CSCE/STAT 587 Spring 2023 **Hadoop Homework**

Background

The airline data that we've used for learning how to use map-reduce includes a large number of fields for things like month (`Month`), origin airport (`Origin`), and departure delay (`DepDelay`). The following problems requires you to process the airport data to answer questions about these times.

For each question, use the following data sets and approach:

Dataset: https://cse.sc.edu/~rose/590B/airline/2008.csv

Method 1: map-reduce
Method 2: Pig Latin
Method 3: HiveQL

If you are using wget to transfer the files, the command is of the form:

```
wget https://cse.sc.edu/~rose/590B/airline/2008.csv
```

My suggestion is that you develop your code (map-reduce, pig, and hive) using the smallest data set of 12 records (testNA.csv), then try the data set containing ~25K flight records (test_25K.csv). When you get that to work, then move on to the large file from 2008.

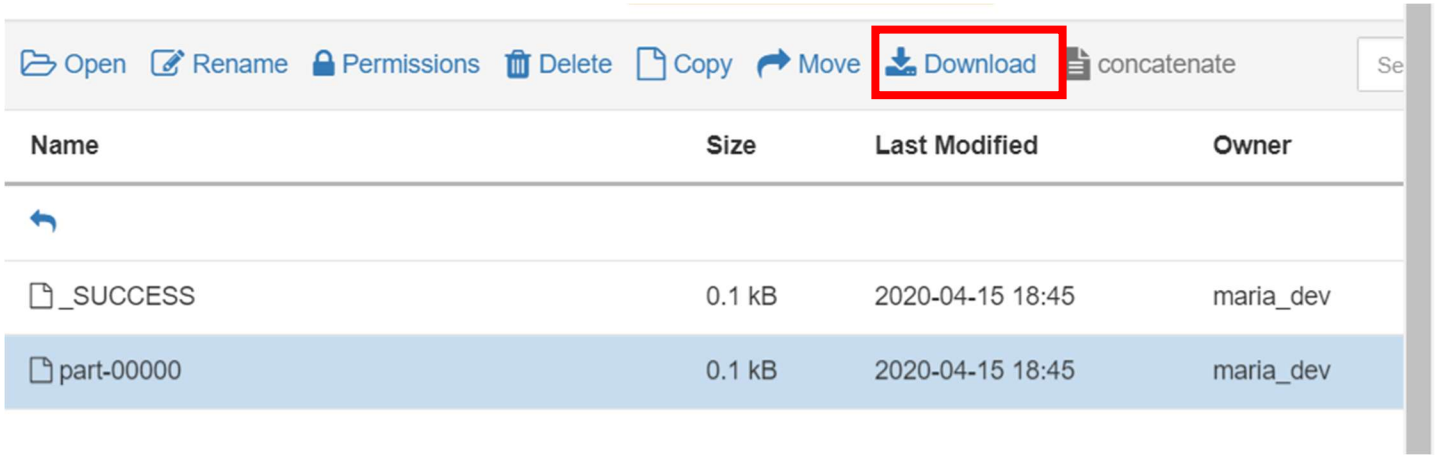
Transferring the input data set to HDFS

The data set 2008.csv is too large to use the drag-and-drop approach with Ambari. You will have to use wget to transfer it to the linux filesystem of your vm and then "hadoop fs -put" it to HDFS.

Saving output from HDFS to your local machine:

1) Login to Ambari (vm-hadoop-xx.cse.sc.edu:8080)
2) click on the table icon (tick-tack-toe) on the menu bar
3) select "Files View" from the table drop down menu
4) navigate to user/maria_dev then to the directory containing your output
5) click on the output file to select it.
6) You should then see a menu bar with file operations in blue lettering. The 7th operation is "Download". See the red box in the figure below:

7) Click on the download icon. This will download the file to the "Downloads" folder on your local machine.



CSCE/STAT 587 Spring 2023 **Hadoop Homework**

I. MAPREDUCE Problem 1 Find the average departure delay time by month using map-reduce: aveDepDelayByMonth

This problem is an organic extension of our understanding of map-reduce from counting words, to counting flights, to calculating average arrival delay to now finding the average departure delay times by Month. Our hypothesis is that there is a seasonal influence on delay and consequently, we expect some months exhibit larger departure delays. As a first step we need to calculate the average departure delay to see if this hypothesis might have merit. Start by extracting the `Month` and `DepDelay` columns. For definition of the columns see: http://stat-computing.org/dataexpo/2009/the-data.html . The goal of this problem is to find the average departure delay time for each month. Since the mapper and reducer communicate via key-value tuples, it would be easiest to select `Month` and `DepDelay` in the mapper and output the key-value pair as `Month`, `DepDelay`. As in the case of our mapreduce lab, the reducer must handle missing values (NA). you saw how to do this in the reducer.) Thus, the reducer must handle missing values (NA) and simply calculate the average of the `DepDelay` that it receives for each month. Also, for this problem we want the computed averages to be floating point results instead of integer results. Use float(x) to coerce x to be a floating point value.

Review your results and write a brief report describing how your results either support or do not support the hypothesis that some months exhibit larger departure delays. Independent of whether your results support this hypothesis or not, explain how you could definitively test this hypothesis using ideas that you should have learned this semester. Be sure to provide enough details as to how you would test this hypothesis. (Hint: review the lecture on hypothesis testing.)

Submit the python code for your mapper, reducer, and your output results.

1) Name your mapper file aveDepDelayByMonthMapper.py
2) Name your reducer file aveDepDelayByMonthReducer.py
3) Name your output file aveDepDelayByMonthMapReduceResults
4) Name your report aveDepDelayByMonthReport

Hints:

1) Read the file "TipsForMR.txt" This file is in the Class Materials folder on Teams. **Be smart! Test your mapper and reducer separately as explained in "TipsForMR.txt" BEFORE using hadoop in streaming mode.**
2) Start by refreshing your understanding of the aveArrDelay map-reduce example. The solution to that problem is very close to this problem.
3) Be sure to check for missing values (NA) in your reducer
4) Be sure to explicitly cast objects to be the type you want. If you want to interpret X as an float, use float(X). If you want to interpret the integer Y as a string use str(Y).

CSCE/STAT 587 Spring 2023 **Hadoop Homework**

Problem 2 Find the average departure delay time by month using Pig: aveDepDelayByMonth Create a pig solution for finding the average departure delay by month. This is the same problem you solved in map-reduce, but now you are creating a solution in Pig Latin. When indexing columns, don't forget that column numbers start from 0 in Pig Latin.

Submit your Pig Latin script and your output results.

1) Name your Pig Latin script aveDepDelayByMonth.pig
2) Name your output file aveDepDelayByMonthPigResults

Hints:

1) Start by refreshing your understanding of the aveArrDelay Pig Latin example. The solution to that problem is very close to this problem.
2) In Pig Latin, missing values are referred to as nulls. In the case of arithmetic, if either subexpression is null, the resulting expression is null. Since the AVG() function handles null values, you don't need worry missing values for this problem.

Problem 3 Find the average departure delay time by day of week using Hive: aveDepDelayByMonth
Create a hive solution for finding the average departure delay by month. This is the same problem you solved in map-reduce, but now you are creating a solution in HiveQL. When indexing columns, don't forget that column numbers start from 1 in HiveQL.

Submit your HiveQL script and output results.

1) Name your HiveQL script aveDepDelayByMonth.hql
2) Name your output file aveDepDelayByDayOfMonthResults

Hints:

1) Start by refreshing your understanding of aveArrDelay HiveQL. The solution to that problem is very close to this problem.
2) Be sure to check for missing values (NA) for DepDelay when using a SELECT statement