Project 1 for CS585/DS503: Big Data Management - Spring 2019

"Friend Analytics"

Total Points:	**100**
Given Out:	**Monday, 28th Jan, 2019**
Due Date:	**Friday, 8th Feb, 2019 (11:59PM)** **Submit the project via CANVAS.**
Teams:	**Project is to be done in teams -- Project1-team.** **Team members will be assigned.**

Project Overview

In this project, you will create datasets and upload them into Hadoop HDFS. Next, you will analyze the data in a scalable fashion by writing custom analytics tasks using map-reduce Java code as well as Apache pig scripts and run those on Hadoop system. Your team should compare the two alternate approaches at analyzing the data in terms of their respective features, considering ease of developing the analytics code, size of the code itself, as well as the resulting performance.

Project Submission (required to receive a grade for this project)

1. You will submit a single zip file containing the programs for creating data files, Java and/or python code and Pig scripts for your MapReduce queries via CANVAS.
2. You will also submit a document (pdf) containing documentation that describes how you accomplished each task. It is not just important that it "run", it also should be a scalable solution. In your report, you must indicate **the relative contributions of each team member explicitly.** [1] This requires you to openly discuss with each other your expectations and if you are satisfied with each other's efforts. By submitting, all team members confirm the division of labor as indicated in your report.
3. Lastly, each of you **must independently** submit an online team survey to the CS585 staff about your contributions in relation to those of your team member to this project effort here: https://goo.gl/forms/QzxmmJ0VOCYfVm163
These comments will be treated confidentially.

[1] For instance, if each team member has done the project independently, and then only at the end you pulled the best of the material together, you need to say so. If you have closely collaborated and done the same amount of effort working side by side, report this.

Project Demonstration

Once completed, one or two teams will be asked to provide a brief demonstration of their results in class to your classmates to review how you solved this project. If you have a good solution, we encourage you to include a PPT of a few pages to highlight your solution. You will get 4 bonus points, if selected to be among the top solutions for project 1.

If necessary for grading, the TAs will communicate with your team and also may request a demonstration of your solution, as needed.

Project Description

1-Creating Datasets [10 Points]

Write a java or python program that creates data sets related to a Facebook-like application, including the following datasets as three separate data files: ***MyPage, Friends,*** and ***AccessLogs.***

Each line in the MyPage file represents one person, and should include at least the following attributes describing the person as listed below.

Each line in the Friends dataset file describes which person has indicated that they are friends with another person (this is a one-directional relationship) and the timing when this friend relationship was declared.

Each line in the AccessLog data file indicates which person p1 has accessed the Facebook page that belongs to a second person p2, including the timing of the access.

The datasets below should have the following attributes, *but you are free to make the actual field values more realistic if you would like as well as to design additional attributes of interest to your application.* The attributes within each line are comma separated.

The ***MyPage*** dataset should have the following attributes for each Facebook page:

ID: unique sequential number (integer) from 1 to 100,000 indicating the owner of the page (there will be 100,000 lines)

Name: random sequence of characters of length between 10 and 20 (do not use commas instead this string)

Nationality: random sequence of characters of length between 10 and 20 (do not use commas instead this string)

CountryCode: random number (integer) between 1 and 70

Hobby: random sequence of characters of length between 10 and 20

The ***Friends*** dataset should have the following attributes for each friend relationship:

FriendRel: unique sequential number (integer) taken from value in the range from 1 to 20,000,000 (the file has 20,000,000 lines and thus friend relationships)

PersonID: Person-ID of a person who has a Facebook page, i.e., from 1 to 100,000 people

MyFriend: References ID of a person that you are friend with, i.e., from 1 to 100,000. This relation is not mutually necessarily, i.e., it just indicates that you declare that you are friends with this friend ID.

DateofFriendship: random number (integer; or some other sequential data type to use as date) between 1 and 1,000,000 to indicate when the friendship started

Desc: text of characters of length between 20 and 50 explaining the type of friendship: collegefriend, girlfriend, family, etc.

The ***AccessLog*** dataset should have the following attributes for each Facebook access:

AccessId: unique sequential number (integer) from 1 to 10,000,000

ByWho: References the Id of the person who has accessed the Facebook page

WhatPage: References the Id of the page that was accessed

TypeOfAccess: random text of characters of length between 20 and 50 explaining if just viewed, left a note, added a friendship, etc.

AccessTime: random number between 1 and 1,000,000 (or other data type of your choosing)

A column name should not include a comma. The column names will not be stored in the file. Only the values are listed. Each value should be separated by a comma. From the order of the columns; you will know what each column represents.

2. Loading Datasets into Hadoop [10 Points]

Use hadoop file system commands (e.g., put) to upload your data files into Hadoop cluster.

To learn about the file system commands, review the link here:
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-Accomplishing Analytics Tasks using MapReduce Jobs [40 Points]

You will write Java programs to realize the following tasks on your data to analyze your data. Before writing your code and/or queries, you should review the "WordCount" example. It is 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). You may want to use that as your guiding example for the java code solution development:
http://hadoop.apache.org/docs/r2.9.2/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html

General Guidelines:

- Learn how Hadoop reads and writes integers, floats, and text fields. Check IntWritable, FloatWritable, and Text classes to know which one to use.
- You should determine whether a query is a map-only job, a map-reduce job, or several map-reduce jobs. If a task can be done with a simpler solution, then you should describe this simpler solution to get full credit.
- Develop a solution with and without a map-reduce combiner, when possible. If not possible for your query, please state so explicitly. Work with as many features of Hadoop as possible (e.g., such as to control how many mappers are used) to get to know Hadoop. Explain explicitly any ideas you have tried out.
- Report the performance for the execution each of your tasks below. In particular, compare the relative performance of different solutions.
- Document each of your tasks in the report line by line what it accomplishes.

TIP: You want to test things on a small test file first and check the query output file from the HDFS website to make sure your answer is correct, before running it on the large datasets. Hint: It is important to study how Hadoop reads and writes integers, floats, and text fields; thus check IntWritable, FloatWritable, and Text classes.

2.a) Task a

Write a job(s) that reports all Facebook users (name, and hobby) whose Nationality is the same as your own Nationality (pick one). Note that nationalities in the data file are a random sequence of characters unless you work with meaningful strings like Chinese or German. This is up to you.).

2.b) Task b

Write job(s) that reports for each country, how many of its citizens have a Facebook page.

2.c) Task c

Find the top 10 interesting Facebook pages, namely, those that got the most accesses based on your AccessLog dataset compared to all other pages.

2.d) Task d

For each Facebook page, compute the "happiness factor" of its owner. That is, for each Facebook page in your dataset, report the owner's name, and the number of people listing him or her as friend.

2.e) Task e

Determine which people have favorites. That is, for each Facebook page owner, determine how many total accesses to Facebook pages they have made (as reported in the AccessLog) and how many distinct Facebook pages they have accessed in total.

2.g) Task f

Find the list of all people that have set up a Facebook page, but have lost interest, i.e., after some initial time unit (say 10 days or whatever you choose) have never accessed Facebook again (meaning no entries in the Facebook AccessLog exist after that date).

2.f) Task g

Identify people that have declared someone as their friend yet who have never accessed their respective friend's Facebook page – indicating that they don't care enough to find out any news about their friend (at least not via Facebook).

2.h) Task h

Report all owners of a Facebook who are famous and happy, namely, those who have more friends than the average number of friends across all owners in the data files.

4 - Accomplishing Analytics Tasks using Apache Pig [40 Points]

Now write the above eight analytics tasks using Apache Pig scripts.

---------------------------- *the end* -----------------------------