

Project 1

Total Points: 200

Release Date: 01/30/2017

Due Date: 02/12/2017 (11:59PM)

Teams: Project to be done in teams of two.

Short Description

In this project, you will write map-reduce jobs in Java language, Apache Pig scripts, and run them on Hadoop system.

Detailed Description

You are asked to perform three activities in this project, (1) Create datasets, (2) upload the datasets into Hadoop HDFS, (3) Query the data by writing map-reduce Java code, and (4) Query the data using Pig scripts.

1-Createing Datasets [20 Points]

Write a java program that creates two datasets (two files), *Customers* and *Transactions*. Each line in *Customers* file represents one customer, and each line in *Transactions* file represents one transaction. The attributed within each line are comma separated.

The *Customers* dataset should have the following attributes for each customer:

- ID: unique sequential number (integer) from 1 to 50,000 (that is the file will have 50,000 line)
- Name: random sequence of characters of length between 10 and 20 (***do not include commas***)
- Age: random number (integer) between 10 to 70
- CountryCode: random number (integer) between 1 and 10
- Salary: random number (float) between 100 and 10000

The *Transactions* dataset should have the following attributes for each transaction:

- TransID: unique sequential number (integer) from 1 to 5,000,000 (the file has 5M transactions)
- CustID: References one of the customer IDs, i.e., from 1 to 50,000 (on Avg. a customer has 100 trans.)
- TransTotal: random number (float) between 10 and 1000
- TransNumItems: random number (integer) between 1 and 10
- TransDesc: random text of characters of length between 20 and 50 (***do not include commas***)

Note: The column names will NOT be stored in the file. Only the values comma separated. Form the order of the columns; you will know each column represents what.

2-Uploading Data into Hadoop [5 Points]

Use hadoop file system commands (e.g., put) to upload the files you created to Hadoop cluster.

To learn about the file system commands check this link:

<https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-common/FileSystemShell.html>

Note: It is good to check your files and see how the files are divided into blocks and each block is replicated. You can do that by checking the web Interface of Hadoop (Check the Readme file in your virtual machine to know to do that).

3-Writing MapReduce Jobs

You will write Java programs to query the data in Hadoop. Before writing your code you should perfectly understand the “WordCount” example (it like the “Hello World...” example in Java). You can find its code online, and it is also included in your virtual machine (Check the Readme file).

Notes:

- You should decide whether each query is a map-only job or a map-reduce job, and write your code based on that. A given query may require more than a single map-reduce job to be done.
- You can always check the query output file from the HDFS website and see its content.
- You can test your code on a small file first to make sure it is working correctly before running it on the large datasets.

Hint: It is important to know how Hadoop reads and writes integers, floats, and text fields. Check `IntWritable`, `FloatWritable`, and `Text` classes to know which one to use and when.

3.1) Query 1 [20 Points]

Write a job(s) that reports the customers whose CountryCode between 2 and 6 (inclusive).

3.2) Query 2 [20 Points]

Write a job(s) that reports for every customer, the number of transactions that customer did and the total sum of these transactions. The output file should have one line for each customer containing:

CustomerID, NumTransactions, TotalSum

You are required to use a Combiner in this query.

3.3) Query 3 [30 Points]

Write a job(s) that joins the Customers and Transactions datasets (based on the customer ID) and reports for each customer the following info:

CustomerID, Name, Salary, NumOf Transactions, TotalSum, MinItems

Where *NumOfTransactions* is the total number of transactions done by the customer, *TotalSum* is the sum of field “TransTotal” for that customer, and *MinItems* is the minimum number of items in transactions done by the customer.

3.4) Query 4 [30 Points]

Write a job(s) that reports for every country code, the number of customers having this code as well as the min and max of *TransTotal* fields for the transactions done by those customers. The output file should have one line for each country code containing:

CountryCode, NumberOfCustomers, MinTransTotal, MaxTransTotal

Hint: To get the full mark of Query 4, you need to do it in a single map-reduce job.

3.5) Query 5 [30 Points]

Write a job(s) that reports the customer names whose associated number of transactions is larger than the average number of transactions of all customers. That is, calculate the average number of transactions over all customers (Say value x), your goal is to report the customer names who have more than x transactions.

4-Writing Apache-Pig Jobs

4.1) Query 1 [15 Points]

Write an Apache Pig query that reports the customer names that have the least number of transactions. Your output should be the customer names, and the number of transactions.

4.2) Query 2 [15 Points]

Write an Apache Pig query that join Customers and Transactions using Broadcast (replicated) join. The query reports for each customer the following info:

CustomerID, Name, Salary, NumOf Transactions, TotalSum, MinItems

Where *NumOfTransactions* is the total number of transactions done by the customer, *TotalSum* is the sum of field "TransTotal" for that customer, and *MinItems* is the minimum number of items in transactions done by the customer.

4.3) Query 3 [15 Points]

Write an Apache Pig query that reports the Country Codes having number of customers greater than 5,000 or less than 2,000.

What to Submit

You will submit a single zip file containing the Java programs for ***Creating Data Files, Java code for the MapReduce Queries, the Apache Pig scripts***, plus a document (.pdf) containing any comments you would like to provide regarding your code.

How to Submit

Use blackboard system to submit your files.