CS595—Big Data Technologies

Assignment #3 (Modules 03a & 03b)

Due by the start of the next class period

Assignments can uploaded via the Blackboard portal

Note: There may be short quiz questions about readings, assignments or articles (except extra credit) in the class period when they are due.

1. Read from (TW)
   - Chapter 8
   - Chapter 9
   - Chapter 17

2) Set up your own Hadoop environment on the Azure Cloud. Follow the instructions in the document “Setting Up You Hortonworks Sandbox” in the ‘Fee Books and Chapters” section of the Blackboard.

3) Please read the document “mrjob Documentation,” which is located in the “Free Books and Chapters” section of the Blackboard, through page 14

4) Install the python mrjob library on your Hadoop sandbox.

- Log on to the maria_dev account
- Enter “su root”
  - You will be asked for the root password, enter the word: hadoop
  - You will then be asked again for this password, and finally asked to supply a new root password which you should remember.
  - Now you need to make a small change to the following file /etc/yum.repos.d/sandbox.repo
  - Use the vi editor to open this file by entering ‘vi /etc/yum.repos.d/sandbox.repo’
  - If you don’t know how to use the vi editor, use google to find a tutorial
  - Then change the line ‘enabled=1’ to ‘enabled=0’
  - Save the file using the ‘wq’ command
- Enter “yum install python-pip”
- Enter “yum install nano”
- Enter “pip install mrjob==0.5.11”
- Enter “exit”
  - This causes you to leave root mode

4) Next you will set up to execute the provided WordCount mapreduce program found in the “Assignments” section of the Blackboard. This is the exact same program we saw in class.

Step 1:

Copy the two files “cs595words.txt” and “WordCount.py” to your PC or Mac. They are part of the documents included with the assignment.

Step 2:

Log on to your Hadoop environment and use the secure copy program to move the WordCount.py cs595words.txt file to the home directory of the maria_dev account which should be “/home/maria_dev”

If we assume that you have downloaded the WordCount.py file to /my/dir/WordCount.py on your mac or pc the command would be something like

scp –P 2222 /my/mydir/WordCount.py maria_dev@localhost:/home/maria_dev

Note that you need to us a capital “-P”.

Step 4:

Do the same for the assignment file cs595words.txt

In this case move the file from “/home/maria_dev” to the Hadoop file system, say to the directory “/user/maria_dev”

Step 5:

Now execute the following

python WordCount.py -r hadoop --hadoop-streaming-jar /usr/hdp/current/hadoop-mapreduce-client/hadoop-streaming.jar hdfs:///user/maria_dev/cs595words.txt

Note there must be three slashes in “hdfs:///” as “hdfs://” indicates that the file you are reading from is in the hadoop file system and the “/user” is the first part of the path to that file. Also note that sometimes copying and pasting this command form the assignment document does not work and it needs to be entered manually.

Check that it produces some reasonable output.

Note, the above command will erase all output files in hdfs. If you want to keep the output use the following command instead:

python WordCount.py -r hadoop --hadoop-streaming-jar /usr/hdp/current/hadoop-mapreduce-client/hadoop-streaming.jar hdfs:///user/maria_dev/cs595words.txt **--output-dir /user/maria_dev/some-non-existent-directory**

5) Now slightly modify the WordCount program. Call the new program WordCount2.py.

Instead of counting how many words there are in the input documents, modify the program to count how many words begin with the small letters a-n and how many begin with anything else.

The output file should look something like

a_to_n, 12

other, 21

Now execute the program and see what happens.

6) (5 points) Submit a copy of this modified program and a screen shot of the results of the program's execution as the output of your assignment.

7) Now do the same as the above for the files Salaries.py and Salaries.tsv. The ".tsv" file holds department and salary information for Baltimore municipal workers. Have a look at Salaries.py for the layout of the ".tsv" file and how to read it in to our map reduce program.

8) Execute the Salaries.py program to make sure it works. It should print out how many workers share each job title.

9) Now modify the Salaries.py program. Call it Salaries2.py

Instead of counting the number of workers per department, change the program to provide the number of workers having High, Medium or Low annual salaries. This is defined as follows:

High	100,000.00 and above
Medium	50,000.00 to 99,999.99
Low	0.00 to 49,999.99

The output of the program should be something like the following (in any order):

High 20

Medium 30

Low 10

Some important hints:

- The annual salary is a string that will need to be converted to a float.

- The mapper should output tuples with one of three keys depending on the annual salary: High, Medium and Low
- The value part of the tuple is not a salary. (What should it be?)

Now execute the program and see what happens.

10) (5 points) Submit a copy of this modified program and a screen shot of the results of the program's execution as the output of your assignment.

11) Now copy the file u.data to /user/maria_dev. This is similar to the file used for some examples in Module 03b. **NOTE: unlike the slide deck examples, this version of u.data has fields separated by commas and not tabs.**

12) (5 points) Review the slides 17-22 in lecture notes Module 3b. Now write a program to perform the task of outputting a count of the number of movies each user (identified via their user id) reviewed.

Output might look something like the following:

186: 2

192: 2

112: 1

etc.

Submit a copy of this program and a screen shot of the results of the program's execution (only 10 lines or so of the result) as the output of your assignment.