Big Data (ECE 595-004/007), Fall 2018
Hands-on 2: Map Reduce Programming
Due date: October 16, 2018, 11:59 PM

Note: Find the instructions for running the programs at the end.

Tasks

1. **(8 points)** Write a MapReduce program that reads "pg20417.txt" file from HDFS cluster and finds top 10 most frequent words and their frequencies. In the text file, many words may appear in different forms, e.g. The, the, you have to treat them same. In addition, some words may have double quote, single quote or other non-alphabet character in the prefix or suffix, your program should be able to remove them and then consider the remaining characters as word.

2. (**12 pts**) Write a MapReduce program that reads "purchases.txt" file from HDFS cluster and finds the standard deviation of stores' sales in each city. To calculate the standard deviation refer Welford's online algorithm. You may find Python code snippet at the link. The required parts of the algorithm are as follows:

$$\overline{x_n} = \overline{x_{n-1}} + \frac{x_n - \overline{x_{n-1}}}{n}$$

$$M_{2,n} = M_{2,n-1} + (x_n - \overline{x_{n-1}})(x_n - \overline{x_n})$$

$$\sigma^2{}_n = \frac{M_{2,n}}{n}$$

Here, x_n denotes the sample mean of the first n samples and $\sigma^2{}_n$ their population variance. Once you calculate the popular variance for each city and then you can take the square root of it, which will give you the standard deviation.

3. **(10 pts)** Write a MapReduce program that reads a web server log file "epa-http.txt" and finds the number of **unique files** request sent by each client. You may refer the link to understand the fields in the log file. The file request could be found inside the double quotes after GET.

Instructions

You can create a folder where you can keep your MapReduce code. Use naming format **mapper_firstname_lastname_problem_number.py** for Mapper code and **reducer_firstname_lastname_problem_number.py** for reducer code for each problem. For example, in my case, mapper_quamar_niyaz_p1.py and reducer_quamar_niyaz_p1.py will be the files' names for Problem 1.

Once you finish mapper and reducer code, you can verify their correctness by using them for a small file similar to the problem data file. You can create it by examining the file content for the given problem or copy-pasting first 50-100 lines from the data file.

```
cat students.txt | ./mapper_std.py | sort | ./reducer_std.py
```

Mapper verification

cat problem_file_small.txt | your_mapper_ Python_file

Check whether you are getting the same output that you want. If you do not then you need to fix the mapper code and execute the command again. It is assumed that you are executing the command from the folder where you have put MapReduce code and small data file.

Reduce verification

cat problem_file_small.txt | mapper.py | sort | your_reducer_ Python_file

Check whether you are getting the same output that you want. If you do not then you need to fix the reducer code and execute the command again. It is assumed that you are executing the command from the folder where you have put MapReduce code and small data file.

If your mapper and reducer work fine, then move the actual data file in the cluster and run MapReduce code on it. Since your MapReduce code is written in Python, you need to use Hadoop streaming. You can find the jar file for it inside **/usr/local/hadoop/share/hadoop/tools/lib/** folder if you have configured Hadoop according to the instructions in Hands-on 1. To avoid the long typing, you can copy the jar file in /usr/local/hadoop folder.

cp /usr/local/hadoop/share/hadoop/tools/lib/hadoop-streaming-2.9.1.jar /usr/local/hadoop/

Now execute the command similar to the example command given below:

```
hadoop jar /usr/local/hadoop/hadoop-streaming-2.9.1.jar  \
-mapper /path/to/your/MapReduceCode/folder/your_mapper_Python_file \
-reducer /path/to/your/MapReduceCode/folder/your_mapper_Python_file \
-input /path/to/folder/for/your/data/file/in/cluster/ -output /path/to/your/output/folder/in/cluster
```

The output file will be created in the output folder that you have provided in the command. The output file will have a prefix part. You can see the content of the file using **hdfs dfs -cat** command or you can download the file in the local file system using **hdfs dfs -get** command.

Note: Do not use any folder name for the output folder if that folder already exists in HDFS cluster. Either remove the existing folder or use other folder name.