Instructions for using the Qloud (Hadoop 0.20)

Cloud Infrastructure at CMUQ (The Qloud)

CMUQ has dedicated cloud infrastructure running on a 14-blade IBM Blade-center with a total of 112 physical cores and 7 TB storage. The cloud is managed using advanced management software (IBM Cloud 1.4) with Tivoli provisioning manager which allows for cloud resources to be provisioned via software.

Users can request *virtual machines* (VMs) on the cloud with customized CPUs, RAM and disk space, as well as custom OS/software images. These machines run Red Hat Linux on top of the Xen virtualization platform and will be pre-configured with Java and the Hadoop SDKs. Since these resources are virtualized and provisioned through the management server, cloud resources can be accessed only through the *Cloud Gateway* (see Figure 1).

From your computer, you will need to log on to hadoop.qatar.cmu.edu server which is where you will run Eclipse and communicate to the provisioned cloud. This machine does not run any of the Hadoop code, it just acts as a liaison with your provisioned cloud.

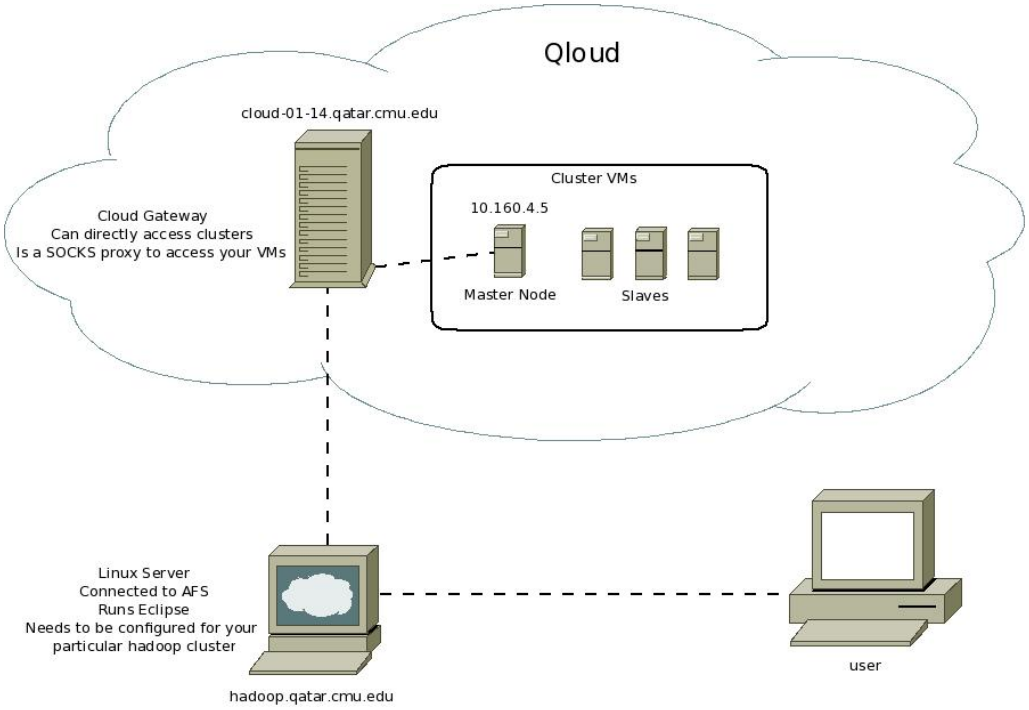

Figure 1 - CMU Cloud Infrastructure Logical View

Qloud User Requirements

1. Login account for CMUQ network. Please contact someone if you need one. In this document, we will assume that this username will be **login1.**
2. Login for the Cloud management software (Provided to you along with this document). We will assume that this username will be **login2.**
3. Basic knowledge of Unix commands and Unix text editors.
4. X-Win32 for windows machines. (can be downloaded from http://www.qatar.cmu.edu/myandrew/ - login required). You may also need WinSCP, PuTTy and other software to interact with the cloud servers from Windows machines.

Part I: Setting up your Cloud

1. Start X-Win32 and configure an SSH connection to **hadoop.qatar.cmu.edu** using **login1**

   Start Firefox remotely on hadoop.qatar.cmu.edu by using the following command: **`firefox &`**
2. Choose *Edit->Preferences*.
3. Click on *Advanced*, and then *Network*, and then *Settings*.
4. Choose Manual proxy configuration, and under SOCKS Host enter "**cloud-01-14.qatar.cmu.edu**", and for the SOCKS port enter 3900.
5. Click *OK*, and then *Close*.
6. Goto the address http://10.160.0.100:9080/cloud/
7. Login using the supplied username and password and request a Cloud.
8. Select the required dates (choose the Cloud for the semester duration). Include your Name in the Project name and Choose Project Type: **"Hadoop customized for CMU**"
9. Confirm your choice and your cloud will be available in about 24 hours.
10. Check back in 24 hours using this website from hadoop.qatar.cmu.edu. Verify that you have the number of requested VMs and that they are all active. You may expand the information on each VM (node) and check its IP address and its Admin password. **The last listed node is your master node.**
11. You may use this web interface at anytime to get current status of your provisioned cloud.

Part II: Configuring Hadoop on your Cloud and Command Line execution of MapReduce programs.

In this section, you will be accessing your Cloud remotely through the master node. This can only be done through the cloud gateway server: **cloud-01-14.qatar.cmu.edu**.

1. SSH to **cloud-01-14.qatar.cmu.edu.** using **login2**
2. SSH to your Master node by using the following command:

   `ssh root@10.160.4.5` *(Use IP address and password of your master node that you got from the last step of the previous part.)*
3. Execute the following commands to start the Hadoop daemons on your Cloud:

   ```
   su - hadoop
   hadoop namenode -format
   start-all.sh
   hadoop dfs -chmod 777 /
   ```

   If, for some reason in the future you need to re-start your Hadoop Cloud, you may run these commands but skip the **`namenode -format`** command.
4. Verify that Hadoop is working correctly on your PC by executing the PiEstimation example in the hadoop examples JAR file.

   ```
   cd /hadoop/hadoop-0.20.1/
   hadoop jar hadoop-0.20.1-examples.jar pi 10 100
   ```

   This will run the Pi estimation code using random sampling, the first argument (10) being the number of maps and the second argument (100) number of samples per map. Source code of this program is available at `src/examples/org/apache/hadoop/examples/PiEstimator.java` in the current folder.

   While your job is running, you may use the jobtracker web interface to see the progress the job. Open Firefox from hadoop.qatar.cmu.edu (as described in steps 1 and 2 of Part 1), and go to the web address `http://10.160.4.5:50030` *(Use IP address of your master node and port 50030)*. Browse through the job tracker interface for more information regarding the jobs on your Hadoop system.

Part III: Using HDFS

In this section, you will be accessing your Cloud through **hadoop.qatar.cmu.edu**. For this, we first need to copy the configuration files from your Cloud to that machine and configure Hadoop to access your Cloud through a SOCKS proxy (**cloud-01-14.qatar.cmu.edu**).

1. SSH to **cloud-01-14.qatar.cmu.edu.** using **login2**
2. SSH to your Masternode by using the following command:

   `ssh root@10.160.4.5` *(Use IP address of your master node.)*

3. Execute the following commands to copy the configuration files to **hadoop.qatar.cmu.edu**:

   "scp root@10.160.4.5:/hadoop/hadoop-0.20.1/conf/*-site.xml ."

   "scp *-site.xml hadoop.qatar.cmu.edu:hadoop-conf/"

4. SSH to **hadoop.qatar.cmu.edu.** using **login1.** Go to the directory **hadoop-conf/.** Edit **core-site.xml**, and add the following property before the last line (**</configuration>**): Use any text editor available (vim and nano).

```
<property>
  <name>hadoop.socks.server</name>
  <value>cloud-01-14.qatar.cmu.edu:3900</value>
  <description> Address (host:port) of the SOCKS server
to be used by the
    SocksSocketFactory.
  </description>
</property>
```

5. Still editing **core-site.xml**, find the *`hadoop.rpc.socket.factory.class.ClientProtocol`* and *`hadoop.rpc.socket.factory.class.JobSubmissionProtocol`* properties. For both of them, change the **`value`** to *`org.apache.hadoop.net.SocksSocketFactory`* and remove the *"`<final>true</final>`" attribute.*

6. Edit the **mapred-site.xml**, and add the following XML properties to the file before **<\configuration>**

   <property>
   <name>mapred.reduce.tasks</name>
   <value>4</value>
   </property>

<property>
<name>mapred.reduce.copy.backoff</name>
<value>1</value>
</property>

7. Run the following commands to setup your HDFS filesystem on this server: These commands will create the input directory for the wordcount application to be run in part V.

```
hadoop dfs -mkdir /user
hadoop dfs -mkdir /user/me
hadoop dfs -mkdir /user/me/wordcount
hadoop dfs -mkdir /user/me/wordcount/in
```

More information on the HDFS shell commands are available at http://hadoop.apache.org/common/docs/current/hdfs_shell.html

8. Once you have the HDFS filesystem ready, you may view the HDFS Namenode interface to get a better idea about your file system at: `http://10.160.4.5:50070` *(Use IP address of your master node and port 50070)*.

Part IV: Configuring Eclipse

1. Launch eclipse using the command **eclipse.** Open a Map/Reduce perspective. Select the Map/Reduce Locations tab at the bottom of the screen. Click the blue elephant icon to create a new location. Use the following configuration (Replace the Map/Reduce Master Host with the IP address of your master node).

```
Location Name: <some meaningful name>
Map/Reduce Master Host: 10.160.4.5
Map/Reduce Master Port: 9001
DFS Master port: 9000
SOCKS Proxy: ENABLED
SOCKS Proxy Host: cloud-01-14.qatar.cmu.edu
SOCKS Proxy Port: 3900
```

Switch to the advanced parameters tab and set
`hadoop.tmp.dir: /tmp/hadoop` (remove any -username bits)
Then click Finish.

Part V: Using Eclipse to run Wordcount

1. SSH to **hadoop.qatar.cmu.edu.** using **login1.**

2. Use the HDFS put command to store some input text files in **`/user/me/wordcount/in`**

   `hadoop dfs –put somefiles /user/me/wordcount/in`

3. Open **`eclipse.`** Open a new project (chose a Map/Reduce Project)
4. Give your project a name/location, then click "Configure Hadoop install directory"
5. Set the Hadoop Installation Directory to "**`/afs/qatar.cmu.edu/course/15/319/hadoop`**" and click Finish
6. In the Project Explorer window, open your project, right click the src folder and create a new class named WordCount (ignore everything but the class name).
7. You will want to copy the entire contents of **`/afs/qatar.cmu.edu/course/15/319/hadoop/src/examples/org/apache/hadoop/examples/WordCount.java`** into your class file, deleting anything that was in there first. Remove the first line that says **`package org.apache.hadoop.example`**
8. Click the green Run icon. Select "Run on Hadoop" and click OK. Choose your existing Hadoop location from the list. You should get an error about usage. This is CORRECT.
9. Right click your project name and then choose Properties. From there, click on Run/Debug Settings, then click on the single defined Launch Configuration (should be named WordCount), and then Edit. Click on the "Arguments" tab. You need to enter **`"/user/me/wordcount/in /user/me/wordcount/out`**" in the Arguments box and then choose OK.
10. Click the Run button again. Your job should now run to completion. This confirms that your Cloud is working correctly with eclipse.