CSE 603 **Programming Assignment 4**
Due Date: Dec 3, 2014
Weight: 13% of your grade

Problem: (a) Simple Parallel Bucket Sorting using MapReduce (8% credit) and
(b) Categorize Twitter Messages using Mahout Naive Bayes Classifier and K-means (5% credit)

Description: In this assignment you have two distinct tasks described separately below:

(a) You have to implement the simple bucket-sorting algorithm using MapReduce (in Hadoop) on the CCR platform. The description of bucket sorting algorithm were given in the previous assignments.

- Your task is to parallelize this algorithm using Hadoop cluster in CCR
- For the sorting algorithm, use the programs that were provided for assignment 2 to generate random numbers with a normal distribution.
- You will need to do a comparison of the performance of this Mapreduce implementation with all previous implementations, *i.e.,* OpenMP, MPI, and GPU.

(b) In this part you will use Mahout Naive Bayes Classifier and Mahout K-means clustering method to categorize Twitter messages into 7 different categories. This part is intended to get you familiar with using Mahout on Hadoop cluster.

- You will be given Twitter messages in Hadoop sequence file format, stored in directory tweets-seq. (To get a better understanding of how this sequence file is obtained, you can check the provided conversion program and run it on the original twitter messages, which is given as tweets.tsv)
- You will have to use Mahout command-line invocation on CCR Hadoop cluster.
- Check the provided SLURM script for Mahout Naive Bayes Classifier; use it as an example to create your own K-means script to cluster the data.
- In both methods, use tf-idf vector as input.

What you need to do:

- Your report should include the following:
  - Your rationale for parallelizing the program
  - Speedup plots for varying problem size and cores
  - Complete source codes in different directories –including makefiles for each embedded within the directories. Include scripts you may have used to run programs with different parameters and also scripts that you may have used to collate results and create graphs (highly recommended to do this to make it easier to finish your work in time).
  - Script (with complete description) to run your programs (including samples)
  - Discussion of results to include the following (graphs)
    - Impact on the performance of the algorithm based on
      - Increase in the size of the problem
      - Increase in the number of cores
      - Impact of load balance on performance

- How does this relate to your theoretical evaluation of the scalability of this sorting algorithm?
- Comparison between results from using Mahout Naive Bayes Classifier and K-means algorithm, in terms of scalability and quality

Extra Credit (1%): Incorporate any other classifier and compare its performance with the other two prescribed classifiers.

Specific Submission Guidelines:

1. Please submit your assignment as username.zip
2. Root directory should include Sort and Twitter directories. The Twitter directory should include Bayes and Kmeans subdirectories and additionally another one indicated by name of method if you are doing the extra credit problem. Folder name should be as indicated above.
3. Each sub directory must contain a Makefile, source file (if any), SLURM script and dataset you may have used.
4. Use provided random_number_generator_normal for sorting problem.
5. For all the subdirectories in Twitter directory, put any input data you have used in /data folder, put SLURM scripts in /script folder. If you are doing extra credit problem, put source file in /src folder, put all the libraries you have used in /lib folder . Extract the entire program in a runnable jar file and include Readme.txt to indicate the classpath, mahout version, hadoop version and any other libraries you may have used. Put all the instructions on running your program in Readme.txt.
6. Name of the executable file (after running make command), should be as same as source file (e.g. for sort file, the executable -o file should be named as sort)
7. For each program, when <-t problem size> arguments are passed, it should print out the sorted results and execution time. For example, when running ./sort -t 100, it should print out the sorted result for problem size 100 on standard output along with execution time.
8. Submit your report as username.pdf.
9. In your report, include execution time for the following cases. (For sorting problem, set maximum number as 10 * problem size)

Program	Number of nodes	Problem size
Sequential	1 node	100,000
Sequential	1 node	1,000,000
Mapreduce	2 nodes	100,000
Mapreduce	2 nodes	1,000,000

Mapreduce	4 nodes	100,000
Mapreduce	8 nodes	100,000
Mapreduce	4 nodes	1,000,000
Mapreduce	8 nodes	1,000,000
Mapreduce	16 nodes	1,000,000

Apart from the above test cases, you should also include other experiment results in your report.

10. For Twitter part, report should include the maximum accuracy you have reached, the corresponding parameters you have used and the corresponding dataset you have used.

Note: username in this context is your UB username, and the other file names should be followed exactly as stated.