Purdue University

Fall 2016

CS 541: Database Systems

Hadoop Cluster: Account Setup

/**********

* Overview

**********/

In the final project for this course you will have the opportunity to evaluate and extend Big Data processing platforms such as Hadoop and Spark. To help you get started on the project, we are providing you access to a Hadoop cluster. You may use this cluster (in addition to your personal machines) only for the purpose of CS541 final project. Please follow the instructions in this handout to get started as early as possible (**Final Project is Due on December 1**). If you encounter any problems, you may post in the Piazza final_project folder.

To learn more about Hadoop, you may read about the architecture of two core Hadoop components, HDFS and YARN:

http://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-hdfs/HdfsDesign.html
http://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/YARN.html

/**********

* Environment

**********/

We will use Purdue's OpenStack cluster for the final project. The master node for this cluster is openstack-vm-11-236.rcac.purdue.edu. You can SSH into the master node with credentials username:[PurdueID]_ostack, password:[PurdueID]_ostackpwd. (By PurdueID, we mean the name that you use to login to MyPurdue, Blackboard, etc.) If you are connecting from off campus, you will need to set up a VPN connection first. Instructions for setting up a VPN to Purdue's network are here: https://www.itap.purdue.edu/connections/vpn/ .

Be sure to change your password after you log in.

```
$ passwd
```

To see a list of all the nodes in the cluster, run

```
$ cat /etc/hosts
```

CAUTION: This cluster is temporary. It will be wiped after the project is graded. If you have any code or results that you wish to save, move them to permanent storage on another system.

NOTE: This cluster does not mount the CS Department's NFS shared file system, so your CS home directory is not available.

We are running the Cloudera Cluster Manager on the cluster. You can access the Cluster Manager web UI at http://openstack-vm-11-236.rcac.purdue.edu:7180 . Log in with username:guest, password:guestpwd. Again, if you are off campus, you must set up a VPN for this to work. You can use the web UI to explore HDFS, check on the status of jobs, check the cluster workload, etc. Note that you will not need to access the web UI to complete the project: The UI is available simply for you to explore.

```
/**********
* Basic commands for HDFS and MapReduce
**********/
```

We would like to confirm that you are able to perform basic data management operations on the cluster:

* Move files from the cluster master node to HDFS and back.
* Compile and run a MapReduce job.

After logging on to the cluster master node with SSH, populate a simple text file:

```
$ printf "aaa\nbbb\nccc\nddd\naaa\nbbb\nccc\nddd\n" > tmp.txt
$ cat tmp.txt
```

Create a directory in your personal HDFS directory to store the file:

```
$ hdfs dfs -mkdir /user/[PurdueID]_ostack/in
```

Copy the text file from the master node to the new HDFS directory:

```
$ hdfs dfs -put ./tmp.txt /user/[PurdueID]_ostack/in
```

List the directory contents:

```
$ hdfs dfs -ls /user/[PurdueID]_ostack/in
```

Now that we have loaded a file into HDFS, we can run a MapReduce job over it. Copy the file Select.java from your personal machine to the cluster master

node. On the master node, navigate to the directory which holds Select.java and run the following commands:

```
$ mkdir select
$ CP=$(hadoop classpath)
$ javac -classpath $CP -d select/ Select.java
$ jar -cvf select.jar -C select .
$ hadoop jar select.jar org.myorg.Select /user/[PurdueID]_ostack/in
/user/[PurdueID]_ostack/out
```

You do not need to worry about the details of the Select.java job or the steps required to compile and submit a MapReduce job. This exercise is simply to test the functionality of the cluster.

Next, check the output:

```
$ hdfs dfs -ls /user/[PurdueID]_ostack/out
$ hdfs dfs -cat /user/[PurdueID]_ostack/out/*
```

Move the output from HDFS back to your directory on the master node:

```
$ hdfs dfs -getmerge /user/[PurdueID]_ostack/out ./output.txt
$ cat output.txt
```

Finally, remove the output directory so that you can run your job again. (Hadoop will complain if you try to write to an HDFS directory that already exists.)

```
$ hdfs dfs -rm -r /user/[PurdueID]_ostack/out
```