Hadoop1.X 伪分布式安装

本文版权归作者和博客园共有，欢迎转载，但未经作者同意必须保留此段声明，且在文章页面明显位置给出原文连接，博主为石山园，博客地址为 http://www.cnblogs.com/shishanyuan 。
该系列课程是应邀实验楼整理编写的，这里需要赞一下实验楼提供了学习的新方式，可以边看博客边上机实验，课程地址为 https://www.shiyanlou.com/courses/237

【注】该系列所使用到安装包、测试文件和代码可以在百度网盘下载，具体地址为 http://pan.baidu.com/s/10PnDs

1 Hadoop 介绍

1.1 Hadoop 简介

Apache Hadoop 软件库是一个框架，允许在集群服务器上使用简单的编程模型对大数据集进行分布式处理。Hadoop 被设计成能够从单台服务器扩展到数以千计的服务器，每台服务器都有本地的计算和存储资源。Hadoop 的高可用性并不依赖硬件，其代码库自身就能在应用层侦测并处理硬件故障，因此能基于服务器集群提供高可用性的服务。

1.2 Hadoop 生态系统

经过多年的发展形成了 Hadoop1.X 生态系统，其结构如下图所示：

- HDFS--Hadoop 生态圈的基本组成部分是 Hadoop 分布式文件系统（HDFS）。HDFS 是一种数据分布式保存机制，数据被保存在计算机集群上，HDFS 为 HBase 等工具提供了基础。
- MapReduce--Hadoop 的主要执行框架是 MapReduce，它是一个分布式、并行处理的编

程模型，MapReduce 把任务分为 map(映射)阶段和 reduce(化简)。由于 MapReduce 工作原理的特性， Hadoop 能以并行的方式访问数据，从而实现快速访问数据。

- Hbase--HBase 是一个建立在 HDFS 之上，面向列的 NoSQL 数据库，用于快速读/写大量数据。HBase 使用 Zookeeper 进行管理，确保所有组件都正常运行。
- Zookeeper--用于 Hadoop 的分布式协调服务。Hadoop 的许多组件依赖于 Zookeeper，它运行在计算机集群上面，用于管理 Hadoop 操作。
- Pig--它是 MapReduce 编程的复杂性的抽象。Pig 平台包括运行环境和用于分析 Hadoop 数据集的脚本语言(Pig Latin)。其编译器将 Pig Latin 翻译成 MapReduce 程序序列。
- Hive--Hive 类似于 SQL 高级语言，用于运行存储在 Hadoop 上的查询语句，Hive 让不熟悉 MapReduce 开发人员也能编写数据查询语句，然后这些语句被翻译为 Hadoop 上面的 MapReduce 任务。像 Pig 一样，Hive 作为一个抽象层工具，吸引了很多熟悉 SQL 而不是 Java 编程的数据分析师。
- Sqoop 是一个连接工具，用于在关系数据库、数据仓库和 Hadoop 之间转移数据。Sqoop 利用数据库技术描述架构，进行数据的导入/导出；利用 MapReduce 实现并行化运行和容错技术。
- Flume 提供了分布式、可靠、高效的服务，用于收集、汇总大数据，并将单台计算机的大量数据转移到 HDFS。它基于一个简单而灵活的架构，并提供了数据流的流。它利用简单的可扩展的数据模型，将企业中多台计算机上的数据转移到 Hadoop

1.3 Apache 版本衍化

Apache Hadoop 版本分为两代，我们将第一代 Hadoop 称为 Hadoop 1.0，第二代 Hadoop 称为 Hadoop 2.0。第一代 Hadoop 包含三个大版本，分别是 0.20.x，0.21.x 和 0.22.x，其中，0.20.x 最后演化成 1.0.x，变成了稳定版，而 0.21.x 和 0.22.x 包含 NameNode HA 等新的重大特性。第二代 Hadoop 包含两个版本，分别是 0.23.x 和 2.x，它们完全不同于 Hadoop 1.0，是一套全新的架构，均包含 HDFS Federation 和 YARN 两个系统，相比于 0.23.x，2.x 增加了 NameNode HA 和 Wire-compatibility 两个重大特性。

Year	Release	说明	是否为稳定版（可用于生产环境）
2010	0.20.2	经典版本，第一次包含的所有特性均可用。	是
	0.21.0	包含 append，RAID 和 symlink，NameNode HA 特性，但不包含 security，	否
2011	0.20.203	包含 security，但不包含 append，RAID 或者 NameNode HA	是，yahoo 在 4,500 节点线上集群部署使用
	0.20.205 (now 1.0)	包含 append 和 security，但不包含 symlink，RAID，NameNode HA	是
	0.22.0	包含 append、HDFS security，RAID，symlinks，NameNode HA，但不包含 MapReduce security 和其他一些优化	否
	0.23.0	下一代 Hadoop，包含 HDFS Federation 和 YARN	否
2012	0.23.1~0.23.5	0.23.0 版本的 bug 修复和性能改进	否
	1.0.1~1.0.4	1.0.0 版本的 bug 修复和性能改进	是
	2.0.0-alpha~2.0.2-alpha	下一代 Hadoop，比 0.23.0 多出 NameNode HA 和 Wire-compatibility	否

来自：董的博客（dongxicheng.org）

2 Hadoop1.X 伪分布安装

Hadoop 安装有如下三种方式

- 单机模式：安装简单，几乎不用作任何配置，但仅限于调试用途；
- 伪分布模式 :在单节点上同时启动 NameNode、DataNode、JobTracker、TaskTracker、Secondary Namenode 等 5 个进程，模拟分布式运行的各个节点；
- 完全分布式模式：正常的 Hadoop 集群，由多个各司其职的节点构成

由于实验环境的限制，**本节课程将讲解伪分布模式安装**，并在随后的课程中以该环境为基础进行其他组件部署实验。以下为伪分布式环境下在 CentOS6 中配置 Hadoop-1.1.2，该配置可以作为其他 Linux 系统和其他版本的 Hadoop 部署参考。

2.1 软硬件环境说明

所有节点均是 CentOS 系统，防火墙和 SElinux 禁用，所有节点上均创建了一个 shiyanlou 用户，并在系统根目录下创建/app 目录，用于存放 Hadoop 等组件运行包。因为该目录用于安装 hadoop 等组件程序，用户对 shiyanlou 必须赋予 rwx 权限（一般做法是 root 用户在根目录下创建/app 目录，并修改该目录拥有者为 shiyanlou(chown –R shiyanlou:shiyanlou /app）。

Hadoop 搭建环境：

- 虚拟机操作系统： CentOS6.6 64 位，单核，1G 内存
- JDK：1.7.0_55 64 位
- Hadoop：1.1.2

2.2 环境搭建（实验楼虚拟机已设置，以下提供参考）

实验环境的虚拟机已经完成的安装环境的配置，其他环境部署时可以参考该章节进行环境搭建。

2.2.1 配置本地环境

该部分对服务器的配置需要在服务器本地进行配置，配置完毕后需要重启服务器确认配置是否生效，特别是远程访问服务器需要设置固定 IP 地址。

2.2.1.1 设置 IP 地址

1. 点击 System-->Preferences-->Network Connections，如下图所示：

2. 修改或重建网络连接，设置该连接为手工方式，设置如下网络信息：

IP 地址: 192.168.42.11

子网掩码： 255.255.255.0

网关： 192.168.42.1

DNS： 221.12.1.227 (需要根据所在地设置 DNS 服务器)

Domain: 221.12.33.227

注意：网关、DNS 等根据所在网络实际情况进行设置，并设置连接方式为"Available to all users"，否则通过远程连接时会在服务器重启后无法连接服务器

3. 在命令行中，使用 ifconfig 命令查看设置 IP 地址信息，如果修改 IP 不生效，需要重启机器再进行设置（如果该机器在设置后需要通过远程访问，建议重启机器，确认机器 IP 是否生效）

2.2.1.2 设置机器名

使用 *sudo vi /etc/sysconfig/network* 打开配置文件，根据实际情况设置该服务器的机器名，新机器名在重启后生效

```
NETWORKING=yes
HOSTNAME=hadoop1
```

2.2.1.3 设置 Host 映射文件

1. 设置 IP 地址与机器名的映射，设置信息如下：

sudo vi /etc/hosts

- 192.168.42.8 b393a04554e1 hadoop

```
192.168.42.8    b393a04554e1 hadoop
ff02::2 ip6-allrouters
127.0.0.1       localhost
::1     localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
```

2. 使用如下命令对网络设置进行重启

sudo /etc/init.d/network restart

3. 使用 ping 命令验证设置是否成功

2.2.2 设置操作系统环境

2.2.2.1 关闭防火墙

在 Hadoop 安装过程中需要关闭防火墙和 SElinux，否则会出现异常

1. 使用 *sudo service iptables status* 查看防火墙状态，如下所示表示 iptables 已经开启

```
[root@hadoop1 hadoop]# service iptables status
Table: filter
Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination
1    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0           state RELATED,
ESTABLISHED
2    ACCEPT     icmp --  0.0.0.0/0            0.0.0.0/0
3    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0
4    ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           state NEW tcp
dpt:22
5    REJECT     all  --  0.0.0.0/0            0.0.0.0/0           reject-with ic
mp-host-prohibited

Chain FORWARD (policy ACCEPT)
num  target     prot opt source               destination
1    REJECT     all  --  0.0.0.0/0            0.0.0.0/0           reject-with ic
mp-host-prohibited

Chain OUTPUT (policy ACCEPT)
num  target     prot opt source               destination
```

2. 以 root 用户使用如下命令关闭 iptables

chkconfig iptables off

```
[root@hadoop1 hadoop]# chkconfig iptables off
[root@hadoop1 hadoop]#
```

2.2.2.2 关闭 SElinux

1. 使用 *sudo getenforce* 命令查看是否关闭，如果显示 Enforcing 表示没有关闭

2. 修改/etc/selinux/config 文件

将 SELINUX=enforcing 改为 SELINUX=disabled，执行该命令后重启机器

```
# 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=enforcing
SELINUX=disable
# SELINUXTYPE= can take one of these two values:
#     targeted - Targeted processes are protected,
#     mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

2.2.2.3 JDK 安装及配置

1. 下载 JDK1.7 64bit 安装包

打开 JDK1.7 64bit 安装包下载链接为：

http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html

打开界面之后，先选中 Accept License Agreement，然后下载 jdk-7u55-linux-x64.tar.gz，如下图所示：

2. 创建/app 目录，把该目录的所有者修改为 shiyanlou

sudo mkdir /app

sudo chown -R shiyanlou:shiyanlou /app

```
[shiyanlou@b393a04554e1 /]$ sudo mkdir /app
[shiyanlou@b393a04554e1 /]$ sudo chown -R shiyanlou:shiyanlou /app
[shiyanlou@b393a04554e1 /]$ ll
total 72
drwxr-xr-x   2 shiyanlou shiyanlou 4096 Jun  1 06:11 app
dr-xr-xr-x   2 root      root      4096 Mar 24 14:17 bin
drwxr-xr-x   4 root      root      4096 Apr  8 04:03 dev
drwxr-xr-x  64 root      root      4096 Jun  1 03:18 etc
```

3. 创建/app/lib 目录，使用命令如下：

mkdir /app/lib

```
[shiyanlou@b393a04554e1 /]$ cd /app
[shiyanlou@b393a04554e1 app]$ ls
[shiyanlou@b393a04554e1 app]$ mkdir lib
[shiyanlou@b393a04554e1 app]$ ll
total 4
drwxrwxr-x 2 shiyanlou shiyanlou 4096 Jun  1 06:12 lib
[shiyanlou@b393a04554e1 app]$
```

4. 把下载的安装包解压并迁移到/app/lib 目录下

cd /home/shiyanlou/install-pack

tar -zxf jdk-7u55-linux-x64.tar.gz

mv jdk1.7.0_55/ /app/lib

ll /app/lib

```
[shiyanlou@b393a04554e1 ~]$ cd /home/shiyanlou/install-pack
[shiyanlou@b393a04554e1 install-pack]$ tar -zxf jdk-7u55-linux-x64.tar.gz
[shiyanlou@b393a04554e1 install-pack]$ ls
apache-maven-3.0.5-bin.tar.gz                  jdk-7u55-linux-x64.tar.gz
chukwa-0.6.0.tar.gz                            mahout-distribution-0.6.tar.gz
eclipse-jee-luna-SR1-linux-gtk-x86_64.tar.gz   MySQL-client-5.6.21-1.el6.x86_64.rpm
flume-1.5.2-bin.tar.gz                         mysql-connector-java-5.1.22-bin.jar
hadoop-1.1.2-bin.tar.gz                        MySQL-devel-5.6.21-1.el6.x86_64.rpm
hadoop-2.2.0.tar.gz                            MySQL-server-5.6.21-1.el6.x86_64.rpm
hadoop-eclipse-plugin-1.1.2.jar                pig-0.13.0.tar.gz
hbase-0.96.2-hadoop1-bin.tar.gz                protobuf-2.5.0.tar.gz
hive-0.13.0-bin.tar.gz                         sqoop-1.4.5.bin__hadoop-1.0.0.tar.gz
jdk1.7.0_55
[shiyanlou@b393a04554e1 install-pack]$ mv jdk1.7.0_55/ /app/lib/
[shiyanlou@b393a04554e1 install-pack]$ ll /app/lib
total 4
drwxr-xr-x 8 shiyanlou shiyanlou 4096 Mar 18  2014 jdk1.7.0_55
[shiyanlou@b393a04554e1 install-pack]$
```

5. 使用 *sudo vi /etc/profile* 命令打开配置文件，设置 JDK 路径

export JAVA_HOME=/app/lib/jdk1.7.0_55

export PATH=$JAVA_HOME/bin:$PATH

export CLASSPATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar

```
unset i
unset -f pathmunge


export JAVA_HOME=/app/lib/jdk1.7.0_55
export PATH=$JAVA_HOME/bin:$PATH
export CLASSPATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar
```

6. 编译并验证

source /etc/profile

java -version

```
[shiyanlou@b393a04554e1 ~]$ source /etc/profile
[shiyanlou@b393a04554e1 ~]$ java -version
java version "1.7.0_55"
Java(TM) SE Runtime Environment (build 1.7.0_55-b13)
Java HotSpot(TM) 64-Bit Server VM (build 24.55-b03, mixed mode)
[shiyanlou@b393a04554e1 ~]$
```

2.2.2.4 更新 OpenSSL

CentOS 自带的 OpenSSL 存在 bug，如果不更新 OpenSSL 在 Ambari 部署过程会出现无法通过 SSH 连接节点，使用如下命令进行更新：

yum update openssl

```
[root@hadoop1 hadoop]# yum update openssl
Loaded plugins: fastestmirror, refresh-packagekit, security
Determining fastest mirrors
 * base: centos.ustc.edu.cn
 * extras: centos.ustc.edu.cn
 * updates: centos.ustc.edu.cn
base                                                     | 3.7 kB     00:00
base/primary_db                                          | 4.4 MB     00:04
extras                                                   | 3.4 kB     00:00
extras/primary_db                                        |  19 kB     00:00
updates                                                  | 3.4 kB     00:00
updates/primary_db                                       | 3.1 MB     00:02
Setting up Update Process
Resolving Dependencies
--> Running transaction check
---> Package openssl.x86_64 0:1.0.1e-15.el6 will be updated
---> Package openssl.x86_64 0:1.0.1e-16.el6_5.7 will be an update
--> Finished Dependency Resolution
```

```
 Package: centos-release-6-5.el6.centos.11.1.x86_64 (@anaconda-CentOS-20131127
149.x86_64/6.5)
 From   : /etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-6
Is this ok [y/N]: y
Running rpm_check_debug
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Updating   : openssl-1.0.1e-16.el6_5.7.x86_64                            1/2
  Cleanup    : openssl-1.0.1e-15.el6.x86_64                                2/2
  Verifying  : openssl-1.0.1e-16.el6_5.7.x86_64                            1/2
  Verifying  : openssl-1.0.1e-15.el6.x86_64                                2/2

Updated:
  openssl.x86_64 0:1.0.1e-16.el6_5.7

Complete!
```

2.2.2.5 SSH 无密码验证配置

1. 使用 *sudo vi /etc/ssh/sshd_config* 打开 sshd_config 配置文件，开放三个配置，如下图所示：

RSAAuthentication yes

PubkeyAuthentication yes

AuthorizedKeysFile .ssh/authorized_keys

```
#MaxSessions 10

RSAAuthentication yes
PubkeyAuthentication yes
AuthorizedKeysFile      .ssh/authorized_keys
#AuthorizedKeysCommand none
#AuthorizedKeysCommandRunAs nobody

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#RhostsRSAAuthentication no
# similar for protocol version 2
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# RhostsRSAAuthentication and HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# To disable tunneled clear text passwords, change to no here!
#PasswordAuthentication yes
#PermitEmptyPasswords no
PasswordAuthentication yes
```

2. 配置后重启服务

sudo service sshd restart

```
[root@hadoop1 hadoop]#
[root@hadoop1 hadoop]# service sshd restart
Stopping sshd:                                             [  OK  ]
Starting sshd:                                             [  OK  ]
[root@hadoop1 hadoop]#
```

3. 使用 shiyanlou 用户登录使用如下命令生成私钥和公钥；

ssh-keygen -t rsa

```
[hadoop@hadoop1 .ssh]$
[hadoop@hadoop1 .ssh]$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/hadoop/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/hadoop/.ssh/id_rsa.
Your public key has been saved in /home/hadoop/.ssh/id_rsa.pub.
The key fingerprint is:
99:6b:89:e2:3f:ec:a8:98:28:bd:a1:4a:44:dd:47:b1 hadoop@hadoop1
The key's randomart image is:
+--[ RSA 2048]----+
|        o.        |
|  . . . .         |
| . . . E          |
|.     .  o        |
| .      S         |
|.      . o        |
| o. ... +         |
|++.o oo.          |
|o ooooo.          |
+-----------------+
[hadoop@hadoop1 .ssh]$ ls
id_rsa  id_rsa.pub
[hadoop@hadoop1 .ssh]$
```

4. 进入/home/shiyanlou/.ssh 目录把公钥命名为 *authorized_keys*，使用命令如下：

cp id_rsa.pub authorized_keys

```
[hadoop@hadoop1 .ssh]$
[hadoop@hadoop1 .ssh]$ cp id_rsa.pub authorized_keys_hadoop1
[hadoop@hadoop1 .ssh]$ ls
authorized_keys_hadoop1  id_rsa  id_rsa.pub
[hadoop@hadoop1 .ssh]$
```

5. 使用如下设置 authorized_keys 读写权限

chmod 400 authorized_keys

```
[hadoop@hadoop1 .ssh]$
[hadoop@hadoop1 .ssh]$ chmod 400 authorized_keys
[hadoop@hadoop1 .ssh]$
```

6. 测试 ssh 免密码登录是否生效

2.3 Hadoop 变量配置

2.3.1 解压并移动 hadoop 安装包

在 Apache 的归档目录中下载 hadoop-1.1.2-bin.tar.gz 安装包，也可以在 /home/shiyanlou/install-pack 目录中找到该安装包，解压该安装包并把该安装包复制到/app 目录中

cd /home/shiyanlou/install-pack

tar -xzf hadoop-1.1.2-bin.tar.gz

mv hadoop-1.1.2 /app

```
[shiyanlou@b393a04554e1 ~]$ cd /home/shiyanlou/install-pack
[shiyanlou@b393a04554e1 install-pack]$ tar -xzf hadoop-1.1.2-bin.tar.gz
[shiyanlou@b393a04554e1 install-pack]$ ls
apache-maven-3.0.5-bin.tar.gz                    jdk-7u55-linux-x64.tar.gz
chukwa-0.6.0.tar.gz                              mahout-distribution-0.6.tar.gz
eclipse-jee-luna-SR1-linux-gtk-x86_64.tar.gz     MySQL-client-5.6.21-1.el6.x86_64.rpm
flume-1.5.2-bin.tar.gz                           mysql-connector-java-5.1.22-bin.jar
hadoop-1.1.2                                     MySQL-devel-5.6.21-1.el6.x86_64.rpm
hadoop-1.1.2-bin.tar.gz                          MySQL-server-5.6.21-1.el6.x86_64.rpm
hadoop-2.2.0.tar.gz                              pig-0.13.0.tar.gz
hadoop-eclipse-plugin-1.1.2.jar                  protobuf-2.5.0.tar.gz
hbase-0.96.2-hadoop2-bin.tar.gz                  sqoop-1.4.5.bin__hadoop-1.0.0.tar.gz
hive-0.13.0-bin.tar.gz
[shiyanlou@b393a04554e1 install-pack]$ mv hadoop-1.1.2 /app/
[shiyanlou@b393a04554e1 install-pack]$ ls /app
hadoop-1.1.2  lib
[shiyanlou@b393a04554e1 install-pack]$
```

2.3.2 在 hadoop-1.1.2 目录下创建子目录

cd /app/hadoop-1.1.2

mkdir tmp

mkdir hdfs

mkdir hdfs/name

mkdir hdfs/data

ls

```
[shiyanlou@b393a04554e1 ~]$ cd /app/hadoop-1.1.2
[shiyanlou@b393a04554e1 hadoop-1.1.2]$ mkdir tmp
[shiyanlou@b393a04554e1 hadoop-1.1.2]$ mkdir hdfs
[shiyanlou@b393a04554e1 hadoop-1.1.2]$ mkdir hdfs/name
[shiyanlou@b393a04554e1 hadoop-1.1.2]$ mkdir hdfs/data
[shiyanlou@b393a04554e1 hadoop-1.1.2]$ ls
bin                     hadoop-client-1.1.2.jar        ivy          sbin
build.xml               hadoop-core-1.1.2.jar          ivy.xml      share
c++                     hadoop-examples-1.1.2.jar      lib          src
CHANGES.txt             hadoop-minicluster-1.1.2.jar   libexec      tmp
conf                    hadoop-test-1.1.2.jar          LICENSE.txt  webapps
contrib                 hadoop-tools-1.1.2.jar         NOTICE.txt
hadoop-ant-1.1.2.jar    hdfs                           README.txt
[shiyanlou@b393a04554e1 hadoop-1.1.2]$ ls hdfs/
data  name
[shiyanlou@b393a04554e1 hadoop-1.1.2]$
```

把 hdfs/data 设置为 755，否则 DataNode 会启动失败

```
[shiyanlou@b393a04554e1 hdfs]$ chmod -R 755 data
[shiyanlou@b393a04554e1 hdfs]$ ll
total 8
drwxr-xr-x 2 shiyanlou shiyanlou 4096 Jun  1 07:46 data
drwxrwxr-x 5 shiyanlou shiyanlou 4096 Jun  1 08:08 name
```

2.3.3 配置 hadoop-env.sh

1. 进入 hadoop-1.1.2/conf 目录，打开配置文件 hadoop-env.sh

cd /app/hadoop-1.1.2/conf

vi hadoop-env.sh

```
[shiyanlou@b393a04554e1 ~]$ cd /app/hadoop-1.1.2/conf
[shiyanlou@b393a04554e1 conf]$ vi hadoop-env.sh
```

2. 加入配置内容，设置了 hadoop 中 jdk 和 hadoop/bin 路径

export JAVA_HOME=/app/lib/jdk1.7.0_55

export PATH=$PATH:/app/hadoop-1.1.2/bin

```
# The scheduling priority for daemon processes.  See 'man nice'.
# export HADOOP_NICENESS=10

export JAVA_HOME=/app/lib/jdk1.7.0_55
export PATH=$PATH:/app/hadoop-1.1.2/bin
```

3. 编译配置文件 hadoop-env.sh 并确认生效

source hadoop-env.sh

hadoop version

```
[shiyanlou@b393a04554e1 conf]$
[shiyanlou@b393a04554e1 conf]$ source hadoop-env.sh
[shiyanlou@b393a04554e1 conf]$ hadoop version
Hadoop 1.1.2
Subversion https://svn.apache.org/repos/asf/hadoop/common/branches/branch-1.1 -r 14
40782
Compiled by hortonfo on Thu Jan 31 02:03:24 UTC 2013
From source with checksum c720ddcf4b926991de7467d253a79b8b
```

2.3.4 配置 core-site.xml

1. 使用如下命令打开 core-site.xml 配置文件

```
cd /app/hadoop-1.1.2/conf
sudo vi core-site.xml
```

2. 在配置文件中，按照如下内容进行配置

```
<configuration>
  <property>
    <name>fs.default.name</name>
     <value>hdfs://hadoop:9000</value>
  </property>
  <property>
    <name>hadoop.tmp.dir</name>
     <value>/app/hadoop-1.1.2/tmp</value>
  </property>
</configuration>
```

```
<!-- Put site-specific property overrides in this file. -->

<configuration>
  <property>
    <name>fs.default.name</name>
     <value>hdfs://hadoop:9000</value>
  </property>
  <property>
    <name>hadoop.tmp.dir</name>
     <value>/app/hadoop-1.1.2/tmp</value>
  </property>
</configuration>
```

2.3.5 配置 hdfs-site.xml

1. 使用如下命令打开 hdfs-site.xml 配置文件

```
cd /app/hadoop-1.1.2/conf
sudo vi hdfs-site.xml
```

2. 在配置文件中，按照如下内容进行配置

```
<configuration>
  <property>
    <name>dfs.replication</name>
     <value>1</value>
  </property>
  <property>
    <name>dfs.name.dir</name>
     <value>/app/hadoop-1.1.2/hdfs/name</value>
```

```
  </property>
  <property>
    <name>dfs.data.dir</name>
     <value>/app/hadoop-1.1.2/hdfs/data</value>
  </property>
</configuration>
```

```
<!-- Put site-specific property overrides in this file. -->

<configuration>
  <property>
    <name>dfs.replication</name>
     <value>1</value>
  </property>
  <property>
    <name>dfs.name.dir</name>
     <value>/app/hadoop-1.1.2/hdfs/name</value>
  </property>
  <property>
    <name>dfs.data.dir</name>
     <value>/app/hadoop-1.1.2/hdfs/data</value>
  </property>
</configuration>
```

2.3.6 配置 mapred-site.xml

1. 使用如下命令打开 mapred-site.xml 配置文件

```
cd /app/hadoop-1.1.2/conf
sudo vi mapred-site.xml
```

2. 在配置文件中，按照如下内容进行配置

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

```
<!-- Put site-specific property overrides in this file. -->

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

2.3.7 配置 masters 和 slaves 文件

1. 设子主节点

cd /app/hadoop-1.1.2/conf

vi masters

设置 hadoop 机器为 master 节点

2. 设置从节点

cd /app/hadoop-1.1.2/conf

vi slaves

设置 hadoop 机器为 slaves 节点，即本身即是 master 也是 slaves 节点

2.3.8 格式化 namenode

在 hadoop 机器上使用如下命令进行格式化 namenode

cd /app/hadoop-1.1.2/bin

./hadoop namenode -format

```
[shiyanlou@b393a04554e1 ~]$ cd /app/hadoop-1.1.2/bin
[shiyanlou@b393a04554e1 bin]$ ls
hadoop              start-all.sh                stop-balancer.sh
hadoop-config.sh    start-balancer.sh           stop-dfs.sh
hadoop-daemon.sh    start-dfs.sh                stop-jobhistoryserver.sh
hadoop-daemons.sh   start-jobhistoryserver.sh   stop-mapred.sh
rcc                 start-mapred.sh             task-controller
slaves.sh           stop-all.sh
[shiyanlou@b393a04554e1 bin]$ ./hadoop namenode -format
15/06/01 08:03:47 INFO namenode.NameNode: STARTUP_MSG:
/************************************************************
STARTUP_MSG: Starting NameNode
STARTUP_MSG:   host = b393a04554e1/192.168.42.8
STARTUP_MSG:   args = [-format]
STARTUP_MSG:   version = 1.1.2
STARTUP_MSG:   build = https://svn.apache.org/repos/asf/hadoop/common/branches/bran
ch-1.1 -r 1440782; compiled by 'hortonfo' on Thu Jan 31 02:03:24 UTC 2013
************************************************************/
Re-format filesystem in /app/hadoop-1.1.2/hdfs/name ? (Y or N) Y
15/06/01 08:03:51 INFO util.GSet: VM type       = 64-bit
15/06/01 08:03:51 INFO util.GSet: 2% max memory = 17.78 MB
15/06/01 08:03:51 INFO util.GSet: capacity      = 2^21 = 2097152 entries
15/06/01 08:03:51 INFO util.GSet: recommended=2097152, actual=2097152
15/06/01 08:03:51 INFO namenode.FSNamesystem: fsOwner=shiyanlou
15/06/01 08:03:51 INFO namenode.FSNamesystem: supergroup=supergroup
15/06/01 08:03:51 INFO namenode.FSNamesystem: isPermissionEnabled=true
15/06/01 08:03:51 INFO namenode.FSNamesystem: dfs.block.invalidate.limit=100
15/06/01 08:03:51 INFO namenode.FSNamesystem: isAccessTokenEnabled=false accessKeyU
pdateInterval=0 min(s), accessTokenLifetime=0 min(s)
15/06/01 08:03:51 INFO namenode.NameNode: Caching file names occuring more than 10
times
15/06/01 08:03:52 INFO common.Storage: Image file of size 115 saved in 0 seconds.
15/06/01 08:03:52 INFO namenode.FSEditLog: closing edit log: position=4, editlog=/a
pp/hadoop-1.1.2/hdfs/name/current/edits
15/06/01 08:03:52 INFO namenode.FSEditLog: close success: truncate to 4, editlog=/a
pp/hadoop-1.1.2/hdfs/name/current/edits
15/06/01 08:03:52 INFO common.Storage: Storage directory /app/hadoop-1.1.2/hdfs/nam
e has been successfully formatted.
15/06/01 08:03:52 INFO namenode.NameNode: SHUTDOWN_MSG:
/************************************************************
SHUTDOWN_MSG: Shutting down NameNode at b393a04554e1/192.168.42.8
************************************************************/
```

2.3.9 启动 hadoop

cd /app/hadoop-1.1.2/bin

./start-all.sh

```
[shiyanlou@b393a04554e1 ~]$ cd /app/hadoop-1.1.2/bin
[shiyanlou@b393a04554e1 bin]$ ./start-all.sh
starting namenode, logging to /app/hadoop-1.1.2/libexec/../logs/hadoop-shiyanlou-namenode-b393a04554e1.out
hadoop: starting datanode, logging to /app/hadoop-1.1.2/libexec/../logs/hadoop-shiyanlou-datanode-b393a04554e1.out
hadoop: starting secondarynamenode, logging to /app/hadoop-1.1.2/libexec/../logs/hadoop-shiyanlou-secondarynamenode-b393a
04554e1.out
starting jobtracker, logging to /app/hadoop-1.1.2/libexec/../logs/hadoop-shiyanlou-jobtracker-b393a04554e1.out
hadoop: starting tasktracker, logging to /app/hadoop-1.1.2/libexec/../logs/hadoop-shiyanlou-tasktracker-b393a04554e1.out
[shiyanlou@b393a04554e1 bin]$
```

2.3.10 用 jps 检验各后台进程是否成功启动

使用 jps 命令查看 hadoop 相关进程是否启动

```
[shiyanlou@b393a04554e1 bin]$ jps
2235 TaskTracker
2016 SecondaryNameNode
2372 Jps
1782 NameNode
2120 JobTracker
[shiyanlou@b393a04554e1 bin]$
```

这时我们发现少了一个 DataNode 进程，到$HADOOP_HOME/logs 目下，使用 cat hadoop-shiyanlou-datanode-5****.log（***表示所在机器名）查看日志文件，可以看到在日

志中提示：Invalid directory in dfs.data.dir:Incorrect permission for /app/hadoop-1.1.2/hdfs/data, expected:rwxr-xr-x, while actual: rwxrwxr-x

```
2015-06-05 02:59:16,680 WARN org.apache.hadoop.hdfs.server.datanode.DataNode:
Invalid directory in dfs.data.dir: Incorrect permission for /app/hadoop-1.1.2/
hdfs/data, expected: rwxr-xr-x, while actual: rwxrwxr-x
```

sudo chmod 755 /app/hadoop-1.1.2/hdfs/data

重新启动 hadoop 集群，可以看到 DataNode 进程