ECE 454/ECE 750 Distributed Systems
Assignment 3

Due Date: July 28^{th} at 11:59pm Waterloo time

Instructor: Dr. Wojciech Golab wgolab@uwaterloo.ca

Assignment TA: Rania Ibrahim rania.ibrahim@uwaterloo.ca

1 Learning Objectives

- To gain hands-on experience with Apache Hadoop:
  - Defining MapReduce jobs to solve real world problems.
  - Implementing MapReduce jobs.
- To gain hands-on experience with Apache Pig.

2 House Rules

- You can work in groups of 1, 2 or 3 students.
- For programming language, you are allowed to use Java 1.7 or scala 2.11.6 (or compatible).
- Electronic submission (one per group) using Dropbox in LEARN.

3 Problem Description

Given a cancer sample - gene matrix G $\in \mathbf{R}^{m \times n}$ where m is the number of cancer samples and n is the number of genes. Cell g_{ij} in the matrix represents the expression value of gene j in cancer sample i (higher expression value indicates that this gene is frequently found in the cancer sample cells).

4 Input Format

Due to the large size of the matrix G, it is horizontally partitioned into multiple files, where each file contains a subset of the cancer samples.

Each file consists of multiple lines, where each line corresponds to a unique cancer sample and each line starts with the sample id, then a comma-separated

list of gene expression values for the sample. For example, an input file may contain the following lines:

sample_1,0.5,0.7,0.8
sample_2,0.3,0.2,0.1

This means that file 1 contains two samples with 3 genes. Sample 1 contains gene 1 with expression value 0.5 and sample 2 contains gene 3 with expression value 0.1.

5 Requirements

The following is a detailed description of the assignment requirements:

- **Part 1**: For each cancer sample, report the gene with the highest expression value using MapReduce jobs. In case of multiple genes having the same highest expression value, you should report them all. The output for this part should be a folder that contains one or more files. Each file contains the output for a subset of the samples where each line of the file represents a sample. The line should start with the sample id and then the gene or genes with the highest expression values comma separated. The following is an example of one of the output files:

  sample_1,gene_1,gene_2
  sample_2,gene_3

  Where sample_1 is the id of sample 1 and gene_1 and gene_2 are the genes with the same highest expression value for this sample and sample_2 is the id of sample 2 and gene_3 is the gene with the highest expression value for sample 2.

- **Part 2**: For each gene x, report the score S_x representing how much this gene is related to the cancer using MapReduce jobs. A gene is related to a cancer sample if its expression value for this sample is above its normal expression value (Assume that normal expression value for all genes is 0.5). The score S_x for gene x is calculated as follows:

  $$S_x = \frac{\text{Number of cancer samples gene x is related to}}{\text{Total number of cancer samples}}$$

  The output for this part should be a folder that contains one or more files. Each file contains the output for a subset of genes, where each line of the file represents a gene. The line should start with the gene id and then the gene score comma separated. The following is an example of one of the output files:

  gene_1,0.7
  gene_2,0.1

- **Part 3**: Compute the similarity between each pair of the cancer samples using MapReduce jobs. Similarity between two samples s_i and s_j can be calculated as the dot product between their gene expression values, as follows:

$$\text{Similarity}(s_i, s_j) = \sum_{k=1}^{n} g_{ik} * g_{jk}$$

where g_{ik} is the expression value of gene k at sample i and g_{jk} is the expression value of gene k at sample j. You should only report non-zero similarities in the output. Note that genes expression values for many samples can be zero which either means that these samples don't contain these genes or these genes expression values are not measured for these samples, which is common in Bioinformatics analysis. The output for this part should be a folder that contains one or more files. Each file contains the output for a subset of pairs of samples where each line of the file represents the similarity between a pair of samples. The line should be in the following format (sample_id1,sample_id2,similarity). The following is an example of one of the output files:

sample_1,sample_2,0.3
sample_3,sample_2,0.7

- **Part 4**: Repeat the previous three parts using Apache Pig. The input to the Pig program will be the same as previous parts. The output for part (1) should be one file for all samples following the same file format specified in part (1). In addition, the output for part(2) should be one file for all genes following the same file format specified for part (2) and the output for part(3) should be one file for all pairs of samples following the same file format specified for part(3).

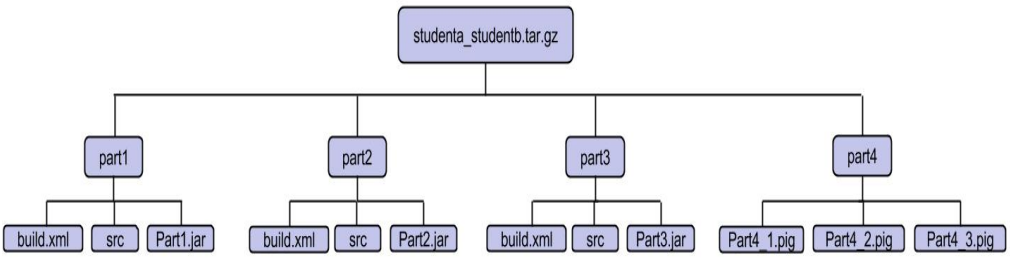
6 Submission Instructions

You should submit two files: a tarball file and a design document (max 1 page). The design document should be in one of the following formats: pdf, doc, docx, rtf, ppt, or pptx. The tarball should be named with all group members Nexus IDs separated by underscores, for example "studenta_studentb.tar.gz" for students with Nexus ids studenta and studentb. The tarball should contain 4 folders named "part1", "part2", "part3" and "part4", where each folder contains the solution to one of the previously descried parts (see section 5 Requirements). Specifically each folder of the first three parts will contain the following:

- "src" folder containing the source code for solving the corresponding part.
- jar file named either "Part1.jar", "Part2.jar" or "Part3.jar" according to the folder name. Note that the Java class name should be the same as the jar name.

- build.xml file to build your source code using the ant tool.

"part4" folder will contain three Pig scripts files named "Part4_1.pig", "Part4_2.pig" and "Part4_3.pig" containing the scripts to solve the sub parts of part 4. The following figure shows the exact hierarchical structure of the tarball file.



The grader will test your code on Hortonwoks sandbox appliance version 2.2.4, Hadoop 2.6.0 and Apache Pig version 0.14.0 using the following commands:

```
hadoop jar Part1.jar Part1 inputFolder outputFolder
hadoop jar Part2.jar Part2 inputFolder outputFolder
hadoop jar Part3.jar Part3 inputFolder outputFolder
pig -param input=inputFolder -param output=outputFile Part4_1.pig
pig -param input=inputFolder -param output=outputFile Part4_2.pig
pig -param input=inputFolder -param output=outputFile Part4_3.pig
```