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. The repetition factor is set to 4. What is the total number of blocks that is used to store log.txt in the HDFS file system?

   a) 260

   b) 128

   c) 12

   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) The number of reducers must be set to a value less than or equal to 253

   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 PoliEnv agency is interested in analyzing the pollution of a large city. They use a set of sensors, located in the city, to gather the daily value of the PM10 pollutant. Moreover, the sensors measure also the average daily temperature.

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

- PM10Readings.txt
  - PM10Readings.txt is a textual file containing one daily PM10 value for each sensor (i.e., one daily reading for each sensor)
  - Each line of the file has the following format
    - sensorid,date,PM10
    - Where sensorid is a sensor identifier, date is the date of the measurement, and PM10 is the measured PM10 value.
    - For example, the line

      sensor#1,01-10-2014,15.3

    - means that on *January, 10 2014, sensor#1* measured a PM10 value equal to *15.3*

Exercise 1 (9 points)

The management of PoliEnv is interested in identifying potential noise in the PM10Readings.txt. Specifically, a PM10 reading (i.e., a line of PM10Readings.txt) is considered a *noise* reading if the value is greater than 45 or less than 0.

For instance, the following two lines are noise readings:

sensor#1,01-10-2014,50

sensor#1,01-11-2014,-1

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

A. *Select noise readings from the PM10 data set*. The application must select the lines of PM10Readings.txt associated with a noise reading/noise value and store them in an HDFS folder. The name of the output folder is one argument of the application.

   The input of the application is PM10Readings.txt, which is one argument of the application.

Exercise 2 (18 points)

The management of PoliEnv is interested in performing some analyses about the correlation between the month and the number of days with a critical PM10 value and some analyses about the time periods associated with a high number of days with a *critical value* of PM10. A PM10 value is *critical* if it is greater than the PM10 threshold, that is one input of the application.

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

The input of the application is the file PM10Readings.txt, which is one argument of the application, and the value of the PM10 threshold that is used to decide if a PM10 value is critical (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. *Number of critical days associated for each pair (sensor, month)*. Compute, for each pair (sensorid, month), the number of days with a critical PM10 value and store the result in an HDFS folder. Each line of the output file contains a pair (month, number of critical days for that month). The name of the output folder is an argument of the application. Pay attention that January 2013 and January 2014 are two different months, i.e., a month is identified by the pair month name, year.

B. *Identify critical periods for each sensor*. If the PM10 value measured by a sensor is critical for three consecutive days, then that set of days is considered a critical time period for that sensor. For instance, suppose than the value of PM10 measured by sensor #3 is critical in the dates 01-10-2014, 01-11-2014, 01-12-2014. Hence, the time period (01-10-2014 - 01-12-2014) is a critical time period associated with sensor #3. The application must select all the time periods, composed of three consecutive days, that are critical periods and the associated sensor. Store the selected time periods, and the associated sensorid, in an HDFS folder. The name of the output folder is one argument of the application. Each line of the output files has the format: first day of the critical time period_sensorid. For instance, based on the example reported before, one line of the output files will be 01-10-2014_sensor#3.

Suppose that someone has already implemented the static method *String dateOffset(String dateValue, int deltaDays)* of the *DateTool* class. The method receives in input a date (*dateValue*), in the format monthname-day-year, and an integer representing a delta in terms of number of days (*deltaDays*). The returned value is a string associated with the date dateValue+deltaDays. For example, the following invocation of the dateOffset method returns the string “01-09-2014”:

String newDate=DateTool.dateOffset(“01-10-2014”, -1);