CLOUD COMPUTING

Practical 05: Pig Latin Script

In the world of Big data, it is crucial to be able to process the data quickly and efficiently. Processing data quickly means one should use massive-parallelism, and processing data efficiently means the execution of processes should be fault-tolerant and linearly scalable. The Hadoop ecosystem is an Open Source set of frameworks designed around this concept. Through its components, the Hadoop ecosystem enables developers to focus on solving their Big Data problems rather than developing ad hoc solutions to managing massive data, as in Big Data applications. Pig is an analysis platform which provides a dataflow language called Pig Latin. In this practical we will model and write some scripts in Pig Latin to perform some data analysis.

Exercise 1

Suppose that we have a large data log about user's movie preferences. The file meta-data contains the following information (*rate_movie, completed_movie, pause_movie, start_movie, brose_movie, list_movie, search_movie, login, logout, incomplete_movie, purchase_movie, …*). The log file sample is called "Movie_Log".

1. How can I print the results in Pig Latin script?
   Use the primitive DUMP <bag of tuples>
2. Print the log file sample grouped by movie.
   Log_sample = LOAD 'movie_log_file' USING PigStorage();
   Movie_logs = GROUP log_sample BY movieID;
   DUMP log_sample
3. What is the primitive "DESCRIBE" in Pig Latin?

4. Extend the previous script to process the clickstream data into user sessions.
   ;

5. How can I use FOREACH statement in Pig Script?
6. Select only the clicks which correspond to starting, browsing, completing, or purchasing movies.

Exercise 2

Let "students.csv" is a file that contains students' data. We assume that the data values are separated by "comma".

1. Create a pig script to load students.csv data.

2. Create a pig script to filter out the first row of the data.

3. Write a pig script to assign names to the data fields of the students.csv data. The output file should be called "students_details".

Assume that we have another file recording the students' attendance; "students_attendance.csv".

4. Perform the previous 3 operations on the file "students_attendance.csv". The output of the 3rd operation in this case should be called "SA_details".

5. Extend your script, if necessary, to filter the data (all hours attended for each student).

6. Write a script to find the sum of hours attended by each student.

7. Write a script to join StudentID, Name, hours attended.
8. Print the results on the screen.

Submission Instructions

All submissions must be done via Moodle. The submission deadline is Friday November 16th at 23:00. Your submission should consist of one file (MS Word or PDFs), which contains the answers to the above questions. The submitted file should be named following the format:

COMP*xxxxx_Surname_FirstName_StudentNo_* ***Pracitcal****05.*{doc, pdf}

Example: *(**COMP**41110_Smith_John_12345 _**Practical**05.pdf)*