Hadoop Introduction - Part I

Goal: In this tutorial you will set up a single-node Hadoop cluster.

Reserve a virtual machine from VCL

- Log into NCSU Virtual Computing Lab (VCL) using your Unity ID.
- Click on "**Make a Reservation**" under "**Reservation System**" on the left side of the page.
- On the "**New Reservation**" page, select the environment "**CentOS 5.9 Base (64 bit VM)**" and the desired start time and duration. Click on the "**Create Reservation**" button to start your reservation.
- When your reservation is ready, you will see a "**Connect**" button. Click on it to see the virtual machine information. Use a SSH client (e.g., Putty) to connect to the virtual machine.

Download and install JDK

- The Java Development Kit (JDK) is required to run Hadoop. Go to the JDK download page, select JDK 7, and download the executable version for Linux 64 bit (e.g., "**jdk-7u25-linux-x64.gz**").
- Transfer the JDK installation file to the home directory on your VCL machine using `scp` or client-side tool WinSCP.
- Create a directory named "**java**" in your home directory, and copy the JDK installation file to it.

```
[user@vcl-host ~]$ mkdir java
[user@vcl-host ~]$ cp jdk-7u25-linux-x64.gz java/
```

- Unpack and install the JDK.

```
[user@vcl-host ~]$ cd java
[user@vcl-host java]$ tar zxvf jdk-7u25-linux-x64.gz
[user@vcl-host java]$ cd ..
```

- Add the JDK bin path to your environment variable `$PATH`. If you are using bash shell (to see your shell, use command `echo $SHELL`), add the following lines to the end of your "**.bashrc**" file (to see your "**.bashrc**" file, use command `ls -all` in your home directory):

```
export JAVA_HOME=$HOME/java/jdk1.7.0_25
export PATH=$JAVA_HOME/bin:$PATH
```

If you are using C shell (e.g., tcsh), add the following lines to the end of your "**.cshrc**" file:

```
setenv JAVA_HOME $HOME/java/jdk1.7.0_25
setenv PATH $JAVA_HOME/bin:$PATH
```

If you are using an EOS VCL machine (e.g., "**Linux Lab Machine**"), it is not recommended to modify your "**.cshrc**" file directly. Instead, add these lines to a new file called "**.mycshrc**" created in your home directory.

- Reconnect to your virtual machine and verify the JDK installation and version.

```
[user@vcl-host ~]$ which java
~/java/jdk1.7.0_25/bin/java
[user@vcl-host ~]$ java -version
java version "1.7.0_25"
Java(TM) SE Runtime Environment (build 1.7.0_25-b15)
Java HotSpot(TM) 64-Bit Server VM (build 23.25-b01, mixed mode
```

Download and install Hadoop

- Select a mirror from Apache Hadoop download page.
- Open directory "**hadoop-1.2.1**", download file "**hadoop-1.2.1.tar.gz**," and transfer it to the home directory on the VCL machine.
- Create a directory "**hadoop**" in your home directory, and copy the Hadoop archive file to it. Extract the file using tar command.

```
[user@vcl-host ~]$ mkdir hadoop
[user@vcl-host ~]$ cp hadoop-1.2.1.tar.gz hadoop/
[user@vcl-host ~]$ cd hadoop
[user@vcl-host hadoop]$ tar xvf hadoop-1.2.1.tar.gz
[user@vcl-host hadoop]$ cd ..
```

Configure a single-node Hadoop cluster

- Hadoop requires SSH access to manage its nodes (i.e. local machine and remote machines). For a single-node setup of Hadoop, you need to configure SSH access for your Unity user. First, generate an SSH key for your Unity user. You will create an RSA key pair with an empty password. Generally, using an empty password is not recommended, but in this case it is necessary to unlock the key without entering the password every time Hadoop interacts with its nodes.

```
[user@vcl-host ~]$ ssh-keygen -t rsa -P ""
Generating public/private rsa key pair.
Enter file in which to save the key (/home/user/.ssh/id_rsa):
Created directory '/home/user/.ssh'.Your identification has been
saved in /home/user/.ssh/id_rsa.
```

```
Your public key has been saved in /home/user/.ssh/id_rsa.pub.
The key fingerprint is:
[…]
```

- Enable SSH access with this newly created SSH key.

```
[user@vcl-host ~]$ cat $HOME/.ssh/id_rsa.pub >>
$HOME/.ssh/authorized_keys
```

- Test the SSH setup by connecting with your Unity user. The step is also needed to save your host key fingerprint to your Unity user's "**known_hosts**" file. Since the VCL machine does not support the command "`ssh localhost`," test with "`ssh 152.xxx.xxx.xxx`," where `152.xxx.xxx.xxx` is the IP address of your VCL machine.

```
[user@vcl-host ~]$ ssh 152.xxx.xxx.xxx
The authenticity of host '152.xxx.xxx.xxx (152.xxx.xxx.xxx)'
can't be established.
RSA key fingerprint is
[…]
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '152.xxx.xxx.xxx' (RSA) to the list
of known hosts.
```

- The next step is to configure Hadoop. The only required environment variable we have to configure for Hadoop in this tutorial is `JAVA_HOME`. Open file "**conf/hadoop-env.sh**" in your Hadoop directory ("**hadoop/hadoop-1.2.1**"). Set the `JAVA_HOME` environment variable to the JDK directory, by changing from:

```
# The java implementation to use. Required
# export JAVA_HOME=/usr/lib/j2sdk1.5-sun
```

to (for bash shell):

```
# The java implementation to use. Required
export JAVA_HOME=$HOME/java/jdk1.7.0_25
```

or to (for C shell):

```
# The java implementation to use. Required
setenv JAVA_HOME $HOME/java/jdk1.7.0_25
```

- You must also configure the directory where Hadoop will store its data files, the network ports it listens to, etc. This setup will use Hadoop's Distributed File System (HDFS). Create a directory named "**tmp**" in the "**hadoop**" directory, which will be used as the base temporary directory for the local file system and the HDFS. You will use the path of this directory as `hadoop.tmp.dir` in file "**conf/core-site.xml**."

```
[user@vcl-host ~]$ cd hadoop
[user@vcl-host hadoop]$ mkdir tmp
[user@vcl-host hadoop]$ cd ..
```

Add the following lines between the `<configuration> … </configuration>` tags in the corresponding XML file.

In file "**conf/core-site.xml**," add:

```
<property>
  <name>hadoop.tmp.dir</name>
  <value>/home/user/hadoop/tmp</value>
  <description>A base for other temporary
  directories.</description>
</property>

<property>
  <name>fs.default.name</name>
  <value>hdfs://localhost:54310</value>
  <description>The name of the default file system.  A URI whose
  scheme and authority determine the FileSystem implementation.
  The uri's scheme determines the config property (fs.SCHEME.impl)
  naming the FileSystem implementation class.  The uri's authority
  is used to determine the host, port, etc. for a filesystem.
  </description>
</property>
```

In the file above, substitute `/home/user` for your home directory. To see the path of your home directory, use command `echo $HOME`.

In file "**conf/mapred-site.xml**," add:

```
<property>
  <name>mapred.job.tracker</name>
  <value>localhost:54311</value>
  <description>The host and port that the MapReduce job tracker
  runs at.  If "local", then jobs are run in-process as a single
  map and reduce task.</description>
</property>
```

In file "**conf/hdfs-site.xml**," add:

```
<property>
  <name>dfs.replication</name>
  <value>1</value>
  <description>Default block replication.
  The actual number of replications can be specified when the file
  is created. The default is used if replication is not specified
```

```
  in create time.</description>
</property>
```

- As mentioned before, the VCL machine does not support the command "`ssh localhost`." Therefore, open files "**conf/masters**" and "**conf/slaves**," and change `localhost` to the IP address of your VCL machine (e.g, `152.xxx.xxx.xxx`) in both files.

Start the single-node Hadoop cluster

- Use command `namenode` to format the HDFS filesystem.

```
[user@vcl-host ~]$ cd $HOME/hadoop/hadoop-1.2.1
[user@vcl-host hadoop-1.2.1]$ bin/hadoop namenode -format
[…] INFO namenode.NameNode: STARTUP_MSG:
/************************************************************
STARTUP_MSG:   Starting NameNode
STARTUP_MSG:   host = bnxxx-xxx.dcs.mcnc.org/152.xxx.xxx.xxx
STARTUP_MSG:   args = [-format]
STARTUP_MSG:   version = 1.2.1
STARTUP_MSG:   build =
https://svn.apache.org/repos/asf/hadoop/common/branches/branch-
1.2 -r 1503152; compiled by 'mattf' on Mon Jul 22 15:23:09 PDT
2013
STARTUP_MSG:   java = 1.7.0_25
************************************************************/
[…]
/************************************************************
SHUTDOWN_MSG: Shutting down NameNode at bnxxx-
xxx.dcs.mcnc.org/152.xxx.xxx.xxx
************************************************************/
```

- Use command `start-all.sh` to start your single-node Hadoop cluster. This command will start up a `NameNode`, `DataNode`, `JobTracker` and `TaskTracker` on your machine.

```
[user@vcl-host hadoop-1.2.1]$ bin/start-all.sh
starting namenode, logging to /home/user/hadoop/hadoop-
1.2.1/libexec/../logs/hadoop-user-namenode-vcl-
host.dcs.mcnc.org.out
152.xxx.xxx.xxx: starting datanode, logging to
/home/user/hadoop/hadoop-1.2.1/libexec/../logs/hadoop- user-
namenode-vcl-host.dcs.mcnc.org.out
152.xxx.xxx.xxx: starting secondarynamenode, logging to
/home/user/hadoop/hadoop-1.2.1/libexec/../logs/hadoop-user-
secondarynamenode-vcl-host.dcs.mcnc.org.out
starting jobtracker, logging to /home/user/hadoop/hadoop-
1.2.1/libexec/../logs/hadoop-user-jobtracker-vcl-
```

```
host.dcs.mcnc.org.out
152.xxx.xxx.xxx: starting tasktracker, logging to
/home/user/hadoop/hadoop-1.2.1/libexec/../logs/hadoop-user-
tasktracker-vcl-host.dcs.mcnc.org.out
```

- Verify that the Hadoop processes are running using command `jps`. If the Hadoop processes are running, then your single-node Hadoop cluster has been successfully set up.

```
[user@vcl-host hadoop-1.2.1]$ jps
2822 NameNode
3135 JobTracker
3253 TaskTracker
3056 SecondaryNameNode
2937 DataNode
4716 Jps
```

Stop the single-node Hadoop cluster

- Stop your single-node Hadoop cluster.

```
[user@vcl-host hadoop-1.2.1]$ bin/stop-all.sh
stopping jobtracker
152.xxx.xxx.xxx: stopping tasktracker
stopping namenode
152.xxx.xxx.xxx: stopping datanode
152.xxx.xxx.xxx: stopping secondarynamenode
[user@vcl-host hadoop-1.2.1]$ cd $HOME
```

References

- http://www.michael-noll.com/tutorials/running-hadoop-on-ubuntu-linux-single-node-cluster/