CS 434: Parallel and Distributed Computing Lab 4

Kweku Yamoah: 71712022

April 19, 2021

Introduction

This report was generated as a requirement of Lab 4 for Parallel and Distributing computing. The report contains two sections; Part 1 and Part 2. Part 1 is the Lab where I experiment with the MaoReduce framework and my understanding in class. Part 2 is the project which focuses on implementing matrix multiplication using MapReduce algorithms.

Part 1

MapReduce Framework & Why

My selected MapReduce framework was MrJob. I chose MrJob because of the following reasons:

- it provides an easy route for writing python programs that run on Apache Hadoop.
- it provides an easy switch for input and output formats with a single line of code.
- All codes are written in a single python class which enforces the understanding of OPP concepts.
- it has an extensive documentation with examples as reference, which, makes learning of the framework easy.

Installing MrJob was easy because it require running a command with the python pip package. However, understanding the documentation and how to submit my first job took some time to understand.

Word-count Algorithm

Firstly, the word-count algorithm prints out the frequency of each word in a text file.

The word count algorithm inherited the **MrJob** class to implement a job that executes the task. This job class required the implementation of the methods **mapper_init, configure_args, mapper and reducer**.

- The *configure_args* method creates command-line options. I used tis to pass the stop words file as a command line argument.
- The *mapper_init* method initializes or setups the resources that the mapper method needs. I read the stop words and converted the words to a python set in this method.
- *mapper* method splits every sentence from the text file into words. All the stop words are removed before further processing. The method finally returns a key-value pair consisting of the word as the key and a one as the value for every word. E.g., word, 1.
- *reducer* method in this program acts as a combiner and a reducer at the same time. This method finds the occurrences of each word in the text by summing up the results from the mapper method passed to it. The output of the reducer method is a key-value pair containing a word as the key and the value as the number of times it appears in the text.

Running the Program

The program needs some requirements to be satisfied first before running.

- The program requires python 3 or later to run.
- The program requires the correct command-line arguments to be passed before it can run.
- The program must be run on a Linux subsystem; Ubuntu is preferred.

To run the program type this in your terminal whilst ensuring that you are in the directory which contains the program files; **python3 mrjob_wordcount.py <input.txt –stop-words=stop_words.txt**

Part 2