Big Data: Architectures and Data Analytics

June 30, 2021

Student ID _____

First Name _____

Last Name _____

Part I

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

1. (2 points) Consider the following Spark application.

```
package it.polito.bigdata.spark.exam;

import ….;

public class SparkDriver {

    public static void main(String[] args) {

        // Create a configuration object and set the name of the application
        SparkConf conf = new SparkConf().setAppName("Spark Code");

        // Create a Spark Context object
        JavaSparkContext sc = new JavaSparkContext(conf);

        // Read a first input file and analyze its content
        JavaRDD<String> logs1RDD = sc.textFile("logs1.txt");
        long numLinesL1 = logs1RDD.count();
        System.out.println(numLinesL1);

        // Read a second input file and analyze its content
        JavaRDD<String> logs2RDD = sc.textFile("logs2.txt");
        long numLinesL2 = logs2RDD.count();
        System.out.println(numLinesL2);

        // Analyze both file together
         JavaRDD<String> unionLogsRDD = logs1RDD.union(logs2RDD);

        // Remove duplicates
         JavaRDD<String> distinctElementsRDD = unionLogsRDD.distinct();

        long numDistinctElements = distinctElementsRDD.count();
        System.out.println(numDistinctElements);
```

```
        // Close the Spark context
        sc.close();
    }
}
```

Which one of the following statements is **true**?

a) Caching both logs1RDD and logs2RDD can improve the efficiency of the application (in terms of execution time)

b) Caching unionLogsRDD can improve the efficiency of the application (in terms of execution time)

c) Caching distinctElementsRDD can improve the efficiency of the application (in terms of execution time)

d) Caching the RDDs of this Spark application does not improve its efficiency (in terms of execution time)

2. (2 points) Consider the input HDFS folder *myFolder* that contains the following two files:

- ItalianCities.txt
    - the text file ItalianCities.txt contains the following three lines
      Turin,Italy
      Rome,Italy
      Turin,Italy
- FrenchCities.txt
    - the text file FrenchCities.txt contains the following two lines
      Nice,France
      Paris,France

Suppose that you are using a Hadoop cluster that can potentially run up to 10 instances of the mapper class in parallel. Suppose the HDFS block size is 128MB. Suppose to execute a MapReduce application for Hadoop that analyzes the content of myFolder. Suppose the map phase emits overall the following 5 key-value pairs (the key part is a city and the value part is always NullWritable):

(“Turin”, NullWritable)
(“Rome”, NullWritable)
(“Turin”, NullWritable)
(“Nice”, NullWritable)
(“Paris”, NullWritable)

Suppose the number of instances of the reducer class is set to 1 and suppose the reduce method of the reducer class emits a pair (key, NullWritable) for each distinct key. Suppose the following 4 pairs are overall emitted by the reduce phase:

("Turin", NullWritable)
("Rome", NullWritable)
("Nice", NullWritable)
("Paris", NullWritable)

Considering the single instance of the reducer class, overall, how many times is the ***reduce method*** invoked?

a) 1

b) 2

c) 4

d) 5

Part II

PoliBikes is an international association that monitors the sales of motorbikes around the world. PoliBikes tracks motorbike sales independently of the manufacturer. The analyses performed by PoliBikes are based on the following input data sets/files.

- BikeModels.txt
  - BikeModels.txt is a text file containing the list of motorbike models produced around the world. Each line of BikeModels.txt is associated with one motorbike model. The number of distinct motorbike models is large.
  - Each line of BikeModels.txt has the following format
    - ModelID,MName,Manufacturer

      where ModelID is the unique motorbike model identifier while MName and Manufacturer are the name and the manufacturer of the motorbike model, respectively.

    - For example, the following line

      Model10,CBR125,Honda

      means that the name of the motorbike model identified by ***Model10*** is ***CBR125*** and its manufacturer is ***Honda***.

- Sales.txt
  - Sales.txt is a text file containing information about all sales around the world. One line for each sale is stored in Sales.txt. This file contains the sales of the last 50 years.
  - Each line of Sales.txt has the following format
    - SID,BikeID,ModelID,Date,Country,Price,EU

      where *SID* is the unique identifier of the sale while *BikeID* is the identifier of the motorbike that was sold and *ModelID* is its model, *Date* is the date of sale, *Country* is the country where the motorbike was sold, *Price* is the sale price, and *EU* is a string attribute that specifies if the Country is a European country or an Extra-European country (EU="T" means European country while EU="F" means Extra-European country). The Date format is "YYYY/MM/DD". Note that the same motorbike can be sold multiple times. Hence, the same BikeID can occur in different lines associated with different sales.

    - For example, the following line

      SID10,BikeIDFT2011,Model10,1985/05/02,Italy,5900,T

      means that the sale identified by ***SID10*** is related to the sale of the motorbike identified by the code ***BikeIDFT2011***. Motorbike

BikeIDFT2011 was sold on ***May 2, 1985*** in ***Italy*** at ***5900*** euro. EU is equal to "T". Hence, this sale is a European sale. The model of the sold motorbike is ***Model10****.*

Exercise 1 – MapReduce and Hadoop (7 points)

The managers of PoliBikes are interested in performing some analyses about their motorbike models.

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

1. *Motorbike models that were sold frequently both in Europe and outside Europe in the year 2020.* The application considers only the sales related to the year 2020 and selects the identifier (ModelID) of the motorbike models that were sold at least 10000 times in Europe (EU="T") and at least 10000 times outside Europe (EU="F"). Store the identifiers (ModelID) of the selected motorbike models in the output HDFS folder (one of the selected ModelID per line).

Suppose that the input of this application is Sales.txt and it has been already set and also the name of the output folder has been already set.

- Write only the content of the Mapper and Reducer classes (map and reduce methods. setup and cleanup if needed). The content of the Driver must not be reported.

- Use the next two specific multiple-choice questions to specify the number of instances of the reducer class for each job.

- If your application is based on two jobs, specify which methods are associated with the first job and which are associated with the second job.

- If you need personalized classes report for each of them:

    - name of the class

    - attributes/fields of the class (data type and name)

    - personalized methods (if any), e.g., the content of the toString() method if you override it

    - do not report get and set methods. I suppose they are "automatically defined"

Exercise 1.1 - Number of instances of the reducer - Job 1 - MapReduce and Hadoop (0.5 points)

Select the number of instances of the reducer class of the first Job

(a) 0

(b) exactly 1

(c) any number >=1

Exercise 1.2 - Number of instances of the reducer - Job 2 - MapReduce and Hadoop (0.5 points)

Select the number of instances of the reducer class of the second Job

(a) One single job is needed for this MapReduce application

(b) 0

(c) exactly 1

(d) any number >=1

Exercise 2 – Spark and RDDs (19 points)

The managers of PoliBikes are interested in performing more analyses related to sales.

The managers of PoliBikes asked you to develop one single application to address all the analyses they are interested in. The application has four arguments: the two input files BikeModels.txt and Sales.txt and two output folders, "outPart1/" and "outPart2/", which are associated with the outputs of the following Points 1 and 2, respectively.

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

1. *Motorbike models with a "large" variation in terms of sale price in the year 2020.* The application considers only the European sales (EU="T") related to the year 2020 and selects the motorbike models with a sale price variation greater than 5000 euro. For each motorbike model, its sale price variation in the year 2020 is given by the difference between the maximum sale price of all sales related to that motorbike model in the year 2020 and the minimum sale price of all sales related to the same motorbike model in the year 2020. The identifiers (ModelIDs) of the selected motorbike models are stored in the first HDFS output folder (one ModelID of the selected motorbike models per line).

2. *Manufacturers with many unsold or infrequently sold models.* The application considers all sales and selects the manufacturers that are associated with many unsold or infrequently sold models. A motorbike model is categorized as an unsold model if it has never been sold. A motorbike model is categorized as an infrequently sold model if it has been sold at least one time but at most 10 times. Specifically, the application selects a manufacturer if (globally) at least 15 of its models are unsold or infrequently sold models. The manufactures that satisfy the reported constraint are stored in the second output folder. For each of the selected manufactures, the following information is stored in the second output folder: manufacturer, number of unsold models+number of infrequently sold models (one of the selected manufactures and the associated information per output line).

   Some examples related to Point 2 (second part of the exercise)

   - Suppose that the manufacturer Honda has 4 unsold models and 12 infrequently sold models. Honda **is selected** because, globally, 16 of its models are either unsold or infrequently sold models. The line "Honda,16" is stored in the second output folder.

   - Suppose that the manufacturer Yamaha has 1 unsold model and 10 infrequently sold models. Yamaha **is not selected** because, globally, only 11 of its models are either unsold or infrequently sold models. Yamaha **is not stored** in the second output folder.

   - Suppose that the manufacturer Aprilia has 0 unsold models and 20 infrequently sold models. Aprilia **is selected** because, globally, 20 of its models are either unsold or infrequently sold models. The line "Aprilia,20" is stored in the second output folder.

If you need personalized classes, report for each of them:

- name of the class
- attributes/fields of the class (data type and name)
- personalized methods (if any), e.g, the content of the toString() method if you override it
- do not report get and set methods. I suppose they are "automatically defined"

Predefined Template

...

```
/* Suppose all the needed imports are already set */

…

public class SparkDriver {

    public static void main(String[] args) {
        String inModels;
        String inSales;
        String outputPathPart1;
        String outputPathPart2;

        inModels = "BikeModels.txt";
        inSales = "Sales.txt";

        outputPathPart1 = "outPart1/";
        outputPathPart2 = "outPart2/";

        // Create a configuration object and set the name of the application
        SparkConf conf = new SparkConf().setAppName("Spark Exam - Exercise #2");

        // Create a Spark Context object
        JavaSparkContext sc = new JavaSparkContext(conf);

        /* Write your code here */



}
```