CSE557: BDA Assignment 2

Aim: Hands on with Apache Spark **Total Points:** 100

Deadline: 9 March 2021; 2359

Instructions:

1. Mention all assumptions if any in the report.
2. Report and code should be submitted in the classroom in a zip folder with the name 'A2_RollNumber_RollNumber.zip'.
3. 1 Member will submit the zip on Google Classroom, second will mark "Turn in".
4. You are free to use any library or data processing techniques.
5. There will be demo's for assignments. So write the code on your own, make sure you don't cheat. If you can't answer the questions during your demo, 50% of your marks will be deducted.

The following should be included in the Report:

1. Explain your methodology: approach and reason clearly in the report.
2. Add all data analysis steps which you have performed on the dataset.
3. Report the output screenshots for all queries.
4. Make a section "Learning", which describes your learning in doing this assignment.

Tasks:

1. Import all the data into postgres or the relational database used in the first assignment. Now use Apache Spark to answer all the queries with Postgres as the backend storage engine. You may use any technology for connecting Postgres with Apache Spark. One such method is to use JDBC. You may like to read the blog at (https://zheguang.github.io/blog/systems/2019/02/16/connect-spark-to-postgres.html). Write modules in spark to answer all the queries in assignment 1.
2. Use MongoDB as storage engine (setup of Mongo DB :https://hevodata.com/blog/install-mongodb-on-ubuntu/). Import the data to MongoDB and connect Spark to mongoDB to answer queries in assignment 1.
3. Now use HDFS to store the data and use apache spark to read from HDFS and perform the queries needed to answer the questions in assignment 1.
4. Execute commands for all parts in two settings:

• Setting 1: with only one executor
• Setting 2: with two executors

Report your average execution time (over some runs of the commands) for each part in both settings.

NOTE:

- There are two ways you can compute your queries: 1) Import the data in the main memory data structure (RDD) and evaluate your queries, 2) Directly execute your queries on the backend if it is some database engine at the backend. Include the merits and demerits of each approach in your report. Include some empirical results using the queries to substantiate the claims.
- Please go through **comments section** in assignment 1 to clear your doubts about queries and ask doubts not covered in that
- Please make sure your code is modular.
- In your report include details on how you stored the data in the respective storage variants and also the challenges you faced in integrating Spark with various storage mediums.

The queries are specified below for completion; they are the same as assignment 1.

Patents data

patents_only(author_id,code,title_localized) - patents_only.csv,
authors(id, country,affiliation,inventor,count_of_patents) - patent_inventors_data.csv.
Note that the count_of_patents only includes the count of distinct patents published. It does not count the reissue of patents.
subclass(id,title) - subclass_current.csv.

You would have to join the tables for answering some queries Points (70) + 30 for report & presentation

1. Find the total number of patents published by all authors. **(Points:5)**
2. Find the affiliation of the author with the maximum number of patents. **(Points: 10)**

3. Find the author who has filed the maximum number of patents under the patent code with the prefix “707/999”. (Hint: join patent_inventors_data and patents_only on author_id). **(Points:15)**
4. Output the top 5 patents that have been reissued in descending order of the number of times they have been reissued (hint the repetition of patent name with different codes indicates it has been reissued). **(Points:10)**
5. Output the names of the top 5 companies in descending order of the number of patents filed. **(Points:5)**
6. Find the total number of patents filed by authors at Google and Apple. **(Points:10)**
7. Find the patent category name that has the maximum number of patents filed under it. (Hint: Use data in subclass_current.csv in addition to other two data files). **(Points:15)**

Publications data:

You would have to join the 2 tables for answering some queries **Points(70)+30 for report and presentation**

1. Report the total number of authors. **(Points:5)**
2. Report the name of the author with the highest h index and the corresponding index (index column in authors_aminer) value. Also report the author with the highest number of papers (hint: for number of papers use number_of_papers column in authors_aminer). **(Points:12)**
3. Report the number of authors and their names who have “data accuracy” as one of the research interests. **(Points:10)**
4. Report the total number of publications in the venue “IEEE Transactions on Parallel and Distributed Systems”. (**Points: 13)**
5. Report the top 5 author names sorted by their number of publications in the year 2016. (hint : join both tables. Do not consider number_of_papers column in authors_aminer).**(Points:10)**
6. Report the paper with the highest number of citations and the corresponding venue in the year 2015. (hint: use only publications_aminer data). (**Points: 15)**
7. Report the top 5 venues sorted by number of publications in that venue. (**Points:5)**

Pull requests data **Points(70)+30 for report and presentation**

1. Report the number of pull requests
    a. "opened" per day.
    b. "Discussed" per day. **(Points: 10)**
2. Output the person who has the highest number of comments per month. Assume a month has 30 days. The comments refer to events of the type "discussed". **(Points:10)**
3. Output the person who has the highest number of comments per week. Assume a week has 7 days. The comments refer to events of the type "discussed". **(Points: 10)**
4. Output the number of Pull Requests (PRs) **opened** per week. **(Points:10)**
5. Count the number of Pull Requests **merged** per month in the year 2010. **(Points:10)**
6. Per day report the total number of events. **(Points: 5)**
7. Report the user who has opened the highest number of PRs in the year 2011. **(Points: 15)**