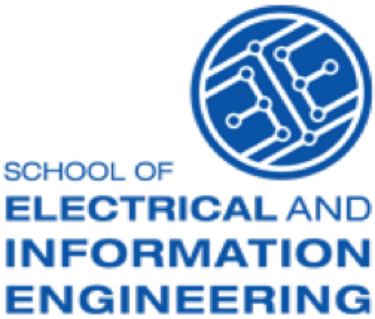

University of the Witwatersrand
School of Electrical and Information Engineering

ELEN4020: Data Intensive Computing

Laboratory Exercise 2

Authors:

Kayla-Jade Butkow
714227

Jared Ping
704447

Lara Timm
704157

Matthew van Rooyen
706692

Date Handed In: 9th March, 2018

1. Matrix Multiplication

The input to map function consists of two separate text files containing a matrix formatted as a list of strings. The first row of the file provides the dimensions of the given matrix. The proceeding rows contain three columns, the first two representing the row and column index of the element and the third the value of the element.

Each matrix is read in of the form M_{ij} and N_{jk} where i represents the row elements and j the column elements. The product matrix is therefore represented as $P_{i,j} = M_{ij} * N_{jk}$. This is achieved by taking the first row of M and multiplying it by the first column in N and summing the values. This function is performed for every column in B before repeating the process for every row in A.

2. Map Reduce

In order to perform matrix multiplication for large matrices, a MapReduce algorithm was implemented. The algorithm consists of two key features, namely the `map()` function and `reduce()` function.

The input matrix to the map function is of the form (row, column, value). This data is then assigned a key (i, j, k) within the may function in order to produce a key-value format at the output. The function is performed on each matrix separately before combining the values of each matrix according to the key value producing key-value pairs.

The algorithm consists of two reduce functions. The reduce function performs calculations on the key-value output pairs produced by the map function. `reducer_multiply` simply multiplies the values found at each key (i, k) and prepares the data for summation. The reduce function only processes the data one key at a time. `reducer_sum` sums the output values from the multiplication at each key performing the final step in the matrix multiplication process.

3. MrJob

MrJob is a python-based MapReduce framework. The steps function represent the jobs handled within the script. This also allows for multiple one-step jobs to be executed within the program. The mapper function performs allows for the map function to be presented as a single job. The input data is read in one line at a time after which the data is broken down into single values and given a key. Each yeild function within the mapper is outputted as a single line of code. The reducer performs the multiplication and sums the output results before emitting them. MrJob allows for the inclusion of two separate files as input streams to be included in the program.

MrJob is preferable in it's application as the code has no dependencies with Hadoop in comparison to the other frameworks. the framework was also chosen due to documentation being more readily available and easier to implement.

4. Results

4.1 Algorithm 1

4.2 Algorithm 2