**Programming Assignment 6  
Matrix Multiplication using Pig**

Due on Tuesday November 20 before midnight

**Description**

The purpose of this project is to develop a simple program for matrix multiplication using Apache Pig.

This project must be done individually. No copying is permitted. **Note: We will use a system for detecting software plagiarism, called**[**Moss**](http://theory.stanford.edu/~aiken/moss/)**, which is an automatic system for determining the similarity of programs.** That is, your program will be compared with the programs of the other students in class as well as with the programs submitted in previous years. This program will find similarities even if you rename variables, move code, change code structure, etc.

Note that, if you use a Search Engine to find similar programs on the web, we will find these programs too. So don't do it because you will get caught and you will get an F in the course (this is cheating). Don't look for code to use for your project on the web or from other students (current or past). Just do your project alone using the help given in this project description and from your instructor and GTA only.

**Platform**

As in the previous projects, you will develop your program on [SDSC Comet](https://lambda.uta.edu/cse6331/comet.html). You may use Pig on Comet in local mode interactively, but you need to setup your PATH first:

module load hadoop

export PATH=$PATH:/oasis/projects/nsf/uot143/fegaras/pig-0.16.0/bin

Then, to evaluate Pig Latin commands interactively, do:

pig -x local

**Setting up your Project**

Login into Comet and download and untar project6:

wget http://lambda.uta.edu/cse6331/project6.tgz

tar xfz project6.tgz

chmod -R g-wrx,o-wrx project6

Go to project6/examples and look at the join.pig example. You can run it in standalone mode using:

sbatch join.local.run

or using

pig -x local join.pig

Optionally, you can run it in distributed mode using:

sbatch join.distr.run

**Project Description**

You are asked to implement matrix multiplication using Apache Pig. A sparse matrix M is represented as a dataset of triples (i,j,v) so that M[i,j]=v. Missing entries are assumed to have a zero value. The Pig schema for such triple is ( I: long, J: long, V: double ). For example, the matrix:

1,0,3.0

0,0,5.0

1,2,-2.0

2,0,9.0

0,1,-3.0

0,2,-1.0

1,1,8.0

2,1,4.0

represents the 3x3 matrix:

5.0 -3.0 -1.0

3.0 8.0 -2.0

9.0 4.0 0.0

The multiplication of two matrices M and N is a matrix P such that Pij=sumkMik\*Nkj. It can be done using the following SQL query:

select M.I, N.J, SUM( M.V \* N.V )

from M, N

where M.J = N.I

group by M.I, N.J

Your project is to implement matrix multiplication using Pig.

An empty multiply.pig is provided as well as scripts to run this code on Comet. There are two small sparce matrices 4\*3 and 3\*3 in the files M-matrix-small.txt and N-matrix-small.txt for testing in local mode. Their matrix multiplication must return the 4\*3 matrix in result-matrix-small.txt. Then there are 2 moderate-sized matrices 200\*100 and 100\*300 in the files M-matrix-large.txt and M-matrix-large.txt for testing in distributed mode. Note: you can access the input matrices in Pig (which are passed as parameters) as $M and $N and the output as $O. That is, you use LOAD $M USING ..., etc.

To run it in local mode over the two small matrices use:

sbatch multiply.local.run

The result matrix in the directory output must be similar to result-matrix-small.txt. After you make sure that your program runs correctly in local mode, you run it in distributed mode using:

sbatch multiply.distr.run

This will multiply the moderate-sized matrices and will write the result in the directory output-distr.

**Documentation**

You can learn more about Pig at:

* [Pig: Getting Started](http://pig.apache.org/docs/r0.16.0/start.html)
* [Pig Latin Basics](http://pig.apache.org/docs/r0.16.0/basic.html)

**What to Submit**

You need to submit the following files only:

project6/multiply.pig

project6/multiply.local.out

project6/output/part-r-00000

project6/multiply.distr.out

Top of Form

Bottom of Form