

生产环境 Hadoop 大集群完全分布式模式安装

介绍

这是利用 Vmware 9.0 在一台服务器上搭建的分布式环境，操作系统 CentOS 6.4 X64 中配置 Hadoop-1.0.4 时的总结文档。 Hadoop 配置建议所有配置文件中使用主机名进行配置，并且机器上应在防火墙中开启相应端口，并设置 SSHD 服务为开机启动，此外 java 环境变量可以在/etc/profile 中配置。

0 安装程序

软件	版本	地址
VMware-workstation-full-9.0.0-812388	9.0.0-812388	http://www.ihacksoft.com/vmware-workstation-900.html SN:JV695-DR0D3-LZUC0-H8852-CAWJ4
Centos	6.4 X64	http://mirrors.163.com/centos/6.4/isos/x86_64/CentOS-6.4-x86_64-bin-DVD1.iso
jdk	jdk-6u45-linux-x64	http://www.oracle.com/technetwork/java/javase/downloads/jdk6download-1902814.html
hadoop	1.0.4	http://apache.fayea.com/apache-mirror/hadoop/common/stable/hadoop-1.0.4.tar.gz

1 集群网络环境介绍及快速部署

1.1 集群环境

主机名	IP	备注
NFSServer	192.168.1.219	独立服务器，提供 NFS 服务
DNSServer	192.168.1.220	独立服务器，提供 DNS 服务
NameNode	192.168.1.221	NameNode
DataNode1	192.168.1.222	DataNode
DataNode2	192.168.1.223	DataNode
DataNode3	192.168.1.224	DataNode
DataNode4	192.168.1.225	DataNode

集群包含七个节点：1 个 NFSServer，1 个 DNSServer, 1 个 namenode，4 个 datanode，节点之间局域网连接，可以相互 ping 通。

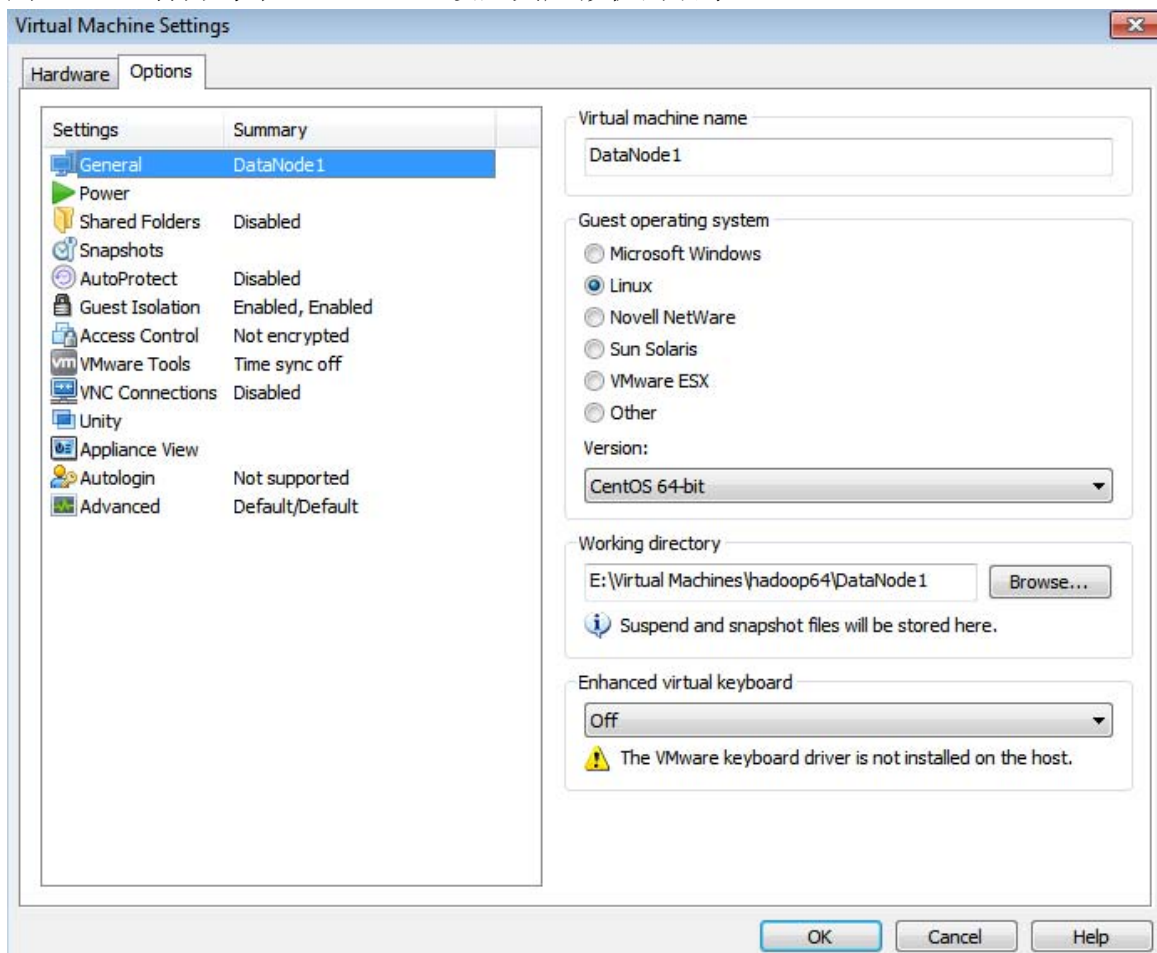
1.2 快速部署节点

所有节点均是 Centos 6.4 64 位系统，**防火墙均禁用，sshd 服务均开启并设置为开机启动。**

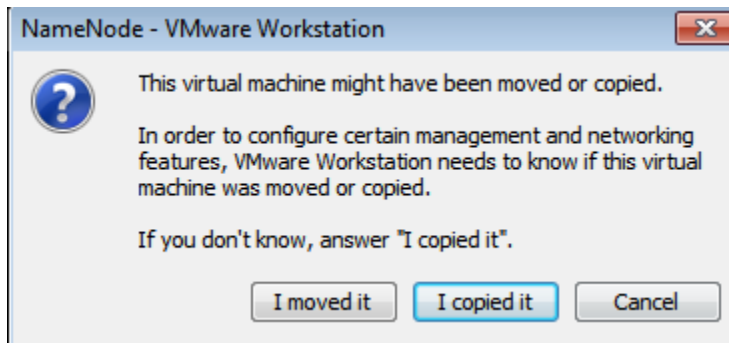
- 首先在 VMware 中安装好一台 Centos 6.4, 创建 hadoop 用户。假设虚拟机的名字为 NameNode
- 关闭虚拟机，把 NameNode 文件夹，拷贝 6 份，并命名为 DataNode1,...,DataNode4,NFSServer,DNSServer

DataNode1	2013/3/10 17:48	File folder
DataNode2	2013/3/10 17:42	File folder
DataNode3	2013/3/10 17:42	File folder
DataNode4	2013/3/10 17:42	File folder
NameNode	2013/3/10 17:48	File folder

- 用 VMware 打开每个 DataNode，设置其虚拟机的名字



- 打开操作系统，当弹出对话框时，选择 “I copy it”



e) 打开每个虚拟机，查看 ip 地址
ifconfig

f) 每个虚拟机，永久关闭防火墙(非常重要，一定要确认)，并关闭 SELINUX

chkconfig iptables off (永久生效)

service iptables stop (临时有效)

vim /etc/selinux/config

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#   enforcing - SELinux security policy is enforced.
#   permissive - SELinux prints warnings instead of enforcing.
#   disabled - No SELinux policy is loaded.
SELINUX=disabled
# SELINUXTYPE= can take one of these two values:
#   targeted - Targeted processes are protected,
#   mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

```
[root@DataNode1 local]# chkconfig iptables off
[root@DataNode1 local]# service iptables stop
iptables: Flushing firewall rules:          [ OK ]
iptables: Setting chains to policy ACCEPT: filter [ OK ]
iptables: Unloading modules:                [ OK ]
[root@DataNode1 local]#
```

g) 配置 NameNode

第一步，检查机器名

```
#hostname
[ root@NameNode ~ ]#
[ root@NameNode ~ ]# hostname
NameNode
[ root@NameNode ~ ]#
```

如发现不对，则修改，root 用户登陆，修改命令如下

vim /etc/sysconfig/network

```
NETWORKING=yes
HOSTNAME=NameNode
```

依次对每个节点进行处理，修改完之后，重启系统 **#reboot**

h) 规划系统目录

安装目录和数据目录分开，且数据目录和 hadoop 的用户目录分开，如果需要重新格式化，则可以直接删除所有的数据目录，然后重建数据目录。

如果数据目录和安装目录或者用户目录放置在一起，则对数据目录操作时，存在误删除程序或者用户文件的风险。

完整路径	说明
/usr/local	hadoop 的程序安装主目录
/home/hadoop_space/tmp	临时目录
/home/hadoop_space/hdfs/name	namenode 上存储 hdfs 名字空间元数据
/home/hadoop_space/hdfs/data	datanode 上数据块的物理存储位置
/home/hadoop_space/mapred/local	tasktracker 上执行 mapreduce 程序时的本地目录
/home/hadoop_space/mapred/system	这个是 hdfs 中的目录，存储执行 mr 程序时的共享文件

至于这里为什么在 /usr/local 下建立安装目录，解释如下

/usr 文件系统

/usr 文件系统经常很大，因为所有程序安装在这里。/usr 里的所有文件一般来自 Linux distribution；本地安装的程序和其他东西在 /usr/local 下。这样可能在升级新版系统或新 distribution 时无须重新安装全部程序。

/usr/local 本地安装的软件和其他文件放在这里。

小贴士：创建目录：mkdir(make directories)

功能说明：建立目录

语 法：mkdir [-p][--help][--version][-m <目录属性>][目录名称]

补充说明：mkdir 可建立目录并同时设置目录的权限。

参 数：

-m<目录属性>或-mode<目录属性> 建立目录时同时设置目录的权限。

-p 或-parents 若所要建立目录的上层目录目前尚未建立，则会一并建立上层目录。

例：mkdir test

开始建立目录：

在 NameNode 下，root 用户

```
mkdir -p /home/hadoop_space/tmp
mkdir -p /home/hadoop_space/hdfs/name
mkdir -p /home/hadoop_space/hdfs/data
mkdir -p /home/hadoop_space/mapred/local
mkdir -p /home/hadoop_space/mapred/system
chown -R hadoop:hadoop /home/hadoop_space/
```

修改目录 /home/hadoop 的拥有者（因为该目录用于安装 hadoop，用户对其必须有 rwx 权

限。)

```
chown -R hadoop:hadoop /home/hadoop
chown -R hadoop:hadoop /home/hadoop_space
```

1.3 安装 jdk

1.3.1. 下载 JDK

选定 linux 环境版本，下载到的文件是：jdk-6u45-linux-x64.bin

1.3.2. 复制 JDK 安装包到系统指定文件夹

把下载的安装文件（jdk-6u45-linux-x64.bin）拷到 linux 路径/usr/local/下。

1.3.3. 给安装文件赋予权限

a).进入目录，命令：cd /usr/local/java

b).赋予权限，命令：chmod +x jdk-6u45-linux-x64.bin

(如果因权限问题执行失败，则加上 su，即 su chmod u+x jdk-6u45-linux-x64.bin)

1.3.4 安装 JDK

开始安装,在控制台执行命令：./jdk-6u45-linux-x64.bin

(如果因权限问题执行失败，则加上 su，即 sud./jdk-6u45-linux-x64.bin)

文件会被安装到当前目录 /usr/local/jdk1.6.0_45

删除安装文件 rm jdk-6u45-linux-x64.bin

安装完成后，修改/usr/local/jdk1.6.0_45 目录拥有着为 hadoop 用户，

```
chown -R hadoop:hadoop /usr/local/jdk1.6.0_45
```

1.3.5 Java 环境变量配置

root 用户登陆，命令行中执行命令” vim /etc/profile”，并加入以下内容，配置环境变量(注意/etc/profile 这个文件很重要，后面 Hadoop 的配置还会用到)。

```
#java
```

```
JAVA_HOME=/usr/local/jdk1.6.0_45
```

```
JRE_HOME=/usr/local/jdk1.6.0_45/jre
```

```
PATH=$JAVA_HOME/bin:$JRE_HOME/bin:$PATH
```

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

```
export JAVA_HOME CLASSPATH PATH
```

保存并退出，执行以下命令使配置生效

```
chmod +x /etc/profile
```

```
source /etc/profile
```

配置完毕，在命令行中使用命令“java -version”可以判断是否成功。在 hadoop 用户下测试 java -version，一样成功。

```
[root@NameNode hadoop]# java -version
```

```
java version "1.6.0_45"
```

```
Java(TM) SE Runtime Environment (build 1.6.0_45-b06)
```

```
Java HotSpot(TM) 64-Bit Server VM (build 20.45-b01, mixed mode)
```

2. 配置 DNS 解析主机名

说明：在生产的 hadoop 集群环境中，由于服务器可能会有许多台，通过配置 DNS 映射机器名，相比配置/etc/host 方法，可以避免在每个节点都配置各自的 host 文件，而且在新增节点时也不需要修改每个节点的/etc/host 的主机名-IP 映射文件。减少了配置步骤和时间，便于管理。

详细步骤见：【hadoop 大规模应用之 CentOS6.4+DNS-BIND 域名服务器配置基础详解】

<http://blog.csdn.net/liema2000/article/details/9076333>

3. 建立 hadoop 运行账号

在所有的节点创建 hadoop 运行账号

```
[root@gc ~]# groupadd hadoop
```

```
[root@gc ~]# useradd -g hadoop hadoop
```

说明：在大的 hadoop 集群安装环境中，这步可以在批量安装 linux 系统之前完成，然后再进行系统的复制。

4. 通过 NFS 配置 ssh 免密码连入

说明：通过 NFS 配置 ssh 免密码连入时，当我们有新的节点接入时，不再需要分别向其它节点各自添加自己公钥信息，只需要把公钥信息追加到共享的 authorized_keys 公钥当中，其它节点就直接指向最新的公钥文件。便于分配公钥和管理。

详细步骤见：【hadoop 大规模应用之 NFS 安装及 SSH 免密码验证配置】

<http://blog.csdn.net/liema2000/article/details/9076357>

5. 解压 hadoop 安装包

--可先一某节点(如 NameNode)解压配置文件

```
[root@NameNode hadoop]# pwd
```

```

/usr/local/hadoop
[root@NameNode hadoop]#
[root@NameNode hadoop]# tar -zxvf hadoop-1.0.4.tar.gz

```

6. 配置 Hadoop

(1) 配置 Hadoop 的配置文件

(a) 配置 hadoop-env.sh

```

$ vim /usr/local/hadoop-1.0.4/conf/hadoop-env.sh
# set java environment
export JAVA_HOME=/usr/local/jdk1.6.0_45

```

```
export HADOOP_HOME_WARN_SUPPRESS=1
```

```
export HADOOP_CLASSPATH=/usr/local/hadoop-1.0.4:/usr/local/hadoop-1.0.4/lib
```

(b) 配置 **/etc/profile**

```
#hadoop
```

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

```
export HADOOP_HOME_WARN_SUPPRESS=1
```

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

```
export PATH
```

修改完，重启/etc/profile

```
[root@NameNode ~]# source /etc/profile
```

(c) 配置 conf/hadoop-site.xml

Hadoop-0.20.2 之后的版本请分别配置 core-site.xml, hdfs-site.xml 和 mapred-site.xml 三个配置文件。

配置文件名 (xml)	字段名	字段值	说明
core-site	fs.default.name	hdfs://NameNode:9000	
	hadoop.tmp.dir	/home/hadoop_space/tmp	临时目录
hdfs-site	dfs.name.dir	/home/hadoop_space/hdfs/name	namenode 上存储 hdfs 名字空间元数据
	dfs.data.dir	/home/hadoop_space/hdfs/data	datanode 上数据块的物理存储位置
	dfs.replication	3	副本个数, 不配置默认是 3, 应小于 datanode 机器数量
mapred-site	mapred.job.tracker	NameNode:9001	jobtracker 标识: 端口号, 不是 URI
	mapred.local.dir	/home/hadoop_space/mapred/local	tasktracker 上执行 mapreduce 程序时的本地目录

	mapred.system.dir	/home/hadoop_space/mapred/system	这个是 hdfs 中的目录，存储执行 mr 程序时的共享文件
--	-------------------	----------------------------------	--------------------------------

编辑 core-site.xml

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

<configuration>
  <property>
    <name>fs.default.name</name>
    <value>hdfs://NameNode.hdcw.org:9000</value>
  </property>
  <property>
    <name>hadoop.tmp.dir</name>
    <value>/home/hadoop_space/tmp</value>
  </property>
</configuration>
```

编辑 hdfs-site.xml

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!-- Put site-specific property overrides in this file. -->
<configuration>
  <property>
    <name>dfs.name.dir</name>
    <value>/home/hadoop_space/hdfs/name</value>
  </property>
  <property>
    <name>dfs.data.dir</name>
    <value>/home/hadoop_space/hdfs/data</value>
  </property>
  <property>
    <name>dfs.replication</name>
    <value>3</value>
  </property>
</configuration>
```

编辑 mapred-site.xml

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!-- Put site-specific property overrides in this file. -->
<configuration>
```



```

<property>
  <name>mapred.job.tracker</name>
  <value>NameNode.hdcw.org:9001</value>
</property>
<property>
  <name>mapred.local.dir</name>
  <value>/home/hadoop_space/mapred/local</value>
</property>
<property>
  <name>mapred.system.dir</name>
  <value>/home/hadoop_space/mapred/system</value>
</property>
</configuration>

```

(d) 配置 masters 文件, 把 localhost 修改为 namenode 的主机名

```
NameNode.hdcw.org
```

(e) 配置 slaves 文件, 删除 localhost, 加入所有 datanode 的主机名

```

DataNode1.hdcw.org
DataNode2.hdcw.org
DataNode3.hdcw.org
DataNode4.hdcw.org

```

Namenode 节点上 root 执行执行

```
chown -R hadoop:hadoop /usr/local
```

```
chown -R hadoop:hadoop /home/hadoop_space/
```

7. 向各节点复制 hadoop (awk 命令)

--把 NameNode 主机上面 hadoop 配置好的文件分别 copy 到各节点

--原来 copy 的方法是一个个命令执行, 如下

```
[hadoop@NameNode hadoop]$ scp -rp hadoop-1.0.4 hadoop@DataNode1:/usr/local/
```

```
[hadoop@NameNode hadoop]$ scp -rp hadoop-1.0.4 hadoop@DataNode2:/usr/local/
```

--但站在 hadoop 大集群的角度, 上面方法会很费时, 我们可用 awk 命令生成批量执行的脚本, 批量执行, 省时省力, 如下:

```
[root@NameNode local]# cat $HADOOP_HOME/conf/slaves | awk '{print "scp -rp hadoop-1.0.4 hadoop@"$1":/usr/local/"}' > scp.sh
```

```
[root@NameNode local]# cat scp.sh
```

```
scp -rp hadoop-1.0.4 hadoop@DataNode1.hdcw.org:/usr/local/
```

```
scp -rp hadoop-1.0.4 hadoop@DataNode2.hdcw.org:/usr/local/
```

```
scp -rp hadoop-1.0.4 hadoop@DataNode3.hdcw.org:/usr/local/
```

```
scp -rp hadoop-1.0.4 hadoop@DataNode4.hdcw.org:/usr/local/
```

切换到 hadoop 用户，执行复制

复制 java

```
[root@NameNode local]# cat $HADOOP_HOME/conf/slaves | awk '{print "scp -rp /usr/local/jdk1.6.0_45 hadoop@"$1":/usr/local/"}' > scp_java.sh
```

复制 hadoop_space

```
[root@NameNode local]# cat $HADOOP_HOME/conf/slaves | awk '{print "scp -rp /home/hadoop_space hadoop@"$1":/home/"}' > scp_hadoop.sh
```

#复制/etc/profile

```
[root@NameNode local]# cat $HADOOP_HOME/conf/slaves | awk '{print "scp -rp /etc/profile root@"$1":/etc/"}' > scp_profile.sh
```

root 用户执行

8. 格式化 namenode

--在 namenode 节点进行格式化

```
[hadoop@NameNode local]$ cd $HADOOP_HOME
```

```
[hadoop@NameNode hadoop-1.0.4]$ bin/hadoop namenode -format
```

```
13/06/11 20:28:16 INFO namenode.NameNode: STARTUP_MSG:
```

```
/*****
```

```
STARTUP_MSG: Starting NameNode
```

```
STARTUP_MSG: host = java.net.UnknownHostException: NameNode: NameNode
```

```
STARTUP_MSG: args = [-format]
```

```
STARTUP_MSG: version = 1.0.4
```

```
STARTUP_MSG: build = https://svn.apache.org/repos/asf/hadoop/common/branches/branch-1.0 -r 1393290; compiled by 'hortonfo' on Wed Oct 3 05:13:58 UTC 2012
```

```
*****/
```

```
Re-format filesystem in /home/hadoop_space/hdfs/name ? (Y or N) Y
```

```
13/06/11 20:28:18 INFO util.GSet: VM type = 64-bit
```

```
13/06/11 20:28:18 INFO util.GSet: 2% max memory = 19.33375 MB
```

```
13/06/11 20:28:18 INFO util.GSet: capacity = 2^21 = 2097152 entries
```

```
13/06/11 20:28:18 INFO util.GSet: recommended=2097152, actual=2097152
```

```
13/06/11 20:28:19 INFO namenode.FSNamesystem: fsOwner=hadoop
```

```
13/06/11 20:28:19 INFO namenode.FSNamesystem: supergroup=supergroup
```

```
13/06/11 20:28:19 INFO namenode.FSNamesystem: isPermissionEnabled=true
```

```
13/06/11 20:28:19 INFO namenode.FSNamesystem: dfs.block.invalidate.limit=100
```

```
13/06/11 20:28:19 INFO namenode.FSNamesystem: isAccessTokenEnabled=false accessKeyUpdateInterval=0 min(s), accessTokenLifetime=0 min(s)
```

```
13/06/11 20:28:19 INFO namenode.NameNode: Caching file names occuring more than 10 times
```

```
13/06/11 20:28:29 INFO common.Storage: Image file of size 112 saved in 0 seconds.
```

```
13/06/11 20:28:29 INFO common.Storage: Storage directory /home/hadoop_space/hdfs/name has been successfully formatted.
```

```
13/06/11 20:28:29 INFO namenode.NameNode: SHUTDOWN_MSG:
```

```
/*****
```

```
SHUTDOWN_MSG: Shutting down NameNode at java.net.UnknownHostException: NameNode: NameNode
```

*****/

[hadoop@NameNode hadoop-1.0.4]\$

```
[hadoop@NameNode local]$ cd $HADOOP_HOME
hadoop@NameNode hadoop-1.0.4$ bin/hadoop namenode -format
13/06/11 20:28:16 INFO namenode.NameNode: STARTUP_MSG:
*****
STARTUP_MSG: Starting NameNode
STARTUP_MSG: host = java.net.UnknownHostException: NameNode: NameNode
STARTUP_MSG: args = [-format]
STARTUP_MSG: version = 1.0.4
STARTUP_MSG: build = https://svn.apache.org/repos/asf/hadoop/common/branches/branch-1.0 -r 1393290; compiled by 'hortonfo' on wed oct 3 05:13:58 UTC 2012
*****
Re-format filesystem in /home/hadoop_space/hdfs/name ? (y or n) Y
13/06/11 20:28:18 INFO util.GSet: VM type = 64-bit
13/06/11 20:28:18 INFO util.GSet: 2% max memory = 19.33375 MB
13/06/11 20:28:18 INFO util.GSet: capacity = 2421 = 2097152 entries
13/06/11 20:28:18 INFO util.GSet: recommended=2097152, actual=2097152
13/06/11 20:28:19 INFO namenode.FSNamesystem: fsowner=hadoop
13/06/11 20:28:19 INFO namenode.FSNamesystem: supergroup=supergroup
13/06/11 20:28:19 INFO namenode.FSNamesystem: isPermissionEnabled=true
13/06/11 20:28:19 INFO namenode.FSNamesystem: dfs.block.invalidate.limit=100
13/06/11 20:28:19 INFO namenode.FSNamesystem: isAccessTokenEnabled=false, accessKeyUpdateInterval=0 min(s), accessTokenLifetime=0 min(s)
13/06/11 20:28:19 INFO namenode.NameNode: caching file names occurring more than 10 times
13/06/11 20:28:20 INFO common.Storage: Image file of size 112 saved in 0 seconds.
13/06/11 20:28:20 INFO common.Storage: Storage directory /home/hadoop_space/hdfs/name has been successfully formatted.
13/06/11 20:28:29 INFO namenode.NameNode: SHUTDOWN_MSG:
*****
SHUTDOWN_MSG: Shutting down NameNode at java.net.UnknownHostException: NameNode: NameNode
*****
[hadoop@NameNode hadoop-1.0.4]$
```

9. 启动 hadoop

--在 NameNode 节点启动 hadoop 守护进程

[hadoop@NameNode hadoop-1.0.4]\$ bin/start-all.sh

starting namenode, logging to /usr/local/hadoop-1.0.4/libexec/../logs/hadoop-hadoop-namenode-NameNode.out

DataNode1.hdcw.org: starting datanode, logging to /usr/local/hadoop-1.0.4/libexec/../logs/hadoop-hadoop-datanode-DataNode1.out

DataNode4.hdcw.org: starting datanode, logging to /usr/local/hadoop-1.0.4/libexec/../logs/hadoop-hadoop-datanode-DataNode4.out

DataNode2.hdcw.org: starting datanode, logging to /usr/local/hadoop-1.0.4/libexec/../logs/hadoop-hadoop-datanode-DataNode2.out

DataNode3.hdcw.org: starting datanode, logging to /usr/local/hadoop-1.0.4/libexec/../logs/hadoop-hadoop-datanode-DataNode3.out

NameNode.hdcw.org: starting secondarynamenode, logging to

/usr/local/hadoop-1.0.4/libexec/../logs/hadoop-hadoop-secondarynamenode-NameNode.out

starting jobtracker, logging to /usr/local/hadoop-1.0.4/libexec/../logs/hadoop-hadoop-jobtracker-NameNode.out

DataNode2.hdcw.org: starting tasktracker, logging to /usr/local/hadoop-1.0.4/libexec/../logs/hadoop-hadoop-tasktracker-DataNode2.out

DataNode4.hdcw.org: starting tasktracker, logging to /usr/local/hadoop-1.0.4/libexec/../logs/hadoop-hadoop-tasktracker-DataNode4.out

DataNode1.hdcw.org: starting tasktracker, logging to /usr/local/hadoop-1.0.4/libexec/../logs/hadoop-hadoop-tasktracker-DataNode1.out

DataNode3.hdcw.org: starting tasktracker, logging to /usr/local/hadoop-1.0.4/libexec/../logs/hadoop-hadoop-tasktracker-DataNode3.out

[hadoop@NameNode hadoop-1.0.4]\$

```
[hadoop@NameNode hadoop-1.0.4]$ bin/start-all.sh
starting namenode, logging to /usr/local/hadoop-1.0.4/libexec/../logs/hadoop-hadoop-namenode-NameNode.out
DataNode1.hdcw.org: starting datanode, logging to /usr/local/hadoop-1.0.4/libexec/../logs/hadoop-hadoop-datanode-DataNode1.out
DataNode4.hdcw.org: starting datanode, logging to /usr/local/hadoop-1.0.4/libexec/../logs/hadoop-hadoop-datanode-DataNode4.out
DataNode2.hdcw.org: starting datanode, logging to /usr/local/hadoop-1.0.4/libexec/../logs/hadoop-hadoop-datanode-DataNode2.out
DataNode3.hdcw.org: starting datanode, logging to /usr/local/hadoop-1.0.4/libexec/../logs/hadoop-hadoop-datanode-DataNode3.out
NameNode.hdcw.org: starting secondarynamenode, logging to /usr/local/hadoop-1.0.4/libexec/../logs/hadoop-hadoop-secondarynamenode-NameNode.out
starting jobtracker, logging to /usr/local/hadoop-1.0.4/libexec/../logs/hadoop-hadoop-jobtracker-NameNode.out
DataNode2.hdcw.org: starting tasktracker, logging to /usr/local/hadoop-1.0.4/libexec/../logs/hadoop-hadoop-tasktracker-DataNode2.out
DataNode4.hdcw.org: starting tasktracker, logging to /usr/local/hadoop-1.0.4/libexec/../logs/hadoop-hadoop-tasktracker-DataNode4.out
DataNode1.hdcw.org: starting tasktracker, logging to /usr/local/hadoop-1.0.4/libexec/../logs/hadoop-hadoop-tasktracker-DataNode1.out
DataNode3.hdcw.org: starting tasktracker, logging to /usr/local/hadoop-1.0.4/libexec/../logs/hadoop-hadoop-tasktracker-DataNode3.out
[hadoop@NameNode hadoop-1.0.4]$
```

10. 用 jps 检验各后台进程是否成功启动

--在 NameNode 节点查看后台进程

[hadoop@NameNode hadoop-1.0.4]\$ jps

28219 JobTracker

27989 NameNode

28141 SecondaryNameNode

28376 Jps

[hadoop@NameNode hadoop-1.0.4]\$

```
[hadoop@NameNode hadoop-1.0.4]$
[hadoop@NameNode hadoop-1.0.4]$ jps
28219 JobTracker
27989 NameNode
28141 SecondaryNameNode
28376 Jps
[hadoop@NameNode hadoop-1.0.4]$
```

--在 datanode 节点查看后台进程

```
[hadoop@DataNode1 home]$ jps
```

```
56556 DataNode
```

```
56681 Jps
```

```
56635 TaskTracker
```

```
[hadoop@DataNode1 home]$
```

```
[grid@hotel03 ~]$ /usr/java/jdk1.6.0_18/bin/jps
```

```
29936 TaskTracker
```

```
29828 DataNode
```

```
30924 Jps
```

```
[hadoop@DataNode1 home]$ jps
56826 DataNode
56945 Jps
56906 TaskTracker
[hadoop@DataNode1 home]$
```

```
[hadoop@DataNode2 local]$ jps
56575 DataNode
56696 Jps
56655 TaskTracker
[hadoop@DataNode2 local]$
```

```
[hadoop@DataNode3 ~]$ jps
56674 TaskTracker
56594 DataNode
56721 Jps
[hadoop@DataNode3 ~]$
```

```
[hadoop@DataNode4 ~]$ jps
56689 TaskTracker
56609 DataNode
56730 Jps
[hadoop@DataNode4 ~]$
```

11. 通过 web 了解 Hadoop 的活动

通过用浏览器和 http 访问 jobtracker 所在节点的 50030 端口监控 jobtracker

Jobtracker 监控

<http://192.168.1.221:50030/jobtracker.jsp>

通过用浏览器和 http 访问 namenode 所在节点的 50070 端口监控集群

<http://192.168.1.221:50070/dfshealth.jsp>

常见问题

1、权限问题

```
[hadoop @NameNode local]# scp -r hadoop-1.0.4 hadoop@DataNode1.hdcw.org:/usr/local
```

scp: /usr/local/hadoop-1.0.4: Permission denied

[hadoop @NameNode local]

原因: /usr/local 文件夹的属主是 root, hadoop 没有权限操作

解决方法 1: 规划时, 把 hadoop 的目录放到/home/hadoop/下面

解决方法 2: 更改/usr/local/的属主为 hadoop

现在采用的是方法 2