

OPEN MP PROGRAMMING: ASSIGNMENT 3

MEASUREMENT REPORT

Siddhartha Srinadhuni

9508286854

sisr16@student.bth.se

Madhukar Enugurthi

9506077537

maen16@student.bth.se

Measurement Report:

Execution time of Quick Sort.

Time	Sequential Version of Quick Sort
User	25.60 seconds
System	0.13 seconds
Elapsed	0:25.80 seconds
CPU Usage	99%

Table 1: Sequential execution time of quick sort.

Time	Parallel Version of Quick Sort	
No. of CPUs	1 CPU	8 CPU
User	25.62 seconds	25.86 seconds
System	0.13 seconds	0.13 seconds
Elapsed	0:25.76 seconds	0:07.6 seconds
CPU Usage	99%	367%

Table 2: Parallel execution time of quick sort on 1 and 8CPUs.

The parallel version of quick sort evidently performs better than the sequential version. **The speedups established when 8CPUs are used is 3.39.** To overcome the overhead when threads are created, a small segment of code is implemented sequentially.

Execution time of Gaussian Elimination.

Time	Sequential Version of Gaussian Elimination
User	24.45 seconds
System	0.03 seconds
Elapsed	0:24.53 seconds
CPU Usage	99%

Table 3: Serial execution time of Gaussian Elimination

Time	Parallel Version of Gaussian Elimination	
No. of CPUs	1 CPU	8 CPU
User	26.65 seconds	59.44 seconds
System	0.04 seconds	8.45 seconds
Elapsed	0:26.71 seconds	0:09.48 seconds
CPU Usage	99%	716%

Table 4: Parallel execution time of Gaussian Elimination.

Our OpenMP implementation of the Gaussian elimination program has a **speedup of 2.6 (on 8 cpus) over the sequential version.**

Compiling and execution procedures for quick sort:

- The local machine has to be connected with kraken. Kraken.tek.bth.se is used as a remote computer.
- For compiling the parallel version of Quicksort using OpenMP, use the command “gcc -fopenmp -o parqs quicksortfinal.c” where quicksortfinal.c is the file name.
- The number of cores can be changes in the commandline using export OMP_NUM_THREADS= 1 or 8 after compiling
- Upon compiling, use “/usr/bin/time ./parqs” to measure the execution time.

Compiling and execution procedures for Gaussian elimination:

- For compiling the parallel version of Gaussian Elimination using OpenMP, use the command “gcc -fopenmp -o pargauss gomp.c” where gomp.c is the file name.
- The number of cores can be switched between 1 and 8 within the declaration in NUMB_CORES. Meaning, if the program has to be executed on 8 cores, NUMB_CORES has to be 8 in the code.
- Upon compiling, use “/usr/bin/time ./pargauss” to measure the execution time.