件，可以从 oracle 官网上下载，我使用的 jdk 软件版本为：`jdk1.7.0_09`，我将 java 的 JDK 解压安装在 `/opt/jdk1.7.0_09` 目录中，接着配置 `JAVA_HOME` 宏变量及 hadoop 路径，这是为了方便后面操作，这部分配置过程主要通过修改 `/etc/profile` 文件来完成，在 `profile` 文件中添加如下几行代码：

```
30 export JAVA_HOME=/opt/jdk1.7.0_09
31 export HADOOP_INSTALL=/home/zhm/hadoop
32 export PATH=$PATH:$HADOOP_INSTALL/bin
```

```
zhm@master: ~/hadoop$ source /etc/profile
zhm@master: ~/hadoop$ echo $JAVA_HOME
/opt/jdk1.7.0_09
zhm@master: ~/hadoop$
```

然后执行：

让配置文件立刻生效。上面配置过程每个结点都要进行一遍。

到目前为止，准备工作已经完成，下面开始修改 hadoop 的配置文件了，即各种 site 文件，文件存放在/hadoop/conf 下，主要配置 core-site.xml、hdfs-site.xml、mapred-site.xml 这三个文件。

Core-site.xml 配置如下：

```
1 <?xml version="1.0"?>
2 <?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
3
4 <!-- Put site-specific property overrides in this file. -->
5
6 <configuration>
7   <property>
8     <name>fs.default.name</name>
9     <value>hdfs://master:9000</value>
10    <final>true</final>
11  </property>
12  <property>
13    <name>hadoop.tmp.dir</name>
14    <value>/home/zhm/hadoop/tmp</value>
15    <description>A base for other temporary directories</description>
16  </property>
17 </configuration>
```

Hdfs-site.xml 配置如下：

```
1 <?xml version="1.0"?>
2 <?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
3
4 <!-- Put site-specific property overrides in this file. -->
5
6 <configuration>
7   <property>
8     <name>dfs.name.dir</name>
9     <value>/home/zhm/hadoop/name</value>
10    <final>true</final>
11  </property>
12
13  <property>
14    <name>dfs.data.dir</name>
15    <value>/home/zhm/hadoop/data</value>
16    <final>true</final>
17  </property>
18  <property>
19    <name>dfs.replication</name>
20    <value>2</value>
21    <final>true</final>
22  </property>
23 </configuration>
```

接着是 mapred-site.xml 文件：

```

1 <?xml version="1.0"?>
2 <?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
3
4 <!-- Put site-specific property overrides in this file. -->
5
6 <configuration>
7   <property>
8     <name>mapred.job.tracker</name>
9     <value>192.168.1.100:9001</value>
10  </property>
11 </configuration>

```

六、配置 `hadoop-env.sh` 文件

这个需要根据实际情况来配置。

```

8 # The java implementation to use. Required.
9 export JAVA_HOME=/opt/jdk1.7.0_09
10
11 # Extra Java CLASSPATH elements. Optional.
12 #export HADOOP_CLASSPATH=/home/zhm/hadoop_dir/hadoop-0.20.2

```

七、配置 `masters` 和 `slaves` 文件

根据实际情况配置 `masters` 的主机名，在本实验中，`masters` 主结点的主机名为 `master`，于是在 `masters` 文件中填入：

```
1 master
```

同理，在 `slaves` 文件中填入：

```
1 node1
2 node2
```

八、向各节点复制 `hadoop`

向 `node1` 节点复制 `hadoop`：

```
zhm@master:~$ scp -r ./hadoop node1:~
```

向 `node2` 节点复制 `hadoop`：

```
zhm@master:~$ scp -r ./hadoop node2:~
```

这样，结点 `node1` 和结点 `node2` 也安装了配置好的 `hadoop` 软件了。

九、格式化 `namenode`

这一步在主结点 `master` 上进行操作：


```

zhm@master:~$ hadoop namenode -format
12/12/02 22:44:08 INFO namenode.NameNode: STARTUP_MSG:
/*****
STARTUP_MSG: Starting NameNode
STARTUP_MSG:   host = master/192.168.1.100
STARTUP_MSG:   args = [-format]
STARTUP_MSG:   version = 0.20.2
STARTUP_MSG:   build = https://svn.apache.org/repos/asf/hadoop/common/branches/branch-0.20 -r 911707; compiled by 'chrisdo' on Fri Feb 19 08:07:34 UTC 2010
*****/
Re-format filesystem in /home/zhm/hadoop/name ? (Y or N) Y
12/12/02 22:44:13 INFO namenode.FSNamesystem: fsOwner=zhm,zhm,adm,dialout,cdrom,plugdev,lpadmin,admin,sambashare
12/12/02 22:44:13 INFO namenode.FSNamesystem: supergroup=supergroup
12/12/02 22:44:13 INFO namenode.FSNamesystem: isPermissionEnabled=true
12/12/02 22:44:13 INFO common.Storage: Image file of size 93 saved in 0 seconds.
12/12/02 22:44:13 INFO common.Storage: Storage directory /home/zhm/hadoop/name has been successfully formatted.
12/12/02 22:44:13 INFO namenode.NameNode: SHUTDOWN_MSG:
/*****
SHUTDOWN_MSG: Shutting down NameNode at master/192.168.1.100
*****/

```

注意：上面只要出现“successfully formatted”就表示成功了。

十、启动 hadoop

这一步也在主结点 master 上进行操作：

```

zhm@master:~$ start-all.sh
starting namenode, logging to /home/zhm/hadoop/bin/../logs/hadoop-zhm-namenode-master.out
node1: starting datanode, logging to /home/zhm/hadoop/bin/../logs/hadoop-zhm-datanode-node1.out
node2: starting datanode, logging to /home/zhm/hadoop/bin/../logs/hadoop-zhm-datanode-node2.out
master: starting secondarynamenode, logging to /home/zhm/hadoop/bin/../logs/hadoop-zhm-secondarynamenode-master.out
starting jobtracker, logging to /home/zhm/hadoop/bin/../logs/hadoop-zhm-jobtracker-master.out
node1: starting tasktracker, logging to /home/zhm/hadoop/bin/../logs/hadoop-zhm-tasktracker-node1.out
node2: starting tasktracker, logging to /home/zhm/hadoop/bin/../logs/hadoop-zhm-tasktracker-node2.out
zhm@master:~$

```

十一、用 jps 检验各后台进程是否成功启动

在主结点 master 上查看 namenode,jobtracker,secondarynamenode 进程是否启动。

```
zhm@master:~$ jps
3732 Jps
3377 NameNode
3636 JobTracker
3560 SecondaryNameNode
zhm@master:~$
```

如果出现以上进程则表示正确。

在 node1 和 node2 结点了查看 tasktracker 和 datanode 进程是否启动。

先来 node1 的情况:

```
zhm@node1:~/hadoop/data$ jps
2898 DataNode
3099 Jps
3008 TaskTracker
zhm@node1:~/hadoop/data$
```

下面是 node2 的情况:

```
zhm@node2:~/hadoop/tmp$ jps
2891 DataNode
3091 Jps
2999 TaskTracker
```

进程都启动成功了。恭喜~~~

十二、 通过网站查看集群情况

在浏览器中输入: <http://192.168.1.100:50030>, 网址为 master 结点所对应的 IP:



master Hadoop Map/Reduce Administrative Interface

State: RUNNING

Started: Sun Dec 02 22:46:08 CST 2012

Version: 0.20.2, r911707

Compiled: Fri Feb 19 08:07:34 UTC 2010 by chrisdo

Identifier: 201212022246

Cluster Summary (Heap Size is 7.56 MB/966.69 MB)

Maps	Reduces	Total Submissions	Nodes	Map Task Capacity	Reduce Task Capacity
0	0	0	2	4	4

Scheduling Information

Queue Name	Scheduling Information
default	N/A

Filter (Jobid, Priority, User, Name)

Example: 'user:smith 3200' will filter by 'smith' only in the user field and '3200' in all fields

Running Jobs

none

Completed Jobs

none

Failed Jobs

none

在浏览器中输入: <http://192.168.1.100:50070>, 网址为 master 结点所对应的 IP:

NameNode 'master:9000'

Started: Sun Dec 02 22:45:41 CST 2012
Version: 0.20.2, r911707
Compiled: Fri Feb 19 08:07:34 UTC 2010 by chrisdo
Upgrades: There are no upgrades in progress.

[Browse the filesystem](#)

[Namenode Logs](#)

Cluster Summary

8 files and directories, 1 blocks = 9 total. Heap Size is 9.01 MB / 966.69 MB (0%)

Configured Capacity	:	37.65 GB
DFS Used	:	72 KB
Non DFS Used	:	25.59 GB
DFS Remaining	:	12.05 GB
DFS Used%	:	0 %
DFS Remaining%	:	32.01 %
Live Nodes	:	2
Dead Nodes	:	0

至此，hadoop 的完全分布式集群安装已经全部完成