ELEN4020A: Data Intensive Computing
Laboratory Exercise No 3: Using MapReduce Framework

To Be Completed By: 11:55Hrs (11:55AM) April 15th, 2019

Outcome

This is a group assignment. The objective of the assignment is introduce students to the principles of the mapreduce framework for processing big data. Main features in such a framework being the reliable and fault-tolerant processing techniques employed. While not all problems are solvable by the MapReduce techniques, a large number of decomposable problems can be split to be solved by the MapReduce techniques.

The main outcome will be:

i.) Learning how to decompose a big data processing problem into sub-tasks executed by workers but coordinated by a master task. This phase is referred to the *mapping* phase.

ii.) Learning how the results from worker performing the sub-tasks are eventually merged into a final result. This phase is referred to as the *reduction* phase.

iii.) Learning to use the fundamental principles of how MapReduce works using an alternate approach to Hadoop, e.g., Phoenix++, Mrs-MapReduce, MrJob or DISCO. Hadoop is an opensource implementation from Apache of the MapReduce framework written in Java. The original MapReduce concept came from Google. Pheonix-2/Pheonix++ is a C and C++ respective implementations. The rest are Python equivalent implementation.

Problem Description

Work Schedule

The work involves:

1. Designing and implementing MapReduce algorithms for a variety of common data processing tasks. These need not be a cluster of machines but on a single machine with multi-cores (up to say 8 cores). The required algorithms are:

   i) A simple word count algorithm of a text. This gives the frequencies of currencies of words in a text. You need not include *Stop Words, e.g, for, as the, is, at, which, on.* You can include your list of *Stop Words* that you ignored in your submission. Consider words to be *case-insensitive.*

   ii) Top-K query. The K most frequently occurring words, ignoring stop words, for $K = 10, 20$.

   iii) An inverted index of the text. This involves listing, for each word, the line numbers of the text that the words occur. Only list out about 50 lines of the distinct words.

2. You are free to make an implementation language of your choice; either in C, C++ or Python. Some Python-Based or C/C++-Based MapReduce framework are given below. There are other C++-based and Python3-Based MapReduce frameworks available. My recommendation is to choose one from the following.

C/C++: Pheonix++ [https://github.com/kozyraki/phoenix];

[https://github.com/kozyraki/phoenix/tree/master/phoenix-2.0];

[https://csinparallel.org/csinparallel/modules/PhoenixMRIntro.html].

Python: Mrs-MapReduce: [https://pythonhosted.org/mrs-mapreduce/index.html];
MrJob: [https://pythonhosted.org/mrjob/guides/quickstart.html];

[https://github.com/Yelp/mrjob];

Spark: [http://spark.apache.org/];

DISCO: [http://discoproject.org/];

3. Conduct some program tests with a small and then a medium/large texts. Choose a small text of your own.

4. Conduct your tests with *File2ForLab3.txt*, for the large text.

5. The first task of the word-count algorithm is the most common algorithm used in explaining MapReduce. I hope your reading of the listed Websites and possible download of the codes will assist you to get going. The subsequent tasks will require you to think a bit more on how to solve the tasks.

The Deliverable

- Submit your codes, for marking in your group's repository on GitHub.
- Provide high level description of your algorithms to each of the required tasks in pseudo-codes.
- Give values of your performance results and the results for running your programs with *File1ForLab3.txt* and *File2ForLab3.txt*.
- Write a short set of instructions on how to access your GitHub repository and send this to Sakai. This should be submitted by only the lead member of your group.

Resources

You can download your choice of Mapreduce framework onto your computer/laptop and work from there. Alternatively, you can work by logging onto "hornet01.eie.wits.ac.za" and work in your home directory. Please note that our network has frequently been giving some problems but it has been stable in the last two days. Please read instructions on how to logon to hornet01. Our jaguar cluster has still some problems, but should be ready soon. If you run on hornet01, the GCC is gcc-6.5. It has cudu-9.1 enabled but no cuDNN. If you intend to use python3 on hornet01, please create your own anaconda3 installation in your home directory and work via setting up a python environment.