Big Data: Architectures and Data Analytics

Month Day, Year

Student ID _____

First Name _____

Last Name _____

The exam is **open book** and lasts **2 hours**.

Part I

Answer to the following questions. There is only one right answer for each question.

1. (2 points) Consider the HDFS file log.txt. The size of log.txt is 260MB and the block size is set to 128MB. Suppose to execute the word count application, based on MapReduce, on log.txt. What is the maximum number of mappers that can be executed in parallel?

   a) 260

   b) 128

   c) 2

   d) 3

2. (2 points) Consider the HDFS file words.txt. The size of words.txt is 512MB and the block size is set to 256MB. The number of distinct words appearing in words.txt is 253. Suppose to execute the word count application, based on MapReduce, on words.txt. Which one of the following statements is true?

   a) Setting the number of reducers to a value greater than 253 is useless

   b) The number of reducers is automatically set to 2 by Hadoop

   c) The number of reducers must be set to 256

   d) The number of reducers must be set to 512

Part II

The PoliMovie company, which provides a movie on-demand service, is interested in analyzing some of its data to characterize its movies and customers.

The analyses are based on the following data sets/files.

- WatchedMovies.txt
  - WathcedMovies.txt is a textual file containing the log of the list of movies watched by the users of the movie on-demand service
  - Every time a user watches a movie a new line is inserted in the WatchedMovies.txt file, i.e., each line of the file contains the information about one visualization
  - Each line of the file has the following format
    - userid,movieid,start-timestamp,end-timestamp
    - Where userid is a user identifier, movieid is a movie identifier, start-timestamp is the time at which userid started watching movieid, and end-timestamp is the time at which the visualization ended.
    - For example, the line

      user1,movie235,20160506_10:15,20160506_12:11

    - means that *user1* watched *movie235* from *10:15 of May 6, 2016* to *12:11 of May 6, 2016*

- UsersPreferencesProfile.txt
  - UsersPreferencesProfile.txt is a textual file containing the list of preferences (liked genres) for each user. Specifically, for each user, it contains the information about the genres the user likes.
  - Each line of the file contains the information about one preference (one like) of a user. Each user can occur in multiple lines.
  - Each line of the file has the following format
    - userid,movie-genre
    - The meaning of each line of the file is the following:
      - The user with id userid likes the movies of type movie-genre
    - For example, the two lines

      user1,adventure
      user1,horror

    - mean that *user1* likes the *adventure* and *horror* genres

- Movies.txt
  - Movies.txt is a textual file containing the list of available movies with the associated characteristics
  - The file contains one single line for each movie
  - Each line of the file has the following format
    - movieid,title,movie-genre
    - Where, movied is the identifier of a movie, title is its title, and movie-genre is the genre of the movie. Each movie is characterized by one single genre.
    - For example, the line

      movie235,Harry Potter II,Fantasy

    - means that the title of *movie235* is *Harry Potter II* and the genre of *movie235* is *Fantasy*.

Exercise 1 (10 points)

The management of PoliMovie is interested in identifying the most watched movie. The most watched movie is defined as the one that has been watched more times according to the content of WatchedMovies.txt. However, only the visualizations with a duration greater than 10 minutes must be considered.

Design a single application, based on MapReduce and Hadoop, and write the corresponding Java code, to address the following point:

A. *Select the most watched movie*. Specifically, the application must select the most watched movie, according to the definition provided above, and store the movieid of the selected movie in an HDFS folder. The name of the output folder is one argument of the application.

   The input of the application is file WathcedMovies.txt, which is one argument of the application.

To compute the duration of a visualization use the static method *int computeDuration(String startTime, String endTime)* of the class *BigDataTime*. The method returns the difference between *endTime* and *startTime* in minutes. Suppose that the class *BigDataTime* and its method have been already implemented by someone else.

Exercise 2 (17 points)

The management of PoliMovie is interested in identifying the users with a “misleading profile” or a “useless profile”. A user is characterized by a *misleading profile* if more than threshold% of the movies he/she watched are not associated with a genre he/she likes. Differently, a user is characterized by a *useless profile* if he/she likes more than 90% of the possible genres. The number of possible genres is equal to the number of distinct genres occurring in Movies.txt.

The managers of PoliMovie asked you to develop an application to address the analyses they are interested in.

The inputs of the application are the files WatchedMovies.txt, UsersPreferencesProfile.txt, and Movies.txt, which are three arguments of the application, and the value of the threshold that is used to decide if a profile is misleading (also this value is an argument of the application).

Specifically, design a single application, based on Spark and RDDs, and write the corresponding Java code, to address the following points:

A. *Identify the users with a useless profile*. Specifically, the application must select the userids of the users with a useless profile and store the userids of these users in an HDFS folder. The name of the output folder is one argument of the application.

B. *Identify the users with a misleading profile*. Specifically, the application must select the userids of the users with a misleading profile and store the userids of these users in an HDFS folder. The name of the output folder is one argument of the application.