CS434: Introduction to Parallel and Distributed Computing Laboratory Exercise No 3: Using a MapReduce Framework

To Be Completed By: 11:55Hrs (11:55AM) March 16th, 2020

Outcome

This is a group assignment. The objective of the assignment is to 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 even on a distributed computing network.

The main outcome will be:

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

ii.) Learning how the results from workers 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 alternative to Hadoop, such as Phoenix++, Mrs-MapReduce, MrJob or DISCO. Hadoop is an opensource implementation from Apache for MapReduce 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 on 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 occurrencies of words in a text. You need not include *Stop Words; e.g, for, as. the, is, at, which, on. etc.*. You can include your list of *Stop Words* that you ignored in your submission. Consider words to be *case-insensitive.*, i.e., ”Rebel” is the same ”rebel.”

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

2. You are free to select 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://mrjob.readthedocs.io/en/latest/];

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

Spark: [http://spark.apache.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 them .

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 *File2ForLab3.txt*.
- Write a short set of instructions on how to access your GitHub repository and send this to Canvas. This should be submitted by only one of the members of your group.

Resources

You can download your choice of Mapreduce framework onto your computer/laptop and work from there. If you intend to use python3 please create your own anaconda3 installation in your home directory and work via setting up a python environment.