UGA CSCI4795/6795: Cloud Computing (Spring 2019)
Programming Assignment 4 (PA#4) (out: Apr 02 2019; due: Apr 13 2019 – 11:59 p.m.)

Goals

Gain hands-on experience with Google Cloud Platform and MapReduce (Hadoop)

Introduction

Replacing existing heating/cooling ("HVAC") systems can have a significant impact on the environment (as well as saving some money!) But in the absence of an outright failed system, identifying the specific system to replace can be challenging – one must consider years in service, efficiency of the unit, maintenance cost/records, general comfort level, user complaints, tax benefits, etc. In this assignment, we are going to do some analysis that could help us make such a decision. You are given a dataset of measurements from a small collection of buildings. In this assignment, you must use Hadoop to determine:

1. The 3 worst HVAC systems, based on all available data (where "worst" means the greatest difference between desired temperature and actual temperature)
2. The 3 hottest buildings, based on all available data, during normal business hours: show a plot of average temperature in each of the 3 buildings as a function of time-of-day

Note that in many situations (e.g., as a "data scientist"), you are asked to determine certain info from data. At a high level, the request is fairly clear. (e.g., "let's identify the worst HVAC system -- maybe it is more cost-effective to just replace it!"), but then as you get closer to the actual data, it's not so obvious how to map/transform/analyze the available data to answer the question. In some of these situations, you can go back to the request-maker, explain the lack of a trivial mapping from the data to the answer to the question, and they can say what to do. But in other situations, the request-maker is "disconnected" from the data and would not understand how to make such a mapping. It would be up to you to determine how best to use the data. We're going to treat PA#4 as this latter situation. YOU must make (and justify) an interpretation of the data that is logical to answer the higher-level question.

In addition, as a data scientist, be careful: do NOT just do something that is easy to compute as opposed to logical. For example, while it is easy to compute the HVAC unit that has the MAXIMUM single-day difference between "desired temp" and "actual temp", this is probably not the most logical way to make a decision to replace an HVAC unit. (e.g., what if the "actual temp" reading for that day was just WRONG?) i.e., go from [1] high-level question, to [2] necessary computation on the data to answer the high-level question, to [3] implementation of #2. Do not consider #3 before #2 and choose something that's "easy to implement", and then try to justify it as appropriate for #2.

You are required to do this assignment alone and please read the assignment document carefully.

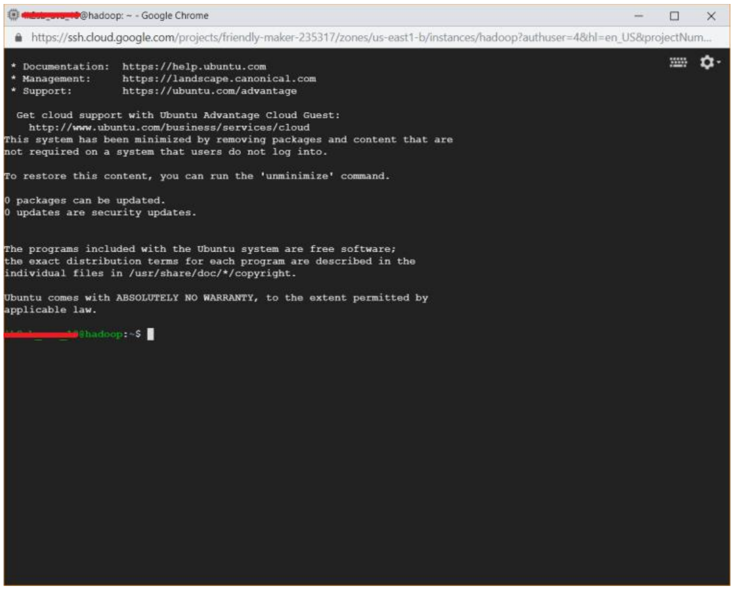
Part 1: Create an account for Google Cloud Platform and get an education credit of $50

I assume that you all have received an email titled “**[CSCI 4795/6795 -- Cloud] [Important] Google Cloud Platform Education Credit**” and you all have completed the procedure to get the credit. If not, please do this now. Without the credit, you cannot do this assignment.

Part 2: Create an Ubuntu 16.04 VM instance from Google Cloud Platform

In this part, we will create our first ubuntu VM instance from GCP and set up firewall rule for Hadoop assignment.

1. Go to Google Cloud Platform (GCP): https://cloud.google.com/
2. Click “Go To Console” button. If you cannot see “Go To Console” button, please click “Get Started” or “Try GCP Free.” You may be required to complete the account setup for Google Cloud Platform. i.e., address info, credit card information.
3. Click “Compute Engine” on left side, click “VM instance”, and click “Create.”
4. [On the “Create an Instance” page] name the instance “hadoop” and change machine type to “4 vCPUs” (15 GB memory)
5. Change Boot disk from “Debian GNU/Linux 9 (stretch) “ to “Ubuntu 16.04 LTS“, (on the boot disk page) change “boot disk size“ to “32“ GB, then hit the “select“ button
6. Check both “Allow HTTP traffic” and “Allow HTTPS traffic” and hit “create”. Then you will see your first VM instance in GCP with the name “hadoop”
7. Click the “Hadoop” instance and you will see the detail page of the instance. Go to “Network interfaces”, which is in the middle of the page. Click nic name (e.g., **`nic0`**).
8. [On the Network interface details page] Click Firewall rules on the left side.
9. [On the Firewall rules page] Click “Create Fire Rule”.
10. [On the Create a firewall rule page] Name it “default-allow-all”, select “All instances in the network” for Targets, Source IP ranges should be “0.0.0.0/0”, check “Allow all” for protocols and ports, and hit “create”
11. Click “Google Cloud Platform” on the left top. Next is “Compute Engine” --> VM Instances, then you will see your hadoop instance. Click “**`ssh`**” button, then another pop-up window will show up and you will log into your instance.



[redacted]@hadoop: ~ - Google Chrome

https://ssh.cloud.google.com/projects/friendly-maker-235317/zones/us-east1-b/instances/hadoop?authuser=4&hl=en_US&projectNum...

```
 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  Get cloud support with Ubuntu Advantage Cloud Guest:
    http://www.ubuntu.com/business/services/cloud
This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.

0 packages can be updated.
0 updates are security updates.


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

[redacted]@hadoop:~$ 
```

Part 3: Install/Configure Hadoop on GCP Ubuntu 16.04 Server

In this part, we will install Hadoop version 2.7.6 on the VM instance you created in the last part.

12. Install Java
    a. sudo apt-get update
    b. sudo apt-get install -y default-jdk
13. Add Hadoop/YARN Users
    a. sudo addgroup hadoop
    b. sudo adduser --ingroup hadoop hduser (please enter password and just enter for other fields)
    c. sudo adduser --ingroup hadoop yarn (please enter password and just enter for other fields)
    d. sudo usermod -a -G hadoop $(whoami)
14. Setup `SSH Key` for both Hadoop and Yarn
    a. Set key for Hadoop
        i. sudo su - hduser
        ii. ssh-keygen -t rsa -P "" (Press enter when asks for a file)
        iii. cat $HOME/.ssh/id_rsa.pub >> $HOME/.ssh/authorized_keys
        iv. ssh localhost (Enter "yes" when confirmation asked, this is asked only first time.)
        v. exit
        vi. exit (two exits)
    b. Set key for YARN
        i. sudo su - yarn
        ii. ssh-keygen -t rsa -P "" (Press enter when asks for a file)
        iii. cat $HOME/.ssh/id_rsa.pub >> $HOME/.ssh/authorized_keys
        iv. ssh localhost (Enter "yes" when confirmation asked, this is asked only first time.)
        v. exit
        vi. exit
15. Download `hadoop-2.7.6.tar.gz`
    a. cd
    b. wget http://www-us.apache.org/dist/hadoop/common/hadoop-2.7.6/hadoop-2.7.6.tar.gz
    c. tar xvf hadoop-2.7.6.tar.gz
    d. sudo mv hadoop-2.7.6 /usr/local/hadoop
16. Create directories for namenode and datanode, and set permission for these directories
    a. sudo mkdir -p /usr/local/hadoop/data/namenode
    b. sudo mkdir -p /usr/local/hadoop/data/datanode
    c. sudo mkdir -p /usr/local/hadoop/logs
    d. sudo chown -R hduser:hadoop /usr/local/hadoop
    e. sudo su - hduser
    f. chmod g+w /usr/local/hadoop/logs
    g. exit

17. Update .bashrc for your own account, hduser, and yarn
    a. (for your own account)
        i. open **.bashrc** (using editor. e.g, emacs)
        ii. add the following lines at the end of the **.bashrc** file

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
export HADOOP_INSTALL=/usr/local/hadoop
export HADOOP_CONF_DIR=$HADOOP_INSTALL/etc/hadoop
export YARN_CONF_DIR=$HADOOP_INSTALL/etc/hadoop
export PATH=$PATH:$HADOOP_INSTALL/bin
export PATH=$PATH:$HADOOP_INSTALL/sbin
export HADOOP_MAPRED_HOME=$HADOOP_INSTALL
export HADOOP_COMMON_HOME=$HADOOP_INSTALL
export HADOOP_HDFS_HOME=$HADOOP_INSTALL
export YARN_HOME=$HADOOP_INSTALL
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_INSTALL/lib/native
export HADOOP_OPTS="-Djava.library.path=$HADOOP_INSTALL/lib"
export HADOOP_OPTS="$HADOOP_OPTS -Djava.library.path=$HADOOP_INSTALL/lib/native"
export JAVA_LIBRARY_PATH=$HADOOP_INSTALL/lib/native
export LD_LIBRARY_PATH=$HADOOP_INSTALL/lib/native:$LD_LIBRARY_PATH
export YARN_EXAMPLES=$HADOOP_INSTALL/share/hadoop/mapreduce
export HADOOP_MAPRED_STOP_TIMEOUT=30
export YARN_STOP_TIMEOUT=30
```

        iii. **source ~/.bashrc**
    b. (for hduser)
        i. sudo su - hduser
        ii. open and edit **~/.bashrc** (using editor. e.g, emacs)
        iii. exit
    c. (for yarn)
        i. sudo su - yarn
        ii. open and edit **~/.bashrc** (using editor. e.g, emacs)
        iii. exit

18. Set Hadoop configurations (**pseudo-distributed mode**)
    a. (sudo) Edit **/usr/local/hadoop/etc/hadoop/core-site.xml** to be:

```
<configuration>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://localhost:9000</value>
  </property>
</configuration>
```

    b. (sudo) Edit **/usr/local/hadoop/etc/hadoop/hdfs-site.xml** to be:

```
<configuration>
  <property>
      <name>dfs.replication</name>
      <value>1</value>
  </property>
  <property>
      <name>dfs.namenode.name.dir</name>
      <value>file:/usr/local/hadoop/data/namenode</value>
  </property>
  <property>
      <name>dfs.datanode.data.dir</name>
      <value>file:/usr/local/hadoop/data/datanode</value>
  </property>
  <property>
      <name>dfs.permissions.supergroup</name>
      <value>hadoop</value>
  </property>
</configuration>
```

c. Create **/usr/local/hadoop/etc/hadoop/mapred-site.xml** from the template (**mapred-site.xml.template**)
   i. sudo su hduser
   ii. cp /usr/local/hadoop/etc/hadoop/mapred-site.xml.template /usr/local/hadoop/etc/hadoop/mapred-site.xml
   iii. exit

d. (sudo) Edit **/usr/local/hadoop/etc/hadoop/mapred-site.xml** to be:

```
<configuration>
  <property>
      <name>mapreduce.framework.name</name>
      <value>yarn</value>
  </property>
</configuration>
```

e. (sudo) Edit **/usr/local/hadoop/etc/hadoop/yarn-site.xml** to be:

```
<configuration>
  <property>
      <name>yarn.nodemanager.aux-services</name>
      <value>mapreduce_shuffle</value>
  </property>
  <property>
      <name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>
      <value> org.apache.hadoop.mapred.ShuffleHandler</value>
  </property>
  <property>
    <name>yarn.resourcemanager.hostname</name>
    <value>localhost</value>
  </property>
</configuration>
```

f. (sudo) Edit **/usr/local/hadoop/etc/hadoop/hadoop-env.sh** to point **JAVA_HOME** at **/usr/lib/jvm/java-8-openjdk-amd64** -- i.e., change this line: “`export JAVA_HOME = $(JAVA_HOME)`” to: “`export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64`”

g. (sudo) Edit **/usr/local/hadoop/etc/hadoop/mapred-env.sh** to be:

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
export HADOOP_MAPRED_IDENT_STRING=hduser
```

Part 4: Initialize and boot Hadoop

19. sudo su - hduser
20. $HADOOP_INSTALL/bin/hdfs namenode -format (This is to format namenode.)
21. $HADOOP_INSTALL/sbin/start-dfs.sh (accept connection to 0.0.0.0)
22. $HADOOP_INSTALL/sbin/stop-dfs.sh
23. exit

Part 5: Run sample streaming wordcount (python)

24. Start HDFS and YARN resource manager
    a. sudo su -p - hduser -c $HADOOP_INSTALL/sbin/start-dfs.sh
    b. sudo su -p - yarn -c $HADOOP_INSTALL/sbin/start-yarn.sh
25. hdfs dfs -mkdir /user
26. hdfs dfs -mkdir /user/$(whoami)
27. cd
28. wget http://cobweb.cs.uga.edu/~kim/classes/S19-CSCI4795-6795/PA4/WordCntmapper.py
29. chmod 755 WordCntmapper.py
30. wget http://cobweb.cs.uga.edu/~kim/classes/S19-CSCI4795-6795/PA4/WordCntreducer.py
31. chmod 755 WordCntreducer.py
32. hdfs dfs -put /usr/local/hadoop/etc/hadoop CSCI4795input
33. hdfs dfs -ls CSCI4795input/ (check all files are in CSCI4795input directory in HDFS)
34. hadoop jar /usr/local/hadoop/share/hadoop/tools/lib/hadoop-streaming-2.7.6.jar -file WordCntmapper.py -mapper WordCntmapper.py -file WordCntreducer.py -reducer WordCntreducer.py -input CSCI4795input -output py_wc_out
35. hdfs dfs -cat py_wc_out/part-00000 | more (after job completion)

Part 6: Analyze HVAC dataset (python)

Now, using Python, write the two separate Map/Reduce programs (identified earlier) using Hadoop 2.7.6 on GCP to compute the following using the sample HVAC data: http://cobweb.cs.uga.edu/~kim/classes/S19-CSCI4795-6795/PA4/HVAC.csv You are required to run your program(s) via Hadoop 2.7.6 pseudo-distributed mode on an GCP instance. You cannot use any "add-on" to Hadoop (such as Hive). **The great majority of the data processing must be performed within Hadoop; relatively minor "post-processing" (e.g., sort) is allowed outside of Hadoop.** You are required to submit two items via eLC:

1. A PDF document containing the results of the execution of the 2 programs. Along with each result, write a short paragraph explaining how you used your programs to generate the results. Include a description of any post-processing you performed outside of Hadoop to generate these results. Re: the "plot" required of the second program: one way to do this is to generate the necessary info (cloud-side) and then cut-and-paste it into Excel running on your laptop (and then make the visual rep of this).
2. A zip file containing all of your Map/Reduce programs.

Additional Notes

1. Start Hadoop (HDFS and YARN Resource Manager)
    - sudo su -p - hduser -c $HADOOP_INSTALL/sbin/start-dfs.sh
    - sudo su -p - yarn -c $HADOOP_INSTALL/sbin/start-yarn.sh
2. Stop Hadoop (HDFS and YARN Resource Manager)
    - sudo su -p - hduser -c $HADOOP_INSTALL/sbin/stop-dfs.sh
    - sudo su -p - yarn -c $HADOOP_INSTALL/sbin/stop-yarn.sh
    - HDFS commands guide: https://hadoop.apache.org/docs/r2.7.0/hadoop-project-dist/hadoop-common/FileSystemShell.html