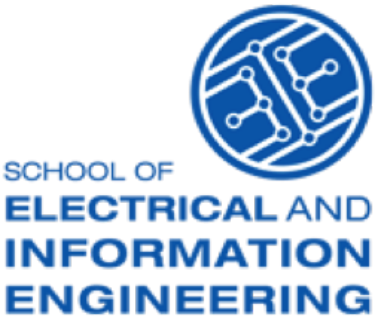

University of the Witwatersrand
School of Electrical and Information Engineering

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 Transposition

In order to allow for simple matrix transposition, the two-dimensional matrix is created as a one-dimensional array, populated in row-major form. The matrix is sequentially filled with integers corresponding to the index+1.

Transposition of the matrix begins with allocating memory for an array to store the indices of swapped values. To for each existing index a new index is calculated using Equation

$$newIndex = (oldIndex * N)\%(N^2 - 1) \tag{1}$$

2. OpenMP

3. PThreads

4. Comparison of Performance

Table 1: Performance of the algorithm using 4 threads

	$N_0 = N_1 = 128$	$N_0 = N_1 = 1024$	$N_0 = N_1 = 8192$
PThread			
OpenMP			

Table 2: Performance of the algorithm using 8 threads

	$N_0 = N_1 = 128$	$N_0 = N_1 = 1024$	$N_0 = N_1 = 8192$
PThread			
OpenMP			

Table 3: Performance of the algorithm using 16 threads

	$N_0 = N_1 = 128$	$N_0 = N_1 = 1024$	$N_0 = N_1 = 8192$
PThread			
OpenMP			

Table 4: Performance of the algorithm using 64 threads

	$N_0 = N_1 = 128$	$N_0 = N_1 = 1024$	$N_0 = N_1 = 8192$
PThread			
OpenMP			

Table 5: Performance of the algorithm using 128 threads

	$N_0 = N_1 = 128$	$N_0 = N_1 = 1024$	$N_0 = N_1 = 8192$
PThread			
OpenMP			

Appendix