

Hadoop-2.2.0 编译安装步骤

Linux 公社（LinuxIDC.com）于 2006 年 9 月 25 日注册并开通网站，Linux 现在已经成为一种广受关注和支持的一种操作系统，IDC 是互联网数据中心，LinuxIDC 就是关于 Linux 的数据中心。

Linux 公社是专业的 Linux 系统门户网站，实时发布最新 Linux 资讯，包括 Linux、Ubuntu、Fedora、RedHat、红旗 Linux、Linux 教程、Linux 认证、SUSE Linux、Android、Oracle、Hadoop 等技术。

并被收录到 Google 网页目录-计算机 > 软件 > 操作系统 > Linux 目录下。

Linux 公社（LinuxIDC.com）设置了有一定影响力的 Linux 专题栏目。



包括：

[Ubuntu 专题](#)

[Fedora 专题](#)

[Android 专题](#)

[Oracle 专题](#)

[Hadoop 专题](#)

[RedHat 专题](#)

[SUSE 专题](#)

[红旗 Linux 专题](#)

环境： 系统 CentOS 6.3 64 位 * 3

Master 192.168.80.90

Slave1 192.168.90.91

Slave2 192.168.80.92

落叶 2013.11.13

软件: [Hadoop-2.2.0-src.tar.gz](#)

一，准备环境

- 1.1 分别设置三台节点 ip 地址（固定） 此步骤所有节点都要操作

```
vi /etc/sysconfig/network-scripts/ifcfg-eth0
```

- 1.2 修改三台节点主机名 此步骤所有节点都要操作

```
vi /etc/sysconfig/network
```

以上两步骤需要重启服务器后才生效

- 1.3 创建 hadoop 用户 此步骤所有节点都要操作

```
useradd hadoop
```

- 1.4 添加 hosts 记录 此步骤所有节点都要操作

```
vi /etc/hosts
192.168.80.90    Master.Hadoop
192.168.80.91    Slave1.Hadoop
192.168.80.92    Slave2.Hadoop
```

- 1.5 建立 hadoop 用户之间相互信任（互相无密码登陆） 此步骤所有节点都要操作

切换至 hadoop 用户下执行 `ssh-keygen -t rsa`

然后将自己的公钥以及其他服务器 hadoop 用户的公钥放置 hadoop 宿主目录下的 `.ssh/authorized_keys` 中，注意 `authorized_keys` 默认不存在需要手动创建，并且 必须设置权限为 `600`

- 1.6 安装 java（建议每台机器的 JAVA_HOME 路径信息相同） 此步骤所有节点都要操作

下载 jdk（<http://www.oracle.com/technetwork/java/javase/downloads/index.html>）

下载后解压至自己指定的目录如 `/usr/java`

```
[root@Master java]# ll
总用量 4
drwxrwxrwx. 10 root root 4096 10月 30 15:04 jdk1.6.0_23
[root@Master java]# pwd
/usr/java
```

添加变量：以 root 用户 `vi /etc/profile`

根据自己路径在最后添加以下三行

```
export JAVA_HOME=/usr/java/jdk1.6.0_23
```

```
export CLASSPATH=$CLASSPATH:$JAVA_HOME/lib:$JAVA_HOME/jre/lib
export PATH=$PATH:$JAVA_HOME/bin:$JAVA_HOME/jre/bin
```

添加之后执行 `source /etc/profile` 重新加载 建议 root 用户和 hadoop 用户都操作 此步

1.7 安装基本应用程序（此步骤要求联公网） 此步骤只需在 master 上操作

```
yum -y install svn ncurses-devel gcc*
```

```
yum -y install lzo-devel zlib-devel autoconf automake libtool cmake openssl-devel
```

1.8 安装 maven

下载 apache-maven-3.1.1-bin.tar.gz

（下载地址 <http://maven.apache.org/download.cgi> ）

并解压 `tar zxvf apache-maven-3.1.1-bin.tar.gz`

`mv apache-maven-3.1.1 maven` （改名）

```
[root@Master hadoop]# ll
总用量 5372
drwxr-xr-x 6 root root    4096 11月 11 15:45 apache-maven-3.1.1
-rw-r--r-- 1 root root 5494427 11月 11 15:45 apache-maven-3.1.1-bin.tar.gz
[root@Master hadoop]# mv apache-maven-3.1.1 maven
[root@Master hadoop]# ll
总用量 5372
-rw-r--r-- 1 root root 5494427 11月 11 15:45 apache-maven-3.1.1-bin.tar.gz
drwxr-xr-x 6 root root    4096 11月 11 15:45 maven
[root@Master hadoop]# cd maven/
[root@Master maven]# ll
总用量 40
drwxr-xr-x 2 root root    4096 11月 11 15:45 bin
drwxr-xr-x 2 root root    4096 11月 11 15:45 boot
drwxr-xr-x 3 root root    4096 9月 17 23:19 conf
drwxr-xr-x 3 root root    4096 11月 11 15:45 lib
-rw-r--r-- 1 root root 14737 9月 17 23:24 LICENSE
-rw-r--r-- 1 root root   182 9月 17 23:24 NOTICE
-rw-r--r-- 1 root root 2513 9月 17 23:19 README.txt
[root@Master maven]# pwd
/home/hadoop/maven
```

注意一下 maven 存放路径，可以自定义路径

添加变量：以 root 用户 `vi /etc/profile`

根据自己路径在最后添加以下两行

```
export MAVEN_HOME=/home/hadoop/maven
export PATH=$PATH:$MAVEN_HOME/bin
```

添加之后执行 `source /etc/profile` 重新加载

1.9 安装 protobuf（不安装，编译将无法完成）

下载：<http://code.google.com/p/protobuf/downloads/detail?name=protobuf-2.5.0.tar.gz>

解压，设置属主为 root `chown -R root.root protobuf-2.5.0`

编译安装 protobuf

- ① cd protobuf-2.5.0
- ② ./configure
- ③ make
- ④ make install

检测 protoc --version

```
[root@Master protobuf-2.5.0]# protoc --version
libprotoc 2.5.0
```





二，编译安装 hadoop-2.2.0

2.1 下载 hadoop-2.2.0-src.tar.gz

Hadoop-2.2.0 源码包下载地址 <http://apache.claz.org/hadoop/common/hadoop-2.2.0/>

红色标志 hadoop-2.2.0-src.tar.gz 为源码包

Index of /hadoop/common/hadoop-2.2.0

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
 Parent Directory	-	-	-
 hadoop-2.2.0-src.tar.gz	07-Oct-2013 01:46	19M	
 hadoop-2.2.0-src.tar.gz.mds	07-Oct-2013 01:46	1.1K	
 hadoop-2.2.0.tar.gz	07-Oct-2013 01:46	104M	
 hadoop-2.2.0.tar.gz.mds	07-Oct-2013 01:47	958	

Apache/2.2.3 (CentOS) Server at apache.claz.org Port 80

2.2 解压 hadoop-2.2.0-src.tar.gz

```
[root@Master hadoop]# ll
总用量 108
-rw-r--r--  1 root root  9968 10月  7 13:53 BUILDING.txt
drwxr-xr-x  2 root root  4096 10月  7 14:45 dev-support
drwxr-xr-x  3 root root  4096 10月  7 14:45 hadoop-assemblies
drwxr-xr-x  2 root root  4096 10月  7 14:45 hadoop-client
drwxr-xr-x  8 root root  4096 10月  7 14:45 hadoop-common-project
drwxr-xr-x  2 root root  4096 10月  7 14:45 hadoop-dist
drwxr-xr-x  6 root root  4096 10月  7 14:45 hadoop-hdfs-project
drwxr-xr-x 10 root root  4096 10月  7 14:45 hadoop-mapreduce-project
drwxr-xr-x  3 root root  4096 10月  7 14:45 hadoop-maven-plugins
drwxr-xr-x  2 root root  4096 10月  7 14:45 hadoop-minicluster
drwxr-xr-x  3 root root  4096 10月  7 14:45 hadoop-project
drwxr-xr-x  2 root root  4096 10月  7 14:45 hadoop-project-dist
drwxr-xr-x 11 root root  4096 10月  7 14:45 hadoop-tools
drwxr-xr-x  3 root root  4096 10月  7 14:45 hadoop-yarn-project
-rw-r--r--  1 root root 15164 10月  7 14:46 LICENSE.txt
-rw-r--r--  1 root root   101 10月  7 14:46 NOTICE.txt
-rw-r--r--  1 root root 16569 10月  7 13:53 pom.xml
-rw-r--r--  1 root root  1366 10月  7 14:46 README.txt
```

2.3 开始编译:

执行命令 `mvn clean package -Pdist,native -DskipTests -Dtar`

此步骤要求主机能访问公网，并且此步骤很漫长，由于服务器配置和网络速度的不同因素，，时间为 1 小时到 30 分钟不等，等待吧……

完成结果类似一下截图

```
[INFO] hadoop-yarn-site ..... SUCCESS [0.408s]
[INFO] hadoop-yarn-project ..... SUCCESS [5:32.456s]
[INFO] hadoop-mapreduce-client-common ..... SUCCESS [32.607s]
[INFO] hadoop-mapreduce-client-shuffle ..... SUCCESS [6.163s]
[INFO] hadoop-mapreduce-client-app ..... SUCCESS [20.372s]
[INFO] hadoop-mapreduce-client-hs ..... SUCCESS [9.406s]
[INFO] hadoop-mapreduce-client-jobclient ..... SUCCESS [11.093s]
[INFO] hadoop-mapreduce-client-hs-plugins ..... SUCCESS [3.445s]
[INFO] Apache Hadoop MapReduce Examples ..... SUCCESS [11.290s]
[INFO] hadoop-mapreduce ..... SUCCESS [7.810s]
[INFO] Apache Hadoop MapReduce Streaming ..... SUCCESS [9.056s]
[INFO] Apache Hadoop Distributed Copy ..... SUCCESS [30.791s]
[INFO] Apache Hadoop Archives ..... SUCCESS [4.574s]
[INFO] Apache Hadoop Rumen ..... SUCCESS [13.463s]
[INFO] Apache Hadoop Gridmix ..... SUCCESS [7.797s]
[INFO] Apache Hadoop Data Join ..... SUCCESS [5.379s]
[INFO] Apache Hadoop Extras ..... SUCCESS [5.724s]
[INFO] Apache Hadoop Pipes ..... SUCCESS [14.424s]
[INFO] Apache Hadoop Tools Dist ..... SUCCESS [4.215s]
[INFO] Apache Hadoop Tools ..... SUCCESS [0.046s]
[INFO] Apache Hadoop Distribution ..... SUCCESS [52.694s]
[INFO] Apache Hadoop Client ..... SUCCESS [13.703s]
[INFO] Apache Hadoop Mini-Cluster ..... SUCCESS [0.484s]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 32:46.166s
[INFO] Finished at: Mon Nov 11 17:29:56 CST 2013
[INFO] Final Memory: 92M/241M
[INFO] -----
```

挺快的，用了 32 分钟

编译成功后，`/hadoop-dist/target/hadoop-2.2.0.tar.gz` 就是我们需要的文件了

将编译好的 `/hadoop-dist/target/hadoop-2.2.0.tar.gz` 文件分别拷贝至其他节点，并将其解压，设置其属主，属组都为 `hadoop` 用户（路径自定义，注：所有节点 `hadoop` 路径必须相同，我这里所有服务器节点都放置 `/home/hadoop/`）

落叶 2013.11.13

然后添加环境变量: `vi /etc/profile` 此步骤所有节点都要操作
添加以下两条

```
export HADOOP_HOME=/home/hadoop/hadoop-2.2.0
export PATH=$PATH:$HADOOP_HOME/bin
```

添加之后执行 `source /etc/profile` 重新加载 建议 root 用户和 hadoop 用户都操作此步

2.4 Master 上操作

解压/hadoop-dist/target/hadoop-2.2.0.tar.gz 到/home/hadoop
进入配置文件目录

```
cd /home/hadoop/hadoop-2.2.0/etc/hadoop
```

```
[root@Master hadoop]# pwd
/home/hadoop/hadoop-2.2.0/etc/hadoop
[root@Master hadoop]# ll
总用量 120
-rw-r--r-- 1 hadoop hadoop 3560 11月 11 17:28 capacity-scheduler.xml
-rw-r--r-- 1 hadoop hadoop 1335 11月 11 17:28 configuration.xsl
-rw-r--r-- 1 hadoop hadoop 318 11月 11 17:28 container-executor.cfg
-rw-r--r-- 1 hadoop hadoop 774 11月 11 17:28 core-site.xml
-rw-r--r-- 1 hadoop hadoop 3589 11月 11 17:28 hadoop-env.cmd
-rw-r--r-- 1 hadoop hadoop 3390 11月 11 17:28 hadoop-env.sh
-rw-r--r-- 1 hadoop hadoop 1774 11月 11 17:28 hadoop-metrics2.properties
-rw-r--r-- 1 hadoop hadoop 2490 11月 11 17:28 hadoop-metrics.properties
-rw-r--r-- 1 hadoop hadoop 9257 11月 11 17:28 hadoop-policy.xml
-rw-r--r-- 1 hadoop hadoop 775 11月 11 17:28 hdfs-site.xml
-rw-r--r-- 1 hadoop hadoop 1180 11月 11 17:28 httpfs-env.sh
-rw-r--r-- 1 hadoop hadoop 1657 11月 11 17:28 httpfs-log4j.properties
-rw-r--r-- 1 hadoop hadoop 21 11月 11 17:28 httpfs-signature.secret
-rw-r--r-- 1 hadoop hadoop 620 11月 11 17:28 httpfs-site.xml
-rw-r--r-- 1 hadoop hadoop 9116 11月 11 17:28 log4j.properties
-rw-r--r-- 1 hadoop hadoop 918 11月 11 17:28 mapred-env.cmd
-rw-r--r-- 1 hadoop hadoop 1383 11月 11 17:28 mapred-env.sh
-rw-r--r-- 1 hadoop hadoop 4113 11月 11 17:28 mapred-queues.xml.template
-rw-r--r-- 1 hadoop hadoop 758 11月 11 17:28 mapred-site.xml.template
-rw-r--r-- 1 hadoop hadoop 10 11月 11 17:28 slaves
-rw-r--r-- 1 hadoop hadoop 2316 11月 11 17:28 ssl-client.xml.example
-rw-r--r-- 1 hadoop hadoop 2251 11月 11 17:28 ssl-server.xml.example
-rw-r--r-- 1 hadoop hadoop 2178 11月 11 17:28 yarn-env.cmd
-rw-r--r-- 1 hadoop hadoop 4084 11月 11 17:28 yarn-env.sh
-rw-r--r-- 1 hadoop hadoop 690 11月 11 17:28 yarn-site.xml
```

在这里主要修改一下几个文件: (文件内容有些参数根据自己实际情况修改)

slaves core-site.xml hadoop-env.sh hdfs-site.xml
mapred-site.xml (此文件默认不存在) yarn-env.sh yarn-site.xml

2.4.1 修改 slaves 文件, 添加以下两行 (目的: 添加 slave 节点, 建议写 ip 地址)

```
192.168.80.91
192.168.80.92
```

2.4.2 修改 core-site.xml 添加一下内容

```
<configuration>
```

```
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://192.168.80.90:9000</value>
  </property>

  <property>
    <name>io.file.buffer.size</name>
    <value>131072</value>
  </property>

  <property>

    <name>hadoop.tmp.dir</name>
    <value>file:/home/hadoop/hadoop-2.2.0/temp</value>
    <description>Abase for other temporary directories.</description>

  </property>

  <property>
    <name>hadoop.proxyuser.hduser.hosts</name>
    <value>*</value>
  </property>

  <property>

    <name>hadoop.proxyuser.hduser.groups</name>
    <value>*</value>
  </property>
```

```
</configuration>
```

2.4.3 修改 **hdfs-site.xml** 文件，添加以下内容

```
<configuration>

<property>
  <name>dfs.namenode.secondary.http-address</name>
  <value>192.168.80.90:9001</value>
</property>
<property>
  <name>dfs.namenode.name.dir</name>
  <value>file:/home/hadoop/hadoop-2.2.0/dfs/name</value>
</property>
<property>
```

```
<name>dfs.datanode.data.dir</name>
<value>file:/home/hadoop/hadoop-2.2.0/dfs/data</value>
</property>
<property>
  <name>dfs.replication</name>
  <value>1</value>
</property>
<property>
  <name>dfs.webhdfs.enabled</name>
  <value>true</value>
</property>

</configuration>
```

2.4.4 修改 **mapred-site.xml** 文件，添加以下内容

```
<configuration>

<property>
  <name>mapreduce.framework.name</name>
  <value>yarn</value>
</property>
<property>
  <name>mapreduce.jobhistory.address</name>
  <value>192.168.80.90:10020</value>
</property>
<property>
  <name>mapreduce.jobhistory.webapp.address</name>
  <value>192.168.80.90:19888</value>
</property>

</configuration>
```

2.4.5 修改 **yarn-site.xml** 文件，添加以下内容

```
<configuration>

<property>
  <name>yarn.nodemanager.aux-services</name>
  <value>mapreduce_shuffle</value>
</property>
<property>
  <name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>
  <value>org.apache.hadoop.mapred.ShuffleHandler</value>
</property>
<property>
```



```
<name>yarn.resourcemanager.address</name>
<value>192.168.80.90:8032</value>
</property>
<property>
  <name>yarn.resourcemanager.scheduler.address</name>
  <value>192.168.80.90:8030</value>
</property>
<property>
  <name>yarn.resourcemanager.resource-tracker.address</name>
  <value>192.168.80.90:8031</value>
</property>
<property>
  <name>yarn.resourcemanager.admin.address</name>
  <value>192.168.80.90:8033</value>
</property>
<property>
  <name>yarn.resourcemanager.webapp.address</name>
  <value>192.168.80.90:8088</value>
</property>
</configuration>
```

2.4.6 修该 **hadoop-env.sh** 文件

修改 JAVA_HOME 值 (export JAVA_HOME=/usr/java/jdk1.6.0_23)

2.4.7 修改 **yarn-env.sh** 文件

修改 JAVA_HOME 值 (export JAVA_HOME=/usr/java/jdk1.6.0_23)

2.4.8 传送文件至各个节点保持一致

配置文件修改完了，然后将修改后的所有文件都复制（覆盖）到各个节点 **hadoop** 目录下

```
scp hadoop-env.sh hdfs-site.xml mapred-site.xml slaves yarn-env.sh yarn-site.xml hadoop@192.168.80.91
scp hadoop-env.sh hdfs-site.xml mapred-site.xml slaves yarn-env.sh yarn-site.xml hadoop@192.168.80.92
.....
.....
```

三，启动 **hadoop**

3.1 初始化 **hadoop** （注意：请关闭所有节点防火墙）

切换至 **hadoop** 用户，进入 **/home/hadoop/hadoop-2.2.0/bin/** 下执行命令 **hdfs namenode -format** 进行初始化

```
[hadoop@Master bin]$
[hadoop@Master bin]$ pwd
/home/hadoop/hadoop-2.2.0/bin
[hadoop@Master bin]$ hdfs namenode -format
13/11/12 11:44:46 INFO namenode.NameNode: STARTUP_MSG:
/*****
STARTUP_MSG: Starting NameNode
STARTUP_MSG:   host = Master.Hadoop/192.168.80.90
STARTUP_MSG:   args = [-format]
STARTUP_MSG:   version = 2.2.0
STARTUP_MSG:   classpath = /home/hadoop/hadoop-2.2.0/etc/hadoop:/home/hadoop/hadoop-2.2.0/share/hadoop
o-2.1.jar:/home/hadoop/hadoop-2.2.0/share/hadoop/common/lib/commons-configuration-1.6.jar:/home/hadoop
oop/common/lib/jackson-mapper-asl-1.8.8.jar:/home/hadoop/hadoop-2.2.0/share/hadoop/common/lib/jersey-s
2.2.0/share/hadoop/common/lib/guava-11.0.2.jar:/home/hadoop/hadoop-2.2.0/share/hadoop/common/lib/log4j
oop-2.2.0/share/hadoop/common/lib/jetty-util-6.1.26.jar:/home/hadoop/hadoop-2.2.0/share/hadoop/common/
e/hadoop/hadoop-2.2.0/share/hadoop/common/lib/jackson-xml-1.8.8.jar:/home/hadoop/hadoop-2.2.0/share/had
.....

13/11/12 11:44:49 INFO namenode.FSNamesystem: supergroup          = supergroup
13/11/12 11:44:49 INFO namenode.FSNamesystem: isPermissionEnabled = true
13/11/12 11:44:49 INFO namenode.FSNamesystem: HA Enabled: false
13/11/12 11:44:49 INFO namenode.FSNamesystem: Append Enabled: true
13/11/12 11:44:50 INFO util.GSet: Computing capacity for map INodeMap
13/11/12 11:44:50 INFO util.GSet: VM type          = 64-bit
13/11/12 11:44:50 INFO util.GSet: 1.0% max memory = 966.7 MB
13/11/12 11:44:50 INFO util.GSet: capacity        = 2^20 = 1048576 entries
13/11/12 11:44:50 INFO namenode.NameNode: Caching file names occurring more than 10 times
13/11/12 11:44:50 INFO namenode.FSNamesystem: dfs.namenode.safemode.threshold-pct = 0.999000
13/11/12 11:44:50 INFO namenode.FSNamesystem: dfs.namenode.safemode.min.datanodes = 0
13/11/12 11:44:50 INFO namenode.FSNamesystem: dfs.namenode.safemode.extension      = 30000
13/11/12 11:44:50 INFO namenode.FSNamesystem: Retry cache on namenode is enabled
13/11/12 11:44:50 INFO namenode.FSNamesystem: Retry cache will use 0.03 of total heap and re
13/11/12 11:44:50 INFO util.GSet: Computing capacity for map Namenode Retry Cache
13/11/12 11:44:50 INFO util.GSet: VM type          = 64-bit
13/11/12 11:44:50 INFO util.GSet: 0.029999999329447746% max memory = 966.7 MB
13/11/12 11:44:50 INFO util.GSet: capacity        = 2^15 = 32768 entries
13/11/12 11:44:51 INFO common.Storage: Storage directory /home/hadoop/hadoop-2.2.0/dfs/name
13/11/12 11:44:51 INFO namenode.FSImage: Saving image file /home/hadoop/hadoop-2.2.0/dfs/nam
13/11/12 11:44:51 INFO namenode.FSImage: Image file /home/hadoop/hadoop-2.2.0/dfs/name/curre
13/11/12 11:44:51 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txi
13/11/12 11:44:51 INFO util.ExitUtil: Exiting with status 0
13/11/12 11:44:51 INFO namenode.NameNode: SHUTDOWN_MSG:
/*****
SHUTDOWN_MSG: Shutting down NameNode at Master.Hadoop/192.168.80.90
*****/
```

中间会产生很多信息，这个正常

初始化完之后会自动的在/home/hadoop/hadoop-2.2.0/创建 dfs/name/ 目录

每次初始化都会对 dfs 进行操作，一般只要初始化一次就可以了，因为如果 dfs 里面有数据的话会把它给弄丢的

3.2 启动 hadoop （用 hadoop 用户执行）（注意：请关闭所有节点防火墙）

切换至/home/hadoop/hadoop-2.2.0/sbin/下

3.2.1 执行脚本 ./start-dfs.sh

此时 master 上会产生：（两个进程）

SecondaryNameNode

NameNode

```
[hadoop@Master sbin]$ ./start-dfs.sh
Starting namenodes on [Master.Hadoop]
Master.Hadoop: starting namenode, logging to /home/hadoop/hadoop-2.2.0/logs/hadoop-hadoop-namenode-Master.Hadoop.out
192.168.80.91: starting datanode, logging to /home/hadoop/hadoop-2.2.0/logs/hadoop-hadoop-datanode-Slave1.Hadoop.out
192.168.80.92: starting datanode, logging to /home/hadoop/hadoop-2.2.0/logs/hadoop-hadoop-datanode-Slave2.Hadoop.out
Starting secondary namenodes [Master.Hadoop]
Master.Hadoop: starting secondarynamenode, logging to /home/hadoop/hadoop-2.2.0/logs/hadoop-hadoop-secondarynamenode-Master.Hadoop.out
[hadoop@Master sbin]$ jps
19419 Jps
19323 SecondaryNameNode
19138 NameNode
[hadoop@Master sbin]$
```

```
[hadoop@Master sbin]$
[hadoop@Master sbin]$ jps
12509 SecondaryNameNode
13260 Jps
12326 NameNode
[hadoop@Master sbin]$
```

Slave 节点上会产: (一个进程)

DataNode

```
[hadoop@Slave1 hadoop-2.2.0]$ jps
6770 Jps
6707 DataNode
[hadoop@Slave1 hadoop-2.2.0]$
[hadoop@Slave1 hadoop-2.2.0]$
```

3.2.2 执行脚本 ./start-yar.sh

此时 master 上会产生: (多出来一个)

ResourceManager

SecondaryNameNode

NameNode

```
[hadoop@Master sbin]$
[hadoop@Master sbin]$ ./start-yarn.sh
starting yarn daemons
starting resourcemanager, logging to /home/hadoop/hadoop-2.2.0/logs/yarn-hadoop-resourcemanager-Master.Hadoop.out
192.168.80.91: starting nodemanager, logging to /home/hadoop/hadoop-2.2.0/logs/yarn-hadoop-nodemanager-Slave1.Hadoop.out
192.168.80.92: starting nodemanager, logging to /home/hadoop/hadoop-2.2.0/logs/yarn-hadoop-nodemanager-Slave2.Hadoop.out
[hadoop@Master sbin]$ jps
19741 Jps
19498 ResourceManager
19323 SecondaryNameNode
19138 NameNode
[hadoop@Master sbin]$
```


Slave 节点上会产:

DataNode

NodeManager

```
[hadoop@Slave1 ~]$ jps
2085 DataNode
2361 Jps
2210 NodeManager
```

访问网页：<http://192.168.80.90:8088>



Cluster

About

Nodes

Applications

NEW

NEW SAVING

SUBMITTED

ACCEPTED

RUNNING

REMOVING

FINISHING

FINISHED

FAILED

KILLED

Scheduler

Tools

All Applications

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	Active Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes
0	0	0	0	0	0 B	16 GB	0 B	2	0	0	0	0

Show 20 entries

ID	User	Name	Application Type	Queue	StartTime	FinishTime	State	FinalStatus	Progress	Tracking UI
No data available in table										

Showing 0 to 0 of 0 entries

First Previous Next Last

访问网页：<http://192.168.80.90:50070>

NameNode 'Master.Hadoop:9000' (active)

Started:	Wed Nov 13 09:15:44 CST 2013
Version:	2.2.0, Unknown
Compiled:	2013-11-11T09:06Z by root from Unknown
Cluster ID:	CID-8c3fbab4-94b1-4c50-82e0-04f76e57834f
Block Pool ID:	BP-2047591509-192.168.80.90-1384246844116

[Browse the filesystem](#)
[NameNode Logs](#)

Cluster Summary

Security is OFF
1 files and directories, 0 blocks = 1 total.
Heap Memory used 22.50 MB is 57% of Committed Heap Memory 39.05 MB. Max Heap Memory is 966.69 MB.
Non Heap Memory used 30.22 MB is 95% of Committed Non Heap Memory 31.63 MB. Max Non Heap Memory is 130 MB.

Configured Capacity	:	57.67 GB			
DFS Used	:	48 KB			
Non DFS Used	:	18.28 GB			
DFS Remaining	:	39.39 GB			
DFS Used%	:	0.00%			
DFS Remaining%	:	68.30%			
Block Pool Used	:	48 KB			
Block Pool Used%	:	0.00%			
DataNodes usages	:	Min %	Median %	Max %	stdev %
	:	0.00%	0.00%	0.00%	0.00%
Live Nodes	:	2 (Decommissioned: 0)			
Dead Nodes	:	0 (Decommissioned: 0)			
Decommissioning Nodes	:	0			
Number of Under-Replicated Blocks	:	0			

NameNode Journal Status:

Current transaction ID: 6	
Journal Manager	State
FileJournalManager(root=/home/hadoop/hadoop-2.2.0/dfs/name)	EditLogFileOutputStream(/home/hadoop/hadoop-2.2.0/dfs/name/current/edits_inprogress_00000000000000000000)

NameNode Storage:

Storage Directory	Type	State
/home/hadoop/hadoop-2.2.0/dfs/name	IMAGE_AND_EDITS	Active

Startup Progress

Elapsed Time: 0sec		
Percent Complete: 100.00%		
Phase	Completion	Elapsed Time
Loading fsimage /home/hadoop/hadoop-2.2.0/dfs/name/current/fsimage_0000000000000000002 (198 B)	100.00%	0sec
inodes (1/1)	100.00%	0sec
delegation keys (0/0)	100.00%	0sec
delegation tokens (0/0)	100.00%	0sec
Loading edits	100.00%	0sec
/home/hadoop/hadoop-2.2.0/dfs/name/current/edits_0000000000000000003-0000000000000000003 (1 MB) (1/1)	100.00%	0sec
Saving checkpoint	100.00%	0sec
/home/hadoop/hadoop-2.2.0/dfs/name inodes (1/1)	100.00%	0sec
/home/hadoop/hadoop-2.2.0/dfs/name delegation keys (0/0)	100.00%	0sec
/home/hadoop/hadoop-2.2.0/dfs/name delegation tokens (0/0)	100.00%	0sec

至此，hadoop 已经启动

#####

关于 hadoop 的启动和关闭：

在/home/hadoop/hadoop-2.2.0/bin 下

hdfs 命令 用来初始化

hdfs namenode -format

在/home/hadoop/hadoop-2.2.0/sbin 下

启动

start-dfs.sh #Start hadoop dfs daemons

start-yarn.sh #Start all yarn daemons. Run this on master node

关闭

stop-dfs.sh

stop-yarn.sh

另外：

start-all.sh 脚本包含 start-dfs.sh 和 start-yarn.sh 两个脚本 所以启动时可以直接执行此脚本

stop-all.sh 关闭 同理

#####

四，常见问题

4.1 slave 节点 DataNode 无法启动

我搭建了很多次，这个问题每次都出现，所以总结了一下：

- ① 防火墙没关 解决办法：你懂得
- ② 所有节点 hosts 链表不统一，或有错误 解决办法：你懂得
- ③ 因为多次 format namenode 造成 namenode 和 datanode 的 clusterID 不一致

解决方法：

方法一：更改 slave Cluster ID 与 master 相同

修改每一个 datanode 上面的 CID(位于 dfs/data/current/VERSION 文件夹中)使两者一致

方法二：删除所有节点/home/hadoop/hadoop-2.2.0/dfs/data/下 curren 目录 然后 重新 start-dfs.sh 即可

其他问题暂时没有发现，后期再补

匆忙完稿，如有错误，敬请谅解，谢谢！

作者： 落 叶

时间：2013 年 11 月 13 日