Big Data: Architectures and Data Analytics

July 1st, 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 cache "mechanism" of Spark. Which one of the following statements is true?

   a) Caching an RDD is always useful

   b) An RDD that is used only one time in an application must always be cached

   c) An RDD must always be cached by using the MEMORY_ONLY storage level if its size is larger than the maximum amount of main memory of the cluster

   d) Caching an RDD that is used multiple times in an application can improve the efficiency of the application (in terms of execution time).

2. (2 points) Consider the HDFS file log.txt. The size of log.txt is 1024MB. Suppose that you are using an Hadoop cluster that can potentially run up to 2048 mappers in parallel and suppose to execute the word count application, based on MapReduce, by specifying log.txt as input file. Suppose that Hadoop automatically sets the number of mappers to 2 (i.e., it runs two mappers) when you execute the word count application by specifying log.txt as input file. What is the block size of the HDFS file system?

   a) Block size: between 1024MB and 2048MB

   b) Block size: between 512MB and 1023MB

   c) Block size: between 256MB and 511MB

   d) Block size: between 128MB and 255MB

Part II

PoliBooks is a company selling books. The management of PoliBooks is interested in analyzing their customers and books.

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

- BoughtBooks.txt
  - BoughtBooks.txt is a textual file containing the list of books that have been bought (purchased) by the customers of PoliBooks
  - Every time a customer buys a book a new line is appended at the end of BoughtBooks.txt, i.e., each line of the file contains the information about one purchase
  - Each line of the file has the following format
    - customerid,bookid,timestamp,price

      Where *customerid* is a customer identifier, *bookid* is the identifier of a book, timestamp is the time at which *customerid* bought/purchased *bookid* and *price* is the cost of the purchase.

    - For example, the line

      customer1,book122,20160506_23:10,14.99

      means that ***customer1*** bought ***book122*** on ***May 6, 2016*** at ***23:10*** and the price of the purchase was ***14.99*** *euro*

- Books.txt
  - Books.txt is a textual file containing the list of available books with the associated characteristics
  - The file contains one single line for each book
  - Each line of the file has the following format
    - bookid,title,author,suggested_price

      Where, *bookid* is the identifier of a book, *title* is its title, *author* is the author of the book, and suggested_price is the price suggested by the editor of the book.

    - For example, the line

      book122,The Body in the Library,Agatha Christie,25.19

      means that the title of ***book122*** is "***The Body in the Library***", the author is "***Agatha Christie***", and the suggested price is ***25.19*** *euro*.

Exercise 1 – MapReduce and Hadoop (10 points)

The managers of BoughtBooks are interested in identifying the best sellers among the books in the catalogue of the company. A book is considered a ***best seller*** if the gross revenue associated with the book is at least one million euro (the gross revenue associated with a book is given by the sum of the prices of all sales (purchases) of the book).

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

A. *Select the list of best seller books*. Specifically, the application must select the list of books that are best sellers based on the definition reported above. Store the bookids of the selected books in an HDFS folder. The name of the output folder is one argument of the application. The other argument is the path of the input file BoughtBooks.txt.

Exercise 2 – Spark and RDDs (17 points)

The management of PoliBooks is interested in identifying the expensive books that have never been sold in order to remove them from the catalogue of the company. Specifically, a book is classified as an "***expensive never-sold***" book if the suggested price of the book is greater than 30 euro and the book has never been sold.

The management of PoliBooks is also interested in identifying the customers with a propensity for "cheap purchases". A purchase is classified as a "***cheap purchase***" if the price of the purchase is less than 10 euro. For instance, the following line of BoughtBooks.txt is a "cheap purchase":

customer1,book142,20160507_21:15,9.49

A customer is classified as a "***customer with a propensity for cheap purchases***" if at least *min_threshold*% of his/her purchases are "cheap purchases". *min_threshold* is a threshold provided by the management of PoliBook and it is one argument of the application.

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 Books.txt and BoughtBooks.txt and the value of min_threshold, which 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 expensive never-sold books*. Specifically, the application must select the bookids of the “expensive never-sold” books, based on the definition reported above, and store the bookids of the selected books in an HDFS folder. The application must also print the total number of “expensive never-sold” books on the standard output. The name of the output folder is one argument of the application.

B. *Select the customers with a propensity for cheap purchases*. Specifically, the application must select the “customers with a propensity for cheap purchases”, based on the definition reported above, and store the customerids of the selected customers in an HDFS folder. The name of the output folder is one argument of the application.