Homework 2: Map-reduce, Hadoop and Spark

Form:	Theoretical: Written report (word) Practical: iPython
Language:	Theoretical: Pseudo-code Practical: iPython file with pySpark
Requirements:	-
Submission:	Course website - moodle
Contact:	tamirmendel@mail.tau.ac.il
Deadline for submission:	**June 26, 2020**

Students will form teams of two or three people each, and submit a single homework for each team. The same score for the homework will be given to each member of the team. **Please write all group members ids in the PDF and iPython notebook.**

Theoretical

Scenario: A company has two csv files that contain users and transactions details:

Users (id, email, language, location)

Transactions (transaction_id, product_id, user_id, purchase_quantity, item_description)

You are asked to provide the pseudo-code to the following relational algebra operations to map-reduce.:

1) Projection: return the location field

2) Selection: return users from a given location

3) Semi-Join: return users detail with at least one transaction

4) Aggregation: count distinct product purchases for each user (include users without purchases).

A description of each operator can be found at:
http://en.wikipedia.org/wiki/Relational_algebra#Projection_.28.CF.80.29

Note:

*Assume the combiner is not available. Try to reduce data transfers.

*For each one of the four operations, you need to provide the pseudo-code for the map and reduce functions (similarly to the map and reduce functions that were presented in the word-count example).

Practical

Scenario: Your team is in charge of analyzing the traffic of a company's websites. You are given the log files collected at the server side (and already processed by another division) and you need to present reports. You must summarize the traffic of one of the websites, with respect of the users that visited it by different countries.

In this exercise, you will be given data from one data source. You will load the data to HDFS and analyze it.

All the operations must be implemented in PySpark and you should maximize the use of distributed computation.

Data Sources:

Microsoft-com.data: The data was collected by sampling and processing the www.microsoft.com logs. The data records the use of www.microsoft.com by approximately 30,000 users. The data lists all the areas of the web site that a user visited in a one week timeframe (from 21-September-1998 to 27-September-1998).

Analysis of traffic from single data source:

You are asked to provide a report on the usage of the Microsoft website from different parts of the world:

1. Load the Microsoft.com data into HDFS.
2. Parse the data according to the format specified in the *.info* file.
3. Given the list of countries in the file *countries.txt* (e.g., South Africa, Spain, Sweden, Switzerland), filter the data to include only users that visited at least one page related to one of the given countries.
4. Build a report that computes:
   a. For each country, the number of users that visited a page of that country.
   b. The top 5 visited countries.
5. Write the report (4.a) to HDFS for future use. The file must be formatted as *country:number_of_users*.

Prepare a report with statistics by using MapReduce. Statistics on the data to generate:

6. The average number of pages per user.
7. The page id with maximum number of visits.

Finally, write a command(s) that remove your directories from HDFS.

Files:

You will work on data that was taken from a real data source. However, the data has been adapted to the needs of this course, so, please, do not distribute them:

microsoft-com.data.zip:

- microsoft-com.data
- microsoft-com.info
- countries.txt

Evaluation

We will run your code on an extended version of the source data (with the same format, but more rows). Special attention will be given to the efficient implementation of the different parts and the use of distributed computing strategies, so you should make sure to include in the notebook also a description of your functions and motivations of your implementation choices (especially, but not limited to, cases when you decide to not use distributed computation).

Good Luck!