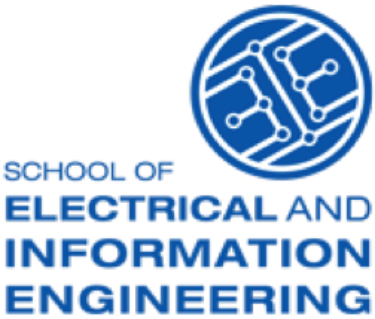

University of the Witwatersrand
School of Electrical and Information Engineering

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 matrix, populated in row-major form.

2. OpenMP

The multi-threaded algorithm was implemented using OpenMP. Multi-threading was performed on all aspects of the process, including the populating of the matrix, the populating of the index placeholder array, and the transposition of the array.

For all of the processes using OpenMP, the loops were divided into chunks of the size of the matrix divided by 256. Each function takes in a parameter called `noOfThreads`, which allows the user to specify the number of threads that should be used to execute the process.

2.1 Populating the matrix

Since the matrix is large, populating it in serial takes a large amount of time. It was thus implemented in parallel. Since each loop takes a consistent amount of time, static scheduling was used [1].

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			

REFERENCES

[1] G. Hager and G. Wellein. *Introduction to High Performance Computing for Scientists and Engineers*. CRC Press, 2011.

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