Distributed architectures for big data processing and analytics

September 14, 2020

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 mapper class, overall, how many times is the ***map method*** invoked?

a) 2

b) 3

c) 4

d) 5

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

```
# Read input file
inputRDD = sc.textFile("TemperatureReadings.txt")

# Select the content of the field temperature
tempsRDD = inputRDD.map(lambda line: float(line.split(",")[1]))

# Print on the standard output of the driver the total number of input lines
print("Total number of lines: " + str(inputRDD.count()))

# Print on the standard output of minimum temperature
print("Min. temperature: " + str(tempsRDD.reduce(lambda t1,t2: min(t1,t2))))

# Print on the standard output of maximum temperature
print("Max. temperature: " + str(tempsRDD.reduce(lambda t1,t2: max(t1,t2))))

# Select low temperatures
lowTempsRDD = tempsRDD.filter(lambda temp: temp<0)

# Store the content of lowTempsRDD
lowTempsRDD.saveAsTextFile("outputFolder/")
```

Suppose the input file TemperatureReadings.txt is read from HDFS. Suppose you execute this Spark application only 1 time. Which one of the following statements is true?

a) This application reads the content of TemperatureReadings.txt 1 time

b) This application reads the content of TemperatureReadings.txt 3 times

c) This application reads the content of TemperatureReadings.txt 4 times

d) This application reads the content of TemperatureReadings.txt 7 times

Part II

PoliStreaming is a music streaming company. The registered users of PoliStreaming can listen to the available songs through the PoliStreaming's web site. The managers of PoliStreaming are interested in computing a set of specific statistics based on the following input data sets/files.

- Users.txt
    - Users.txt is a large text file containing the list of registered users of PoliStreaming. PoliStreaming has millions of users, i.e., Users.txt has millions of lines.
    - Each line of Users.txt is associated with the profile of one user and has the following format
        - UID,Name,Surname,Gender,YearOfBirth

          where, *UID* is the unique user identifier, *name* is his/her name, *surname* is his/her surname, *gender* is his/her gender, and *YearOfBirth* is his/her year of birth.

        - For example, the line

          UID15,Paolo,Garza,Male,1976
    - means that the user identified by the UID ***"UID15"*** is ***"Paolo"*** (name) ***"Garza"*** (surname), he is a ***"Male"*** and was born in ***1976***.

- Songs.txt
  - Songs.txt is a large text file containing the catalog of available songs. It contains more than 10000 songs.
  - Each line of Songs.txt is associated with one song and has the following format
    - SID,Title,MusicGenre

      where, *SID* is the unique song identifier, *Title* is its title, and *MusicGenre* is its music genre (Rock, Metal, etc.). In this catalog, each song is associated with one single music genre.

    - For example, the line

      SID10, New Year's Day, Rock

      means that the title of song *"**SID10**"* is "***New Year's Day***" and its music genre is "***Rock***".

- ListenToSongs.txt
  - ListenToSongs.txt is a large text file.
  - Every time a user listens to a song, a new line is appended to the end of ListenToSongs.txt. ListenToSongs.txt contains the data collected in the last 15 years, i.e., it contains billions of lines.
  - Each line of ListenToSongs.txt has the following format
    - SID,UID,StartTimestamp,EndTimestamp

      where *SID* is a song identifier and *UID* is a user identifier. The meaning of each line is "*UID* listened to *SID* from S*tartTimestamp* to *EndTimestamp"*.

    - For example, the line

      SID10,UID15,2019/06/01_14:18:23,2019/06/01_14:20:05

      means that user ***UID15*** listened to song ***SID10*** from ***14:18:23 of June 1, 2019*** to ***14:20:05 of June 1, 2019***.

Exercise 1 – MapReduce and Hadoop (7 points)

The managers of PoliStreaming are interested in performing some analyses about their users.

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

1. *Year of birth of the youngest female user and number of female users associated with that year of birth*. The application selects the year of birth of the youngest female user and computes how many female users are associated with that year of birth. Store in the output HDFS folder the selected year of birth and the number of female users associated with that year of birth. The format of the output line is YearOfBirth youngest female user*t*Number of female users associated with that year of birth (the separator is the tab symbol).

   Example.

   For instance, suppose that the youngest female user was born in year 2000 and there are 102 female users who were born in year 2000. The following line is stored in the output

   2000*t*102.

Suppose that the input is Users.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 PoliStreaming are interested in performing some analyses related to songs.

The managers of PoliStreaming 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 Users.txt, Songs.txt, ListenToSongs.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. *Songs never listened to by young users in the last two years.* The application selects the songs than were never listened to by young users (a user is a young user if he/she was born after 1999) in the last two years (i.e., from September 14, 2018 to September 13, 2020). The identifiers (SIDs) of the selected songs are stored in the first HDFS output folder (one SID per line).

   Note. The value of StartTimestamp is used to decide when a user listened to a specific song. Do not consider the value of EndTimestamp.

2. *For each song, the year in which it is characterized by its highest yearly popularity.* Considering all lines of ListenToSongs.txt and only the songs that have been listened to at least one time, the application selects for each song the year in which that song is characterized by its highest annual popularity. The annual popularity of a song in a specific year is given by the number of **distinct** users who listened to that song in that specific year. The application stores in the second HDFS output folder for each song its SID and the selected year (one pair (SID,year) per line). Given a song, if there is more than one year associated with that song's highest annual popularity value, the first year according to the temporal order is selected and stored in the output folder.

   Note. The value of StartTimestamp is used to decide in which year a user listened to a specific song. Do not consider the value of EndTimestamp.

   Examples.

   - Suppose that song SID10 was listened to by 500 distinct users in year 2010, 510 distinct users in year 2011, and 490 distinct users in year 2012. The pair SID10, 2011 is stored in the output folder.

   - Suppose that song SID50 was listened to by 400 distinct users in year 2010, 210 distinct users in year 2011, and 400 distinct users in year 2012. The pair SID50, 2010 is stored in the output folder.

   For the sake of simplicity, in these toy examples only three years are considered but pay attention that ListenToSongs.txt contains the data collected in the last 15 years.

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