Distributed architectures for big data processing and analytics

January 22, 2021

Student ID ____

First Name ____

Last Name ____

The exam lasts **2 hours**

Part I

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

1. (2 points) Consider the input HDFS folder *myFolder* that contains the following two files:

   - ProfilesItaly.txt
     - the text file ProfilesItaly.txt contains the following three lines (size: 37 bytes)
       Paolo,Turin
       Luca,Rome
       Giovanni,Turin
   - ProfilesFrance.txt
     - the text file ProfilesFrance.txt contains the following two lines (size: 24 bytes)
       Paolo,Nice
       Luis,Paris

   Suppose that you are using a Hadoop cluster that can potentially run up to 2 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 the following key-value pairs (the key part is a name while the value part is always 1):

   (“Paolo”, 1)
   (“Luca”, 1)
   (“Giovanni”, 1)
   (“Paolo”, 1)
   (“Luis”, 1)

   Suppose the number of instances of the reducer class is set to 3 and suppose the reduce method of the reducer class sums the values associated with each key and emits one pair (name, sum values) for each key. Suppose the following pairs are overall emitted by the reduce phase:

("Paolo", 2)
("Luca", 1)
("Giovanni", 1)
("Luis", 1)

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

a) 2

b) 3

c) 4

d) 5

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

- Temperatures2019.txt: size=1048MB
- Temperatures2020.txt: size=1000MB

Suppose that you are using a Hadoop cluster that can potentially run up to 10 instances of the mapper class in parallel. Suppose to execute a MapReduce application for Hadoop that computes some statistics about temperatures. This application is based on one single MapReduce job. The input of this Hadoop application is the HDFS folder *myFolder*. The HDFS block size is 1024MB. The number of instances of the reducer class is set to 5. How many mappers are instantiated by Hadoop (i.e., how many instances of the mapper class) when you execute this MapReduce application by specifying the folder *myFolder* as input?

a) 10

b) 5

c) 3

d) 2

Part II

PoliBike is an international company characterized by several production plants around the world. In each production plant, robots are used to produce some components of the PoliBike's bicycles. PoliBike computes a set of statistics about its production plants and robots. The analyses are performed by considering the following input data sets/files related to the failures of the PoliBike's robots.

- ProductionPlants.txt
  - ProductionPlants.txt is a text file containing the list of production plants of PoliBike. Each line of ProductionPlants.txt is associated with one production plant. The number of production plants is 1000. In some cities, there is more than one production plant.
  - Each line of ProductionPlants.txt has the following format
    - PlantID,City,Country

      where *PlantID* is the production plant identifier while *City* and Country are the city and the country in which the production plant is located, respectively.
    - For example, the following line

      PID1,Turin,Italy

      means that the production plant ***PID1*** is located in ***Turin***, ***Italy***

- Robots.txt
  - Robots.txt is a text file containing the list of robots managed by PoliBike. One line for each robot of PoliBike is stored in Robots.txt. The number of managed robots is more than 15000.
  - Each line of Robots.txt has the following format
    - RID,PlantID,IP

      where *RID* is the robot identifier, *PlantID* is the identifier of the production plant in which the robot is installed, and *IP* is its IP address.
    - For example, the following line

      R15,PID1,130.192.20.21

      means that robot ***R15*** is installed in the production plant ***PID1*** and the IP address associated with that robot is ***130.192.20.21***.

- Failures.txt
  - Failures.txt containing the historical information about the failures of the robots. A new line is inserted in Failures.txt every time there is a failure of a robot. The number of managed robots is more than 15000 and Failures.txt contains the historical data about the Failures of the last 20 years.
  - Each line of Failures.txt has the following format
    - RID,FailureTypeCode,Date,Time

      where *RID* is the identifier of the robot that had a failure, *Date* is the date of the failure, *Time* is the time of the failure, and *FailureTypeCode* is the code of the type of Failure.
    - For example, the following line

      R15,FCode122,2020/05/01,06:40:51

      means that robot ***R15*** had a failure of type ***FCode122*** at ***06:40:51*** (hour=06, minute=40, second=51) of ***May 1, 2020***.

Exercise 1 – MapReduce and Hadoop (7 points)

The managers of PoliBike are interested in performing some analyses about their robots.

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

1. *Number of distinct failure types occurred in year 2018*. The application considers only the failures of year 2018 and computes the number of **distinct** failure types occurred in year 2018. Store the computed value in the output HDFS folder (the output will contain one single line).

   Example.

- For instance, suppose that Failures.txt contains only the following four lines related to year 2018:

  R15,FCode122,2018/01/01,06:40:21
  R16,FCode122,2018/05/02,06:42:53
  R15,FCode123,2018/05/05,04:41:50
  R20,FCode200,2018/12/02,05:40:01

  In this case three distinct failure types occurred in year 2018 and hence one single line containing the value 3 is stored in the output folder.

Suppose that the input is Failures.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 - 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 - 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 PoliBike are interested in performing some analyses related to their production plants and robots.

The managers of PoliBike asked you to develop one single application to address all the analyses they are interested in. The application has five arguments: the three input files ProductionPlants.txt, Robots.txt, Failures.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 RDDs or Spark DataFrames, and write the corresponding Python code, to address the following points:

1. *City with the maximum number of robots.* The application considers only the robots located in Italian cities and selects the Italian city(ies) with the maximum number of robots. The name(s) of the selected city(ies) is stored in the first HDFS output folder. **Pay attention** that many Italian cities (more than one) might be associated with the maximum number of robots. In that case, the output will contain one line for each of the selected cities.

2. *Maximum number of failures per robot for each production plant.* Considering all failures in Failures.txt, the application selects for each production plant the maximum number of failures per robot. The application stores in the second HDFS output folder for production plant its PlantID and the computed maximum number of failures per robot. Pay attention that the output folder must contain also one line for each production plant with a maximum number of failures per robot equal to 0 (i.e., the production plants for which all its robots had no failures).

   Examples

   - Suppose that in the production plant PID1 there are three robots identified by the identifiers R1, R2, and R3, respectively. Suppose that the number of failures of R1 is 10, the number of failures of R2 is 15, and the number of failures of R3 is 3. In this case, the output line associated with the production plant PID1 will be
     - PID1,15
   - Suppose that in the production plant PID2 there are two robots identified by the identifiers R4 and R5, respectively. Suppose that the number of failures of R4 is 0 and the number of failures of R5 is also 0. In this case, the output line associated with the production plant PID2 will be
     - PID2,0

   Suppose sc (Spark Context) and spark (Spark Session) have been already set.