

# Linux CentOS 下 Hadoop 伪分布模式安装笔记

## 一. 概要

经过几天的调试,终于在 Linux Cent OS 5.0 下成功搭建 Hadoop 测试环境。本次测试在一台服务器上进行伪分布式搭建。Hadoop 伪分布式模式是在单机上模拟 Hadoop 分布式,单机上的分布式并不是真正的伪分布式,而是使用线程模拟分布式。Hadoop 本身是无法区分伪分布式和分布式的,两种配置也很相似,唯一不同的地方是伪分布式是在单机器上配置,数据节点和名字节点均是一个机器。虽然 Hadoop 的安装步骤并不复杂,但是我在安装期间还是遇到了很多琐碎的问题,现将自己搭建 Hadoop 的详细过程和遇到的问题记录下来。

## 二. 环境搭建

搭建测试环境所需的软件包括: jdk1.6.0\_20、hadoop-0.20.2.tar.gz。测试服务器操作系统 Linux Cent OS 5.0。

### 1. SSH 无密码验证配置

Hadoop 需要使用 SSH 协议,namenode 将使用 SSH 协议启动 namenode 和 datanode 进程,伪分布式模式数据节点和名称节点均是本身,必须配置 SSH localhost 无密码验证。

用 root 用户登录,在家目录下执行如下命令: `ssh-keygen -t rsa`

```
[root@master ~]# ssh-keygen -t rsa
```

Generating public/private rsa key pair.

Enter file in which to save the key (/root/.ssh/id\_rsa): & 按回车默认路径 &

Created directory '/root/.ssh'. &创建/root/.ssh 目录&

Enter passphrase (empty for no passphrase):

Enter same passphrase again:

Your identification has been saved in /root/.ssh/id\_rsa.

Your public key has been saved in /root/.ssh/id\_rsa.pub.

The key fingerprint is:

c6:7e:57:59:0a:2d:85:49:23:cc:c4:58:ff:db:5b:38 root@master

通过以上命令将在/root/.ssh/ 目录下生成 id\_rsa 私钥和 id\_rsa.pub 公钥。进入/root/.ssh 目录在 namenode 节点下做如下配置:

```
[root@master .ssh]# cat id_rsa.pub > authorized_keys
```

配置完毕,可通过 ssh 本机 IP 测试是否需要密码登录。

### 2. JDK 安装及 Java 环境变量的配置

## 2.1 JDK 安装

root 用户登陆，新建文件夹 /usr/program ,下载 JDK 安装包 jdk-6u13-linux-i586.bin,复制到目录/usr/ program 下，在命令行进入该目录，执行命令“./jdk-6u20-linux-i586.bin”，命令运行完毕，将在目录下生成文件夹 jdk1.6.0\_20，安装完毕。

## 2.2 java 环境变量配置

root 用户登陆，命令行中执行命令“vi /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)。

```
# set java environment
```

```
export JAVA_HOME=/usr/program/jdk1.6.0_20
```

```
export JRE_HOME=/usr/program/jdk1.6.0_20/jre
```

```
export CLASSPATH=.:$JAVA_HOME/lib:$JAVA_HOME/jre/lib
```

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

在 vi 编辑器增加以上内容后保存退出，并执行以下命令使配置生效

```
chmod +x /etc/profile ; 增加执行权限
```

```
source /etc/profile ;
```

配置完毕后，在命令行中输入 java -version，如出现下列信息说明 java 环境安装成功。

```
java version "1.6.0_20"
```

```
Java(TM) SE Runtime Environment (build 1.6.0_20-b02)
```

```
Java HotSpot(TM) Server VM (build 16.3-b01, mixed mode)
```

## 2. Hadoop 配置

下载 hadoop-0.20.2.tar.gz，将其解压到/usr/local/hadoop 目录下，解压后目录形式是/usr/local/hadoop/hadoop-0.20.2。使用如下命令：

```
tar zxvf hadoop-0.19.1.tar.gz 进行 hadoop 压缩文件解压。
```

2.1 进入/usr/local/hadoop/hadoop-0.20.2/conf， 配置 Hadoop 配置文件

2.1.1 配置 hadoop-env.sh 文件

添加 # set java environment

```
export JAVA_HOME=/usr/program/jdk1.6.0_20
```

编辑后保存退出。

2.1.2 配置 core-site.xml

```
[root@master conf]# vi core-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>fs.default.name</name>
    <value>hdfs://202.173.253.36:9000</value>
  </property>
  <property>
    <name>hadoop.tmp.dir</name>
    <value>/usr/local/hadoop/hadooptmp</value>
  </property>
</configuration>
```

注意配置xml文件时标签一定要在一行,不然xml的设置会无效的

### 2.1.3 配置 hdfs-site.xml

```
[root@master conf]# vi 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>/usr/local/hadoop/hdfs/name</value>
  </property>
  <property>
    <name>dfs.data.dir</name>
    <value>/usr/local/hadoop/hdfs/data</value>
  </property>
  <property>
    <name>dfs.replication</name>
    <value>1</value>
  </property>
</configuration>
```

### 2.1.4 配置 mapred-site.xml

```
[root@master conf]# vi 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>202.173.253.36:9001</value>
  </property>
  <property>
    <name>mapred.local.dir</name>
    <value>/usr/local/hadoop/mapred/local</value>
  </property>
  <property>
    <name>mapred.system.dir</name>
    <value>/tmp/hadoop/mapred/system</value>
  </property>
</configuration>
```

### 2.1.5 配置 masters 文件和 slaves 文件

```
[root@master conf]# vi masters
202.173.253.36
[root@master conf]# vi slaves
202.173.253.36
```

注：因为在伪分布模式下，作为 master 的 namenode 与作为 slave 的 datanode 是同一台服务器，所以配置文件中的 ip 是一样的。

### 2.1.6 编辑主机名

```
[root@master ~]# vi /etc/hosts
# Do not remove the following line, or various programs
that require network functionality will fail.
127.0.0.1          localhost
202.173.253.36     master
202.173.253.36     slave
```

注：因为是在伪分布模式下，所以 master 与 slave 是一台机

## 2.2 Hadoop 启动

2.2.1 进入 /usr/local/hadoop/hadoop-0.20.2/bin 目录下，格式化 namenode

```
[root@master bin]# hadoop namenode -format
```

```
10/07/19 10:46:41 INFO namenode.NameNode: STARTUP_MSG:
```

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

```
STARTUP_MSG: Starting NameNode
```

```
STARTUP_MSG: host = master/202.173.253.36
```

```
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 /usr/local/hadoop/hdfs/name ? (Y or N) Y
```

```
10/07/19 10:46:43 INFO namenode.FSNamesystem:
```

```
fsOwner=root,root,bin,daemon,sys,adm,disk,wheel
```

```
10/07/19 10:46:43 INFO namenode.FSNamesystem: supergroup=supergroup
```

```
10/07/19 10:46:43 INFO namenode.FSNamesystem:
```

```
isPermissionEnabled=true
```

```
10/07/19 10:46:43 INFO common.Storage: Image file of size 94 saved in 0  
seconds.
```

```
10/07/19 10:46:43 INFO common.Storage: Storage directory
```

```
/usr/local/hadoop/hdfs/name has been successfully formatted.
```

```
10/07/19 10:46:43 INFO namenode.NameNode: SHUTDOWN_MSG:
```

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

```
SHUTDOWN_MSG: Shutting down NameNode at master/202.173.253.36
```

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

### 2.2.2 启动 hadoop 所有进程

在/usr/local/hadoop/hadoop-0.20.2/bin 目录下，执行 **start-all.sh** 命令

启动完成后，可用 **jps** 命令查看 **hadoop** 进程是否启动完全。正常情况下应该有如下进程：

```
10910 NameNode
```

```
11431 Jps
```

```
11176 SecondaryNameNode
```

如果HDFS一直没有启动,并查询日志分析得NameNode和DataNode的namespaceID不一致,则可以通过如下办法修复:  
修改每个DataNode的namespaceID(位于/dfs/data/current/VERSION文件中)或修改NameNode的namespaceID(位于/dfs/name/current/VERSION文件中),使其一致  
这个问题一般是由于两次或两次以上的格式化NameNode造成的

11053 DataNode

11254 JobTracker

11378 TaskTracker

我在搭建过程中,在此环节出现的问题最多,经常出现启动进程不完整的情况,要不是 datanode 无法正常启动,就是 namenode 或是 TaskTracker 启动异常。解决的方式如下:

- 1.在 Linux 下关闭防火墙: 使用 service iptables stop 命令;
- 2.再次对 namenode 进行格式化: 在/usr/local/hadoop/hadoop-0.20.2/bin 目录下执行 hadoop namenode -format 命令
- 3.对服务器进行重启
- 4.查看 datanode 或是 namenode 对应的日志文件, 日志文件保存在 /usr/local/hadoop/hadoop-0.20.2/logs 目录下。仔细查看日志报错的原因, (上次日志报错的信息忘记了) 解决方法是进入/usr/local/hadoop/hdfs/name 和 usr/local/hadoop/hdfs/data 目录下, 将目录下的文件全部删除。
- 5.再次在/bin 目录下用 start-all.sh 命令启动所有进程, 通过以上的几个方法应该能解决进程启动不完全的问题了。

2.2.3 查看集群状态

在 bin 目录下执行: **hadoop dfsadmin -report**

```
[root@master bin]# hadoop dfsadmin -report
Configured Capacity: 304427253760 (283.52 GB)
Present Capacity: 282767941632 (263.35 GB)
DFS Remaining: 282767904768 (263.35 GB)
DFS Used: 36864 (36 KB)
DFS Used%: 0%
Under replicated blocks: 0
Blocks with corrupt replicas: 0
Missing blocks: 0

-----

Datanodes available: 1 (1 total, 0 dead)
Name: 202.173.253.36:50010
Decommission Status : Normal
Configured Capacity: 304427253760 (283.52 GB)
DFS Used: 36864 (36 KB)
Non DFS Used: 21659312128 (20.17 GB)
```

DFS Remaining: 282767904768(263.35 GB)

DFS Used%: 0%

DFS Remaining%: 92.89%

Last contact: Mon Jul 19 11:07:22 CST 2010

### 2.3 在 WEB 页面下查看 Hadoop 工作情况

打开 IE 浏览器输入部署 Hadoop 服务器的 IP:

<http://localhost:50070>;

#### NameNode 'master:9000'

**Started:** Mon Jul 19 10:49:51 CST 2010  
**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

6 files and directories, 1 blocks = 7 total. Heap Size is 58.88 MB / 888.94 MB (6%)

Configured Capacity	283.52 GB
DFS Used	36 KB
Non DFS Used	20.17 GB
DFS Remaining	263.35 GB
DFS Used%	0 %
DFS Remaining%	92.89 %
<a href="#">Live Nodes</a>	1
<a href="#">Dead Nodes</a>	0

#### NameNode Storage:

Storage Directory	Type	State
/usr/local/hadoop/hdfs/name	IMAGE_AND_EDITS	Active

[Hadoop](#), 2010.

51CTO.com  
技术博客 Blog

<http://localhost:50030>。

## NameNode 'master:9000'

**Started:** Mon Jul 19 10:49:51 CST 2010  
**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

6 files and directories, 1 blocks = 7 total. Heap Size is 58.88 MB / 888.94 MB (6%)

Configured Capacity	283.52 GB
DFS Used	36 KB
Non DFS Used	20.17 GB
DFS Remaining	263.35 GB
DFS Used%	0 %
DFS Remaining%	92.89 %
<a href="#">Live Nodes</a>	1
<a href="#">Dead Nodes</a>	0

### NameNode Storage:

Storage Directory	Type	State
/usr/local/hadoop/hdfs/name	IMAGE_AND_EDITS	Active

[Hadoop](#), 2010.

51CTO.com  
技术博客 Blog

## 3. Hadoop 使用

一个测试例子 wordcount

计算输入文本中词语数量的程序。WordCount 在 Hadoop 主目录下的 java 程序包 `hadoop-0.20.2-examples.jar` 中，执行步骤如下：

在 `/usr/local/hadoop/hadoop-0.20.2/bin/` 目录下进行如下操作：

`hadoop fs -mkdir bxy`(新建目录名称，可任意命名)

`[root@master log]# hadoop fs -copyFromLocal secure.2 bxy`

(找一个任意文件将其 copy 到 bxy 文件夹)

在 `/usr/local/hadoop/hadoop-0.20.2` 下执行：

`[root@master hadoop-0.20.2]# hadoop jar hadoop-0.20.2-examples.jar`

`wordcount bxy output` (提交作业，此处需注意 bxy 与 output 是一组任务，下次再执行 wordcount 程序，还要新建目录 bxy1 与 output1 不能跟 bxy 与 output 重名)

执行完毕后，可进入 web 界面刷新查看 running job 及 completed job 的显示。



## Cluster Summary (Heap Size is 58.88 MB/888.94 MB)

Maps	Reduces	Total Submissions	Nodes	Map Task Capacity	Reduce Task Capacity	Avg. Tasks/Node	Blacklisted Nodes
0	0	1	1	2	2	4.00	0

## 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

Jobid	Priority	User	Name	Map % Complete	Map Total	Maps Completed	Reduce % Complete	Reduce Total	Reduces Completed	Job Scheduling Information
job_20130227191050_0001	NORMAL	root	word count	100.00%	1	1	100.00%	1	1	

## 补充实验

```
[root@master hadoop-0.20.2]# mkdir input_shen1
[root@master hadoop-0.20.2]# echo "hello world">./input_shen1/test1.txt
[root@master hadoop-0.20.2]# echo "hello hadoop">./input_shen1/test2.txt
[root@master hadoop-0.20.2]# bin/hadoop dfs -put ./input_shen1 in
[root@master hadoop-0.20.2]# bin/hadoop jar hadoop-0.20.2-examples.jar wordcount in out
13/02/27 03:00:34 INFO input.FileInputFormat: Total input paths to process : 2
13/02/27 03:00:35 INFO mapred.JobClient: Running job: job_201302270221_0002
13/02/27 03:00:36 INFO mapred.JobClient: map 0% reduce 0%
13/02/27 03:00:47 INFO mapred.JobClient: map 50% reduce 0%
13/02/27 03:00:50 INFO mapred.JobClient: map 100% reduce 0%
13/02/27 03:00:59 INFO mapred.JobClient: map 100% reduce 100%
13/02/27 03:01:01 INFO mapred.JobClient: Job complete: job_201302270221_0002
13/02/27 03:01:01 INFO mapred.JobClient: Counters: 17
13/02/27 03:01:01 INFO mapred.JobClient:   Job Counters
13/02/27 03:01:01 INFO mapred.JobClient:     Launched reduce tasks=1
13/02/27 03:01:01 INFO mapred.JobClient:     Launched map tasks=2
13/02/27 03:01:01 INFO mapred.JobClient:     Data-local map tasks=2
13/02/27 03:01:01 INFO mapred.JobClient:   FileSystemCounters
13/02/27 03:01:01 INFO mapred.JobClient:     FILE_BYTES_READ=55
13/02/27 03:01:01 INFO mapred.JobClient:     HDFS_BYTES_READ=25
13/02/27 03:01:01 INFO mapred.JobClient:     FILE_BYTES_WRITTEN=180
13/02/27 03:01:01 INFO mapred.JobClient:     HDFS_BYTES_WRITTEN=25
13/02/27 03:01:01 INFO mapred.JobClient:   Map-Reduce Framework
13/02/27 03:01:01 INFO mapred.JobClient:     Reduce input groups=3
13/02/27 03:01:01 INFO mapred.JobClient:     Combine output records=4
13/02/27 03:01:01 INFO mapred.JobClient:     Map input records=2
13/02/27 03:01:01 INFO mapred.JobClient:     Reduce shuffle bytes=61
13/02/27 03:01:01 INFO mapred.JobClient:     Reduce output records=3
13/02/27 03:01:01 INFO mapred.JobClient:     Spilled Records=8
13/02/27 03:01:01 INFO mapred.JobClient:     Map output bytes=41
13/02/27 03:01:01 INFO mapred.JobClient:     Combine input records=4
13/02/27 03:01:01 INFO mapred.JobClient:     Map output records=4
13/02/27 03:01:01 INFO mapred.JobClient:     Reduce input records=4
[root@master hadoop-0.20.2]# bin/hadoop dfs -cat ./out/*
hadoop 1
hello 2
world 1
cat: Source must be a file.
[root@master hadoop-0.20.2]#
```