CPSC 5330 -- Spring 2024
Week 2 Lab: Hadoop Streaming and Sqoop

Part 1 (10 points)

Hadoop for Log Processing

The file `Week2/LabFiles/Hadoop_2k.log` (from https://github.com/logpai/loghub) contains log entries from a Hadoop run sampled for ten minutes (Oct 18, 2015, between 6:01PM and 6:10PM). Each log entry has a severity level, which is one of INFO, WARN, ERROR, and FATAL.

Use Hadoop Streaming to process these log lines to record, for each minute, the number of log entries in each category.

Your script, named `process-log-file`, will copy the input file to HDFS, then run Hadoop streaming to produce a file with one line for each minute in the log; each line should have these 6 fields:

1. Minute number (a number between 1 and 10)
2. Total number of log entries for that minute
3. Number of log entries for that minute with severity INFO
4. Number of log entries for that minute with severity WARN
5. Number of log entries for that minute with severity ERROR
6. Number of log entries for that minute with severity FATAL

Your script will then print the lines of the output file to standard output, in ascending order of minute number.

Use the `run-hadoop-streaming` script to structure your solution: create the directory `log-processing` containing your mapper and reducer, then your script `process-log-files` will call `run-hadoop-streaming` to kick off the log processing job, and then display (write to standard output) the properly sorted output.

For this question hand in:

1. Your script `process-log-file`
2. The folder `log-processing` containing your mapper and reducer
3. A file `log-processing.txt` containing the output of your script when run against the `Hadoop_2k.log` input file.

Part 2 (10 points)

Sqoop for On-Time Airline Tracking

The directory `Week2/LabFiles` contains a SQL dump file, `airline.sql`, taken from this repository: https://relational.fit.cvut.cz/dataset/Airline.

Use this database, Sqoop, and Hadoop to produce output for each airline (Carrier): the minimum, maximum, and average flight delay in minutes.

Your goal is to write a script `report-airline-delays` that writes one record for each carrier, in descending order of average *arrival* flight delay in minutes. Your script needs to set up the database, run Sqoop to import the data to HDFS, run the MapReduce job to aggregate the flight records, then print the sorted output records.

For example, here is one output line:

```
AA      0.000000        1659.000000     10.695254
```

For this question submit:

1. Your script `report-airline-delays`
2. A folder `airline-delays` containing your mapper and reducer code
3. A file `airline-delays.txt` containing the output of your script when run against the airline database table

Hint: the database table has lots of columns, most of which you don't need, and you don't want to define the full schema to Sqoop. Refer to the Sqoop documentation and the class slides to figure out how to import only the columns you need.

Part 3 (10 points)

Hadoop for Airport Delay Statistics

The directory `Week2/LabFiles/airstats` contains files recording information about flight delays and cancellations (and other attributes) for year 2012 through 2015. Each (JSON) file contains statistics for a single airport for a single month.

For this question you are interested in the fraction of flights that were delayed at an airport over the course of a year. Use the Hadoop Streaming framework to process these files and compute, for each year, which airport (code) and the lowest and highest percentage of delayed flights.

For example, here is one output line:

```
2013    EWR,0.26290449554588774,SLC,0.13690745823230036
```

In writing your solution

- You must use the Hadoop streaming framework, so you will write (only) a mapper and a reducer
- You must use the Python JSON library to parse the input. (Each input to the mapper will be a string which can be parsed to a Python dictionary using the Python JSON library.)

For this question submit:

- A directory `airstats` that contains your mapper and reducer code
- A file named `airstats.txt` which is the result of running your application against the files in the `Week2/LabFiles/airstats` directory

Hint: running Hadoop on the full data set takes a long time. Be sure to test first, initially using the Shell, then test using Hadoop on a small set of files. Then run on the full data set once you are confident your code is working properly.

Part 0, Style Points (2 points)

(These are easy points to get, just by following instructions and handing in good clean work!)

Was the output clean and well presented? Was the text and explanations well written?
Were all the files there and named according to instructions?

To Hand In

A Zip file containing (only) these files

- The files and directories called out in each of the three questions.
- A retrospective report in a file `retrospective.pdf` – a reflection on the assignment, with the following components
    - Your name
    - How much time you spent on the assignment
    - If parts of the assignments are not fully working, which parts and what the problem(s) are
    - Were there aspects of the assignment that were particularly challenging? Particularly confusing?
    - What were the main learning take-aways from this lab – that is, did it introduce particular concepts or techniques that might help you as an analyst or engineer in the future?