School of Computer Science, McGill University

COMP-421 Database Systems, Winter 2022

Written Assignment 3: MapReduce and Pig Latin

Due Date April 7, 12:00 noon EST

This is an individual assignment. You are required to work on your own to create the solution.

This assignment is worth 8% of your course grade. The total points in this assignment is 24.

Please read the complete assignment description, including the **Guidelines** section before starting your work. There are important instructions contained in them.

Ex. 1 — Using MapReduce to Process Semi-structured Data(3 Points)

Webserver logs play a significant role in web companies to help study customer characteristics and behavior. An example line in an apache log could look like this (a single line in the log file, is broken into two in this description to fit this document).

```
169.122.23.15 - frank [10/Oct/2000:13:55:36 -0700] ‘‘GET /apache_pb.gif HTTP/1.0’’ 200 2326
‘‘http://www.example.com/start.html’’ ‘‘Mozilla/4.08 [en] (Win98; I ;Nav)’’
```

Among several things, it tells the web server what is the geographical location of the customer (using IP address), type of web browser, operating system, that they use, etc. Therefore extracting such details out of the logs is a valuable activity for many organizations.
You can assume that the following high-level language library functions are available to help you parse the individual lines in the log files to help you get specific information.

```
extract_IP(linetext) // gives 169.122.23.15
extract_url(linetext) // gives /apache_pb.gif
extract_browser(linetext) // gives Mozilla/4.08
```

(a) Write a MapReduce workflow that will produce a report of the number of users for each browser. Assume that the input to the mapper is the linenumber (key) and the value is the text of that line. For simplicity, assume that an IP address uniquely identifies a user. Include only those browsers with more than 100 users.

(b) Once you have developed your workflow, give an example of what the input and output of each mapper and reducer in your workflow would look like. Writing this down as you develop your design may help you notice and fix any logical mistakes.

(c) If you were allowed to use the combine functionality for your solution, can it reduce the amount of I/O? If a combine is useful, what would be its logic? will it be identical to the reducer? Justify your position for each decision.(No points for this without explanation).

Turn in:- Your solution (typed) in **assignment3.pdf** under a section **Q1**.

Ex. 2 — Complex uses of MapReduce (5 Points)

Continuing from the previous question,

(a) Write a MapReduce workflow that compute the total number of users who use more than one browser. Your input is the line number in the log file (key) and the text in that line (value).

(b) How many reducers processes are involved for the last reducer step of your MapReduce workflow?

Make sure to include some comments, example inputs and outputs for each of the mappers and reducers, etc.,
Turn in:- Your solution (typed) in **assignment3.pdf** under a section **Q2**.

Ex. 3 — PigLatin - Warmup(0 Points)

The goal of this exercise is to help you verify that you are able to acess the MapReduce cluster and execute Pig Latin scripts:

The data set used for all the Pig Latin questions are based on a modfied version of the Covid vaccination data set[1] and population data set[2] (Fall 2021 snapshot) provided by WHO[3] which is already loaded into the HDFS.

- `pop.csv` is the population data set, that consists of the following fields:
  - `country` → Name of the country, unique in this data set.
  - `population` → Population of the country **in thousands**.
- `vaccination-data.csv` is the vaccination information as of Fall 2021, which consists of the following fields:
  - `country` → Name of the country, unique in this data set.
  - `iso3` → Country code, unique in this data set.
  - `who_region` → indication of the WHO region to which the country belongs to.
  - `persons_fully_vaccinated` → the number of people in the country that are fully vaccinated.
- `vaccination-metadata.csv` contains some metadata about different vaccines used in various countries and consists of the following fields:
  - `iso3` → Country code.
  - `vaccine_name` → combination of product and company name (i.e., the next two fields)
  - `product_name` → the brand name of the vaccine (e.g `Comirnaty`).
  - `company_name` → name of the company developing the vaccine (e.g `Pfizer BioNTech`).

You are provided with an `example.pig` script, that contains the necessary `LOAD` instructions to load the data from HDFS to a schema described above. You should be able to reuse this for the remaining exercises.

It is important that you read through the supporting **PigLatinInstructions-vvv.pdf** file before starting to write and execute the Pig Latin scripts.

You can either run the script as it is by passing the script as an argument.

```
$ pig example.pig
```

or by starting pig by itself first

```
$ pig
```

and then copy pasting each statement by itself (for interactive programming).

The example script lists the countries with population above 100 million along with their population (in thousands), ordering them by the name of the countries.
The script starts by first selecting only those records from the data set that has population above 100 million. The script then sorts this data by the name of the country.

We can see that in many ways these individual steps are similar to those performed during query evaluation, except that the onus is on the programmer to figure out the order of execution of the steps instead of the data management system performing an optimized order.

The script will take a couple of minutes to run and produce a lot of messages. At the end, you should be able to see an output like this (truncated for brevity).

```
...
(Japan,127749)
(Mexico,127540)
...
```

Turn in:- Nothing.

[1] https://covid19.who.int/info/
[2] https://www.who.int/data/gho/data/indicators/indicator-details/GHO/population-(in-thousands)
[3] https://www.who.int/

Ex. 4 — PigLatin - Vaccination Across Various Regions (3 Points)

Write a PigLatin script such that, for each WHO region, output the region, number of countries and the total number of vaccinated people in those countries. Order the output by the region. The name of the script you turn in should be **Q4.pig**.
Once you have the script completed and is satisfied with its output, execute it the following way (for submission purposes).

```
$ pig Q4.pig > Q4.log 2>&1
$
```

The result part of your script's output should follow the example format below as-is (truncated for brevity).

```
...
(BES,4,)
(EMRO,34,72340623)
...
```

Make sure that the log file also contains the various information that pig has been producing and not just the final results. If those log information from pig is missing, points will be deducted.
Turn in:- Q4.pig and **Q4.log**

Ex. 5 — PigLatin - Vaccine Suppliers (5 Points)

Write a PigLatin script that will list the companies whose vaccines are used across most number of countries. Output the company name and the number of countries they supply to. Order the output with the companies that supply most countries on top. Restrict the output only to top 10 records. The name of the script you turn in should be **Q5.pig**. Similar to previous exercise, also produce **Q5.log**. The result part of your script's output should follow the example format below as-is (truncated for brevity).

```
...
(Moderna,65)
(Janssen Pharmaceuticals,32)
...
```

Turn in:- Q5.pig and **Q5.log**.

Ex. 6 — PigLatin - Vaccination Rate and Vaccine Usage (8 Points)

Write a PigLatin script (**Q6.pig**) that does the following:
For countries with population above 100 million, list each country, its population, the percentage of fully vaccinated people and the number of vaccine brands used in that country. Order the output by the decreasing order of their population. Similar to previous exercise, also produce **Q6.log**.
What does the schema look like immediately after you perform the `GROUP` operation step? Include this under a section **Q6** in your **assignment3.pdf**.
The result part of your script's output should follow the example format below as-is (truncated for brevity).

```
...
(France,64721,65.91931521456,5)
(Italy,59430,68.6257445734,5)
...
```

Turn in:- Q6.pig and **Q6.log** (and the contents in **assignment3.pdf**).

Guidelines

NO Handwritten / scanned submissions are accepted for this assignment.

MapReduce

This discussion is pertaining to Exercises Ex.1 and Ex.2.

- We define the term "browser" to also to include its software version. I.e., `Mozilla/4.08` is considered a different browser to `Mozilla/4.2` although the two only differs in version numbers.
- Your solutions should have only `Mapper` and `Reducer` functions. **DO NOT** use the `Combine` functionality.
- When implementing a solution, remember that in some cases you may need more than one MapReduce job to accomplish a task (Output of one MapReduce's `Reducer` forms the input of another's `Mapper`).
- Each MapReduce step goes through the disk in order to pass data to the next step in the process. Therefore, come up with a solution that will reduce the number of MapReduce jobs required as well as reduce the amount of data that will have to flow from one part of the MapReduce process to the next one (think of some of the simple concepts we applied for query evaluation).
- You can follow the pseudo-code syntax as was shown in class. Our primary interest is to see if you know how to design the workflows to pick the right key/value for the input/output of mappers and reducers and have an understanding of the internal logic that you should put inside these functions. Please do not write Java code, etc.

Pig Latin

This discussion is pertaining to Exercises Ex.3 through Ex.6.

- Watch the tutorial (remember to use the server name **winter2022-comp421.cs.mcgill.ca**) and try to do the statements along side (type it by yourself). This is a good way to get warmed up to the Pig Latin statements before you tackle the assignment problems.
- Your code does not have to be optimized.
- Remember, the `DESCRIBE` command can be very handy to figure out if the schema of your data set is changing as it goes through some of the complex steps (such as `JOIN` and `GROUP`). Trying to access attributes incorrectly may result in pig throwing errors that can be confusing and frustrating. So use this command to investigate it. You can leave the `DESCRIBE` commands in your submission scripts if you would like to or comment them off, once their purpose is served. They are locally interpreted by the pig client and therefore has no significant overhead.
- If you are debugging, it is recommended to run pig interactively rather than using a script. That way, you can adjust the logic as required, run `DESCRIBE` commands, etc., without having to start the execution from scratch as executing a workflow end-to-end can take a few minutes.
- Use the `DUMP` command in the intermediate steps to help you debug to see if you got the logic correct up to there. You can comment it out once your are done with your work. Your final submission should have only one `DUMP` command for your final result. Remember! every `DUMP` command triggers the execution of the MapReduce framework. Verify the log files that you are submitting to ensure that it clearly displays the final (expected) results.
- You must not use the `STORE` command to store anything into the HDFS. Your scripts can also run into problems if it encounters results already stored in the HDFS.
- The example outputs given above are not based on actual solution outputs. So do not try to reproduce those values.The examples are there to show the expected formats. Operations such as `GROUP` can have impact on the data format. It is important that you learn what happens in those circumstances and how you can transform the data back to the required format.
- Your final output (results) format must not be nested, but flat (as shown in the examples.). Below is an example of a format that is not acceptable.
  `(A,(B,C))`
  Instead, it should be of the following format.
  `(A,B,C)`
  Explore the `FLATTEN` command if you have trouble addressing this. Remember that some data has brackets in the data itself. This is ok. e.g - `Bolivia (Plurinational State of)`.

What to turn in

assignment3.pdf that contains the two MapReduce pseudo-code solutions as well as the extra information requested for the Pig Latin questions. The Pig Latin scripts and log files themselves: **Q4.pig**, **Q4.log**, **Q5.pig**, **Q5.log**, **Q6.pig** and **Q6.log**. Ensure that your log files have any information that pig has been producing (not just the results) and most importantly, also the intended final result.

These are the only acceptable file formats.

You may `tar` or `zip` your submission if you have to, but make sure that you verify your submission contains all the files and they are correct. There will be no accommodation for submitting incorrect files.

Questions ?

Please use **Ed** for any clarifications you need (`assignments` → `a3`). Do not email the instructor or TAs as this leads to a lot of duplicate questions and responses (not an efficient system). Such emails will not receive any replies.

Please check the pinned post "A3 general clarifications" in Ed before you post a new question. It might have been already addressed there, in which case we will not address it again.

Questions about general clarifications must be marked public (as other students will also benefit from this and may even have a valid response). TAs and Instructors upon their discretion may toggle any private posts into public mode for the benefit of the student population at large.

There will be specific office hours for the assignment that will be announced closer to the due date.

Extensions and Late submissions

- Remember, your submission is **due on April 7th 12:00 noon**. There is no place for excuses.
- A maximum of 1 day of late submission is allowed with a penalty of 20% of the achieved grade per day (rounded up, even for a minute).
- Penalty waivers are granted only for medically documented emergencies and under any circumstances **will not be granted unless requested 24 hours before the due**. I expect you to be better organized and get the job done ahead and leave the last 24 hours only for one last final check.
- There is no "partial late penalty" concept, it is applied to your entire submission, and not just specific questions that you submitted late.