

Setting Up a Hadoop Cluster *

Shi Lei

shilei@comp.nus.edu.sg

June 21, 2009

Recently I have set up a hadoop cluster with up to five nodes. Here I would like to share my personal experience on configuring the cluster. This article is aimed to provide assistance to those who are new to hadoop. Please feel free to contact me if you have any corrections and/or suggestions.

Michael G. Noll has written tutorials on setting up hadoop on Ubuntu Linux in single-node cluster and multi-node cluster. These are very detailed instructions to guide you start your experience on hadoop. According to my personal experience, I find Ubuntu not very suitable to deploy hadoop, because I have met with some problems when configuring hadoop in Ubuntu, while in other Linux distribution there are no such problems. I have deployed my hadoop cluster (hadoop core 0.20.0) in CentOS Linux.

1 Prerequisites

Configure JDK:

Sun Java JDK is compulsory to run hadoop, therefore all the nodes in hadoop cluster should have JDK configured. To install and configure JDK, please refer here.

Download hadoop package:

You can download hadoop package here.

Notes:

In a multi-node hadoop cluster, the master node uses Secure Shell (SSH) commands to manipulate the remote nodes. This requires all the nodes must have the same version of JDK and hadoop core. If the versions among nodes are different, errors will occur when you start the cluster.

2 Firewall Configuration

Hadoop requires certain ports on each nodes accessible via the network. However, the default firewall *iptables* prohibit these ports being accessed. To run hadoop applications, you must make sure that these ports are open. To check the status of *iptables*, you can use these commands under root privilege:

*This article was submitted to Singapore SIG on Hadoop.

/etc/init.d/iptables status

You can simply turn *iptables* off, or at least open these ports:

9000, 9001, 50010, 50020, 50030, 50060, 50070, 50075, 50090

3 Create Dedicated User and Group

Hadoop requires all the nodes in the cluster have exactly the same structure of directory in which hadoop was installed. It will be beneficial if we create a dedicated user (e.g. “hadoop”) and install hadoop in its home folder. In the following section I will describe how to create a dedicated user “hadoop” and the group it belongs to. You must have root privilege on each nodes to carry on the following steps. To change to “root”, type in “su -” in the terminal and input the password for “root”.

Create group “hadoop_user”:

```
groupadd hadoop_user
```

Create user “hadoop”:

```
useradd -g hadoop_user -s /bin/bash -d /home/hadoop hadoop
```

in which -g specifies user “hadoop” belongs to group “hadoop_user”, -s specifies the shell to use, -d specifies the home folder for user “hadoop”.

Set password for user “hadoop”:

```
passwd hadoop
```

Then type in the password for user “hadoop” twice. Then type in “su - hadoop” to change to user “hadoop”.

4 Establish Authentication among Nodes

By default, if a user from $Node_A$ wants to login to a remote $Node_B$ by using SSH, he will be asked the password for $Node_B$ for authentication. However, it is impossible to input the authentication password every time the masternode wants to operate on a slavenode. Under this circumstance, we must adopt public key authentication. Simply speaking, every node will generate a pair of public key and private key, and $Node_A$ can login to $Node_B$ without password authentication only if $Node_B$ has a copy of $Node_A$'s public key. In other words, if $Node_B$ has $Node_A$'s public key, $Node_A$ is a trusted node to $Node_B$. In hadoop cluster, all the slavenodes must have a copy of masternode's public key. In the following section we will discuss how to generate the keys and how to make masternode authenticated.

Generate keys:

Login to each node with the account “hadoop” and run the following command:

```
ssh-keygen -t rsa
```

This command is used to generate the pair of public and private keys. “-t” specifies the type of keys, here we use RSA algorithm. When questions are asked, simply press “enter” to continue. Then two files “id_rsa” and “id_rsa.pub” are created under the folder /home/hadoop/.ssh/

Establish authentications:

Now we can copy the public key of masternode to all the slavenodes. Login to masternode with account “hadoop” and run the following command:

```
cat /home/hadoop/.ssh/id_rsa.pub >> /home/hadoop/.ssh/authorized_keys
```

```
scp /home/hadoop/.ssh/id_rsa.pub ip_address_of_slavenode_i:/home/hadoop/.ssh/master.pub
```

The second command should be executed several times until the public key is copied to all the slavenodes. Please note that “ip_address_of_slavenode_i” can be replaced with the domain name of *slavenode_i*.

Then we can login to each slavenode with account “hadoop” and run the following command:

```
cat /home/hadoop/.ssh/master.pub >> /home/hadoop/.ssh/authorized_keys
```

Then login back to masternode with account “hadoop”, and run

```
ssh ip_address_of_slavenode_i 1
```

to test whether masternode can login to slavenodes without password authentication.

5 Create Hadoop Folder

After the previous steps, now we can start to install hadoop on each node. In the following section I will use the latest hadoop release 0.20.0 as an example. We will start from the masternode.

Please note that hadoop require its installation directory be exactly the same among all the nodes. Let’s install hadoop in this folder: /home/hadoop/project/

First we need to create the folder “project”:

```
mkdir -p /home/hadoop/project
```

Then put hadoop package that we previously downloaded to this folder. After this, we go into this folder and decompress the package:

```
cd /home/hadoop/project  
tar -xzf ./hadoop-0.20.0.tar.gz
```

¹If *Node_A* connects *Node_B* for the first time, a question will be prompted to ask whether to continue connecting. In this case, just type in “yes” to add the node as a known host.

6 Hadoop Configuration

Initial configuration must be done before starting hadoop clusters. The configuration includes setting environment variables and configuring hadoop parameters.

Setting environment variables:

Some system variables and paths should be configured. Modify “/home/hadoop/.bash_profile” or “/home/hadoop/.profile” (whichever exists), and add the following lines:

```
export JAVA_HOME=PATH_TO_JDK_INSTALLATION
export HADOOP_HOME=/home/hadoop/project/hadoop-0.20.0
export PATH=$JAVA_HOME/bin:$HADOOP_HOME/bin:$PATH
```

Then we should modify “hadoop-env.sh” in *HADOOP_HOME/conf/*. After line 8

```
“# The java implementation to use. Required.”
```

in the next line delete the beginning “#” and fill in the current *JAVA_HOME*

Configuring hadoop parameters:

The previous versions of hadoop have only one configuration file “hadoop-site.xml” in the folder *HADOOP_HOME/conf/*. Release 0.20.0 has separated this configuration file into three different files: “core-site.xml”, “hdfs-site.xml”, and “mapred-site.xml”. All the parameters can be found in

```
HADOOP_HOME/src/core/core-default.xml,
HADOOP_HOME/src/hdfs/hdfs-default.xml,
HADOOP_HOME/src/mapred/mapred-default.xml.
```

Please note that **-default.xml* is the default setting. If user wants to change the parameters, it is better to modify **-site.xml* to override the default settings, rather than modify **-default.xml* directly.

Here I posted a link to the sample configuration for the three *xml* files. You can modify or add new features according to your cluster.

```
core-site.xml
hdfs-site.xml
mapred-site.xml
```

There are two more files need to be configured: *HADOOP_HOME/conf/masters* and *HADOOP_HOME/conf/slaves*. In *masters* we fill in the ip address or domain name of the master-node, and put the list of slavenodes in *slaves*.

7 Remote Copy Hadoop Folder to SlaveNodes

Now that we have configured hadoop on the master node, we can use remote copy command to replicate the hadoop folder to all the slavenodes.

```
scp -r /home/hadoop/project ip_address_of_slavenode_i:/home/hadoop/
```

Moreover, you need to set environment variables on each slavenode, as the first part of Step 6 shows.

8 Start Hadoop

Hooray! Now the hadoop cluster is established! What we need to do now is to format the namenode and start the cluster.

To format the namenode is simple. Login to the masternode with account “hadoop”, run this command:

```
hadoop namenode -format
```

A message will be displayed to report the success of formatting. Then we can start the cluster:

```
start-all.sh
```

Alternatively, you can choose to start the file system only by using *start-dfs.sh*, or start map-reduce job by *start-mapred.sh*. To stop the cluster, use command *stop-all.sh*

If there is no mistakes in the previous installation and configuration, we should find no errors or exceptions in the log files in *HADOOP_HOME/logs/*. We can use the web browser to get more information of the hadoop cluster. Here are some useful links:

Hadoop Distributed File System (HDFS):

```
http://ip_address_or_domain_name_of_namenode:50070
```

Hadoop Jobtracker:

```
http://ip_address_or_domain_name_of_jobtracker:50030
```

Hadoop Tasktracker:

```
http://ip_address_or_domain_name_of_map_reduce_processor:50060
```

9 Deal With Errors

If these web pages cannot be displayed, there must be some errors in the installation and configuration. Please examine the logs on masternode and slavenodes carefully to figure out where the problem locates.

10 Lessons that I learned

1. Solaris is not a suitable operating system for hadoop deployment.

My first hadoop cluster was deployed on Solaris. Though Solaris and Linux share a lot of features in common, difference still exists. There were quite a number of errors and exceptions when I started my Solaris hadoop cluster, making it impossible to run any hadoop applications.

2. Ubuntu Linux is not suitable as hadoop masternode.

When I use the nodes running on Ubuntu Linux as hadoop masternode, I always encounter with IPC binding error that prevent the namenode from starting. I have also tried different Ubuntu distributions (8.04, 8.10, 9.04), and such problem still exists. When

I use a node running on CentOS Linux, the namenode can be started successfully. So far I have not figured out what caused the IPC binding error. If you have successfully configured hadoop cluster under Ubuntu Linux, I will appreciate if you share your experience with me.