Big Data: Architectures and Data Analytics

July 12, 2016

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 2560MB. Suppose that you are using an Hadoop cluster that can potentially run up to 10 mappers in parallel and suppose to execute the word count application, based on MapReduce, by specifying log.txt as input file. Which of the following values is a proper HDFS block size if you want to "force" Hadoop to run 10 mappers in parallel when you execute the word count application by specifying log.txt as input file?

   a) Block size: 256MB

   b) Block size: 1024MB

   c) Block size: 2048MB

   d) Block size: 2560MB

2. (2 points) Consider the following driver of a Spark application.

```
package …
import ….
public class SparkDriver {
        public static void main(String[] args) {
                SparkConf conf=new SparkConf().setAppName("Test");
                JavaSparkContext sc = new JavaSparkContext(conf);
                JavaRDD<String> linesRDD = sc.textFile("input.txt");
                JavaRDD<String> selectedLinesRDD = linesRDD.filter(new Filter());
                JavaRDD<String> diffRDD = linesRDD.subtract(selectedLinesRDD);
                diffRDD.saveAsTextFile("outputFolder/");
```

```
                Long numElements = linesRDD.count();
                System.out.println(numElements);

                String firstValue = linesRDD.first();
                System.out.println(firstValue);

                sc.close();
        }
}
```

Which one of the following statements is true?

a) Caching the RDD linesRDD can improve the efficiency of the application (in terms of execution time)

b) JavaRDD can never be cached

c) Caching the RDD linesRDD is useless

d) All the other answers are wrong

Part II

PoliBikes is a bicycle sharing service with stations located in many cities of Italy. The managers of PoliBikes are interested in analyzing the use of their service. The analyses are based on the following data sets/files.

- StationsOccupancy.txt
  - StationsOccupancy.txt is a textual file containing the historical information about the number of bicycles and free slots for each station.
  - The sampling rate is 10 minutes (i.e., every 10 minutes the status of the stations is sampled and a new line for each station is inserted in StationsOccupancy.txt)
  - Each line of the file has the following format
    - stationId,date,hour,minute,numAvailableBicycles,numFreeSlots

      Where *stationId* is a station identifier, *numAvailableBicycles* is the number of available bicycles at time *date-hour-minute*, and *numFreeSlots* is the number of free slots at time *date-hour-minute*.

    - For example, the line

      station1,2016/05/06,23,10,3,12

      means that there were 3 available bicycles and 12 free slots at ***station1*** on ***May 6, 2016*** at ***23:10***
- Stations.txt
  - Stations.txt is a textual file containing the list of available stations with the associated characteristics
  - The file contains one single line for each station of the bicycle sharing system
  - Each line of the file has the following format
    - stationId,stationName,zone,city,totalNumberOfSlots

      Where *stationId* is the identifier of the station, *stationName* is its name, *zone* is the city zone in which the station is located, *city* is the city in which the station is located, and *totalNumberOfSlots* is the number of slots of the station (i.e., its size).

    - For example, the line

      station1,Politecnico,ZoneA,Turin,15

      means that the name of ***station1*** is "***Politecnico****",* the station is located in ***ZoneA*** of ***Turin***, and the number of slots of station1 is ***15***.

Exercise 1 – MapReduce and Hadoop (10 points)

The managers of PoliBikes are interested in counting the number of large stations for each zone of Turin. A station is a ***large station*** if it has at least 20 slots (*totalNumberOfSlots*) based on the information stored in Stations.txt.

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

A. *Count the number of large stations for each zone of Turin.* Specifically, the application must count for each zone of Turin the number of large stations based on the definition reported above and store the result in an HDFS folder. Each line of the output file has the format

   zone,number of large stations of this zone

   The name of the output folder is one argument of the application. The other argument is the path of the input file Stations.txt.

Exercise 2 – Spark and RDDs (17 points)

The management of PoliBikes is interested in identifying potential critical small stations. A station is classified as a "***small station***" if it has less than 5 slots (*totalNumberOfSlots*) based on the information stored in Stations.txt. A station is considered a "***potential critical station***" if it has been "full" at least one time (based on the historical data stored in StationsOccupancy.txt). A station has been full at least one time if there is at least one line in StationsOccupancy.txt for that station with the number of free slots equal to 0. Pay attention that a station is a "***potential critical small station***" if it satisfies the two definitions reported above (i.e., it is a small station and also a critical station).

The management of PoliBikes is also interested in identifying the well-sized stations. A station is classified as a "***well-sized station***" if it has always had at least 3 free slots (based on the historical data stored in StationsOccupancy.txt).

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

The inputs of the application are the files Stations.txt and StationsOccupancy.txt and two output folders (associated with the outputs of points A and B of this exercise). Inputs and outputs are specified as arguments 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. *Select the list of potential critical small stations*. Specifically, the application must select the stationIds of the "*potential critical small*" stations, based on the definitions reported above, and store the stationIds of the selected stations in an HDFS folder. The name of the output folder is one argument of the application.

B. *Select the list of well-sized stations*. Specifically, the application must select the stationIds of the well-sized stations based on the definition reported above, and store the selected stationIds in an HDFS folder. The name of the output folder is one argument of the application.