Homework 2, 3 and 4 HPC

Mayank Vadsola (300075960)

25/04/2019

Contents

List of Figures

1 RAS: Restricted Additive Schwarz

The computation for RAS was performed on INTEL Xeon Gold 6130 (2.1 GHz) processor that has 32 cores. It took approximately 137.0644 secs on a single core for $96 \times 96 \times 96$ domain for which boundary and initial conditions are set to zero.

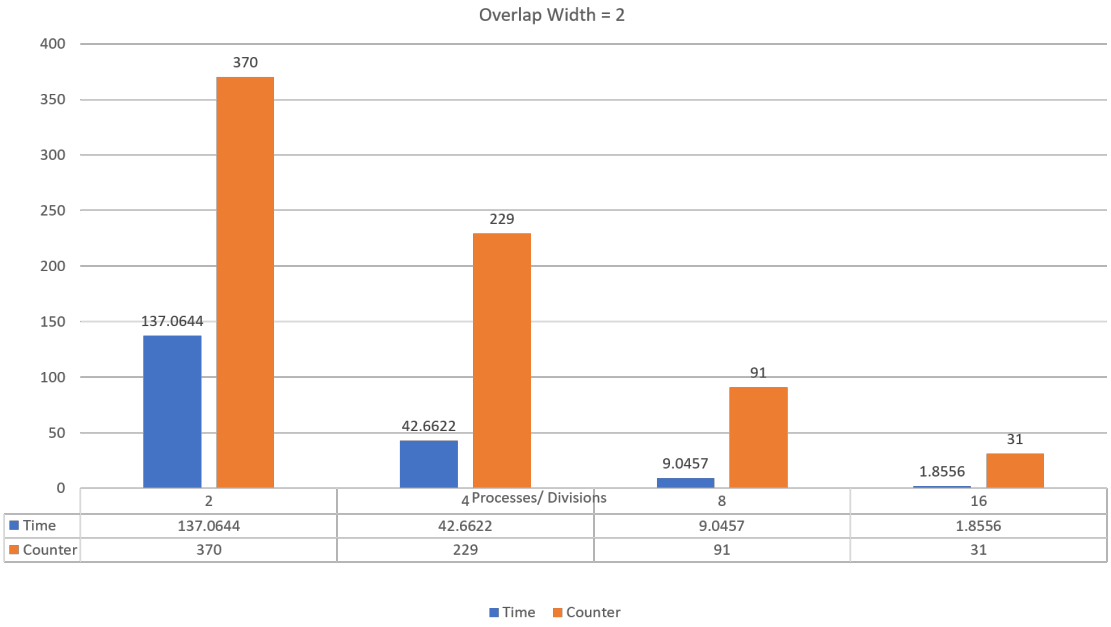

Figure 1: Time and Iterations comparison for $96 \times 96 \times 96$ Domain, Overlap Width $= 2$

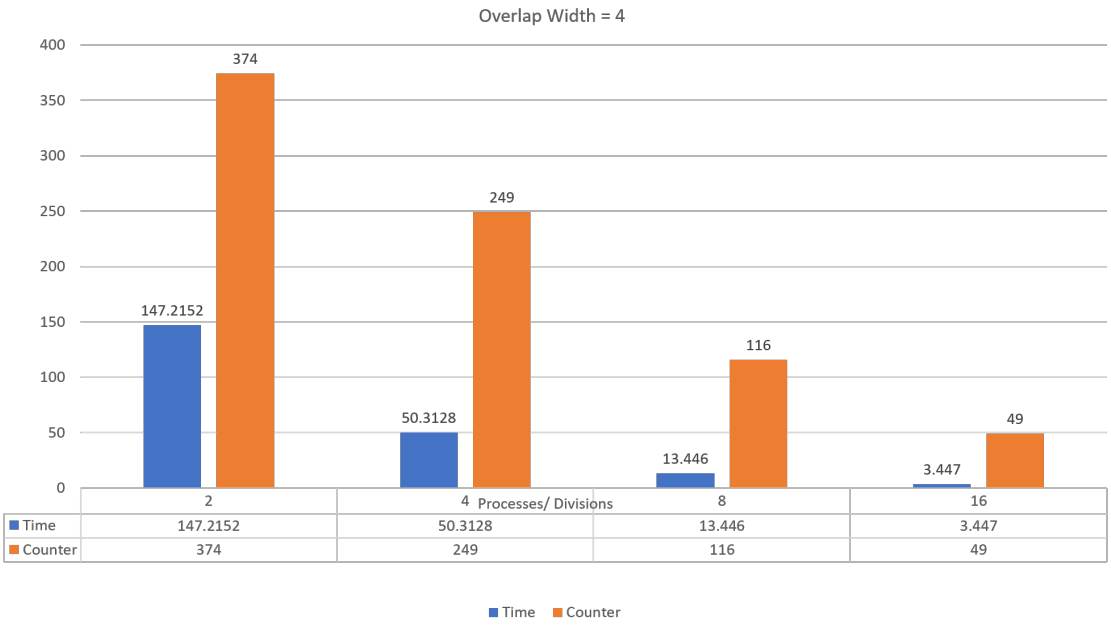

Figure 2: Time and Iterations comparison for $96 \times 96 \times 96$ Domain, Overlap Width $= 4$

The code is developed such that it is compatible with any even size of overlap width. Also, the number of overlaps(divisions) is equal to the number of processes. So, it is not possible to compare efficiency and speedup for the different number of processes as the number of overlaps changes with the number

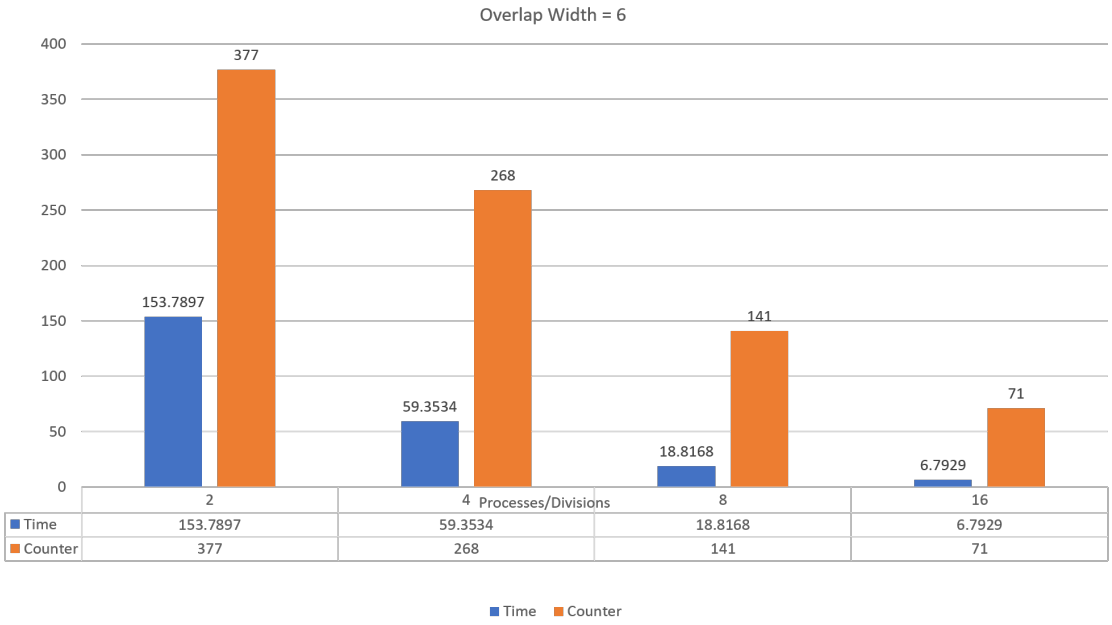

Figure 3: Time and Iterations comparison for $96 \times 96 \times 96$ Domain, Overlap Width $= 6$

of processes.

It can be concluded from the figures 1, 2, and 3 that the time and number of iterations for convergence decreased with increase in number of processes/divisions. Figure 4 shows the Z sectional view of first processor solution that is in agreement with the exact solution.

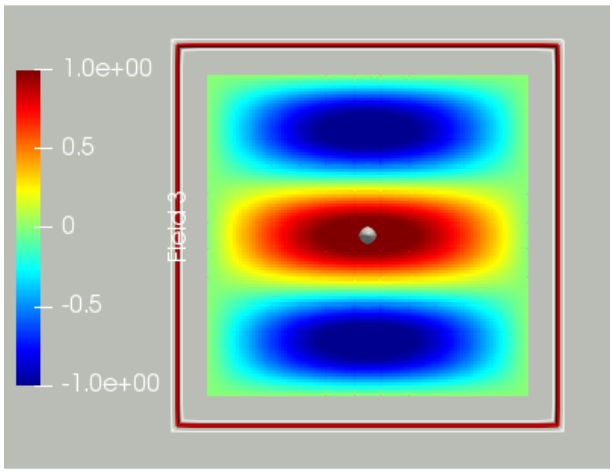

Figure 4: RAS solution, Z-Sectional View of First Processor

2 CGM: Conjugate Gradient Method

The computation for CGM was performed on INTEL Xeon Gold 6130 (2.1 GHz) processor that has 32 cores. It took approximately 0.04068 secs and only one iteration on a single core for $100 \times 100 \times 100$ domain for which boundary and initial conditions are set to zero.

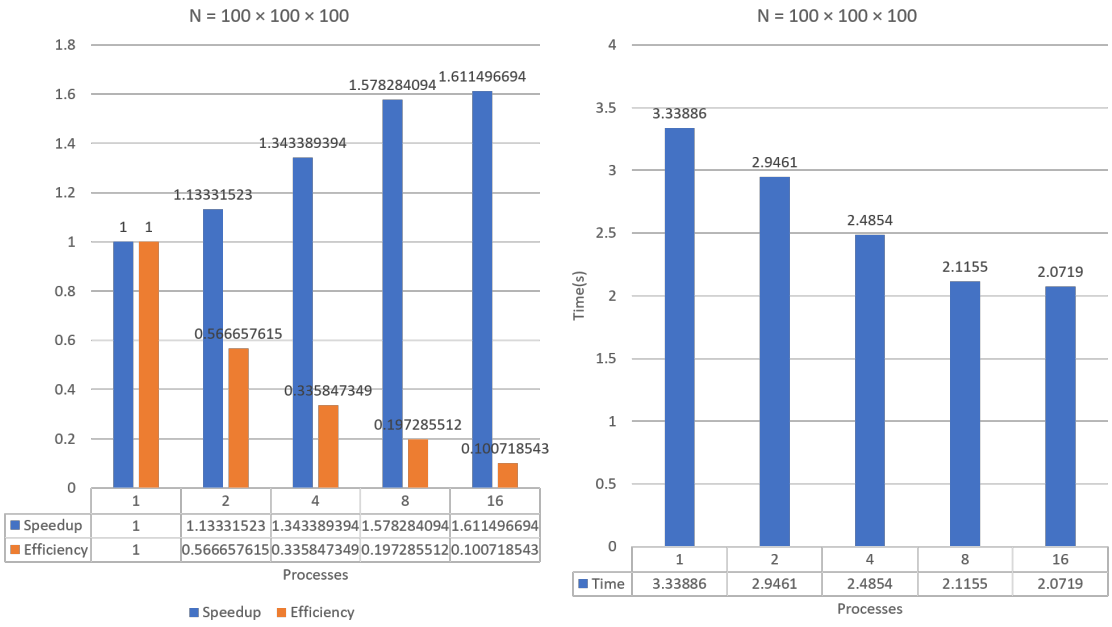

Figure 5: Speedup and Efficiency comparison for $100 \times 100 \times 100$ Domain, Static Scheduler (Chunk Size = Default)

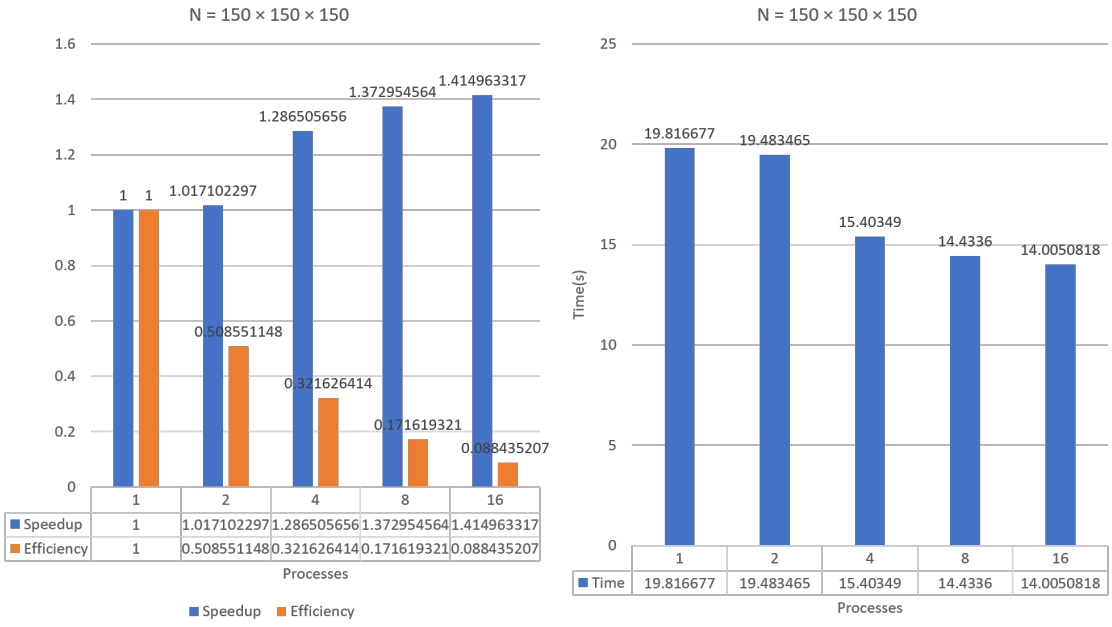

Figure 6: Speedup and Efficiency comparison for $150 \times 150 \times 150$ Domain, Static Scheduler (Chunk Size = Default

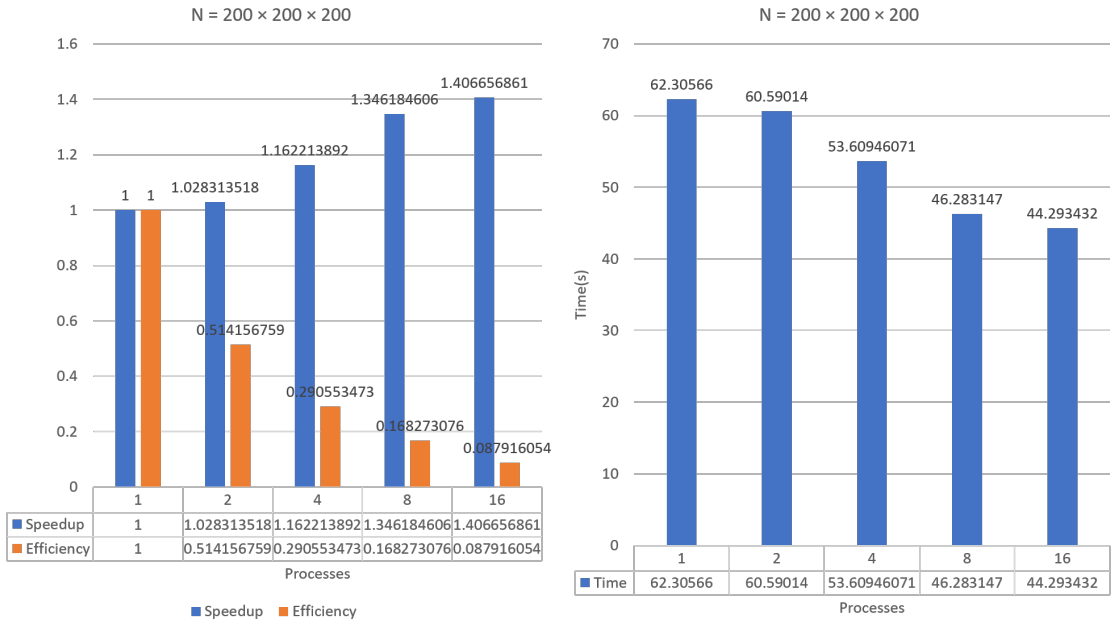

Figure 7: Speedup and Efficiency comparison for $200 \times 200 \times 200$ Domain, Static Scheduler (Chunk Size = Default

It is observed that by setting the initial condition to zero which is the mean of the exact solution, CGM converges in one iteration. So, to measure parallel efficiency initial condition was set to 10 and the boundary condition was kept as 0. Number of iterations for $100 \times 100 \times 100$, $150 \times 150 \times 150$ and $200 \times 200 \times 200$ domains were 325, 488, and 652 respectively.

It can be seen from the figures 5, 6, and 7 that the speedup increased with number of threads but not efficiently. Moreover, it is observed that the speedup and efficiency are better for a default chunk size. Figures 8 and 9 represents Z and X sectional views of solution obtained by CGM that is in harmony with exact solution.

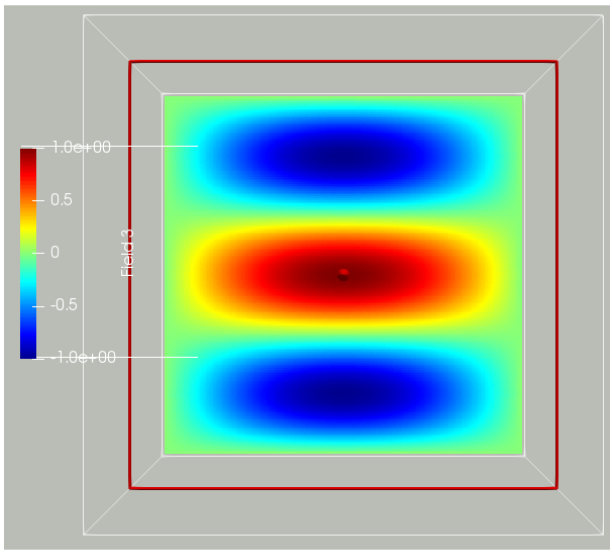

Figure 8: CGM solution, Z-Sectional View

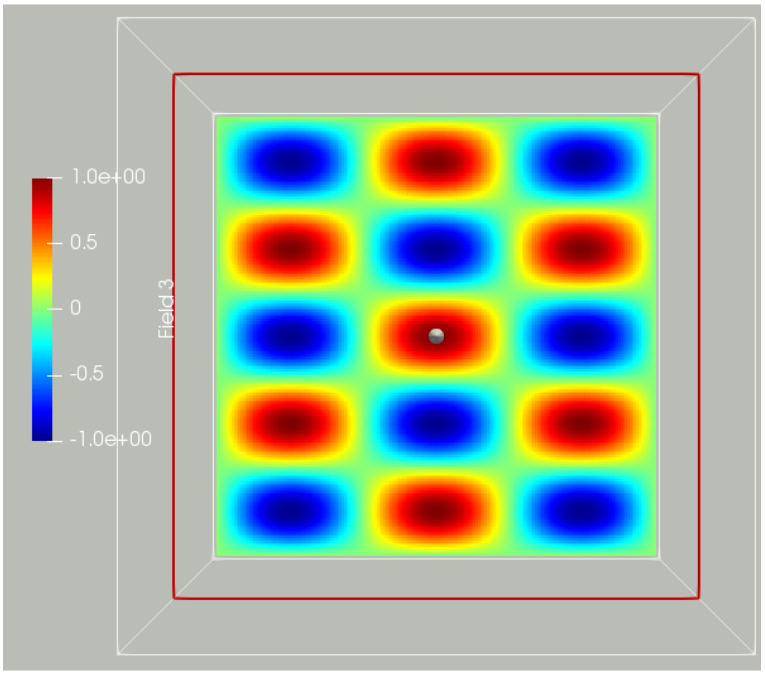

Figure 9: CGM solution, X-Sectional View

3 FFT: Fast Fourier Transform

The computation for FFT was performed on INTEL 7700HQ (2.8 GHz) quad-core processor that can be hyper-threaded to 8. It took approximately 1.01865 secs on a single core for 2^{20} domain.

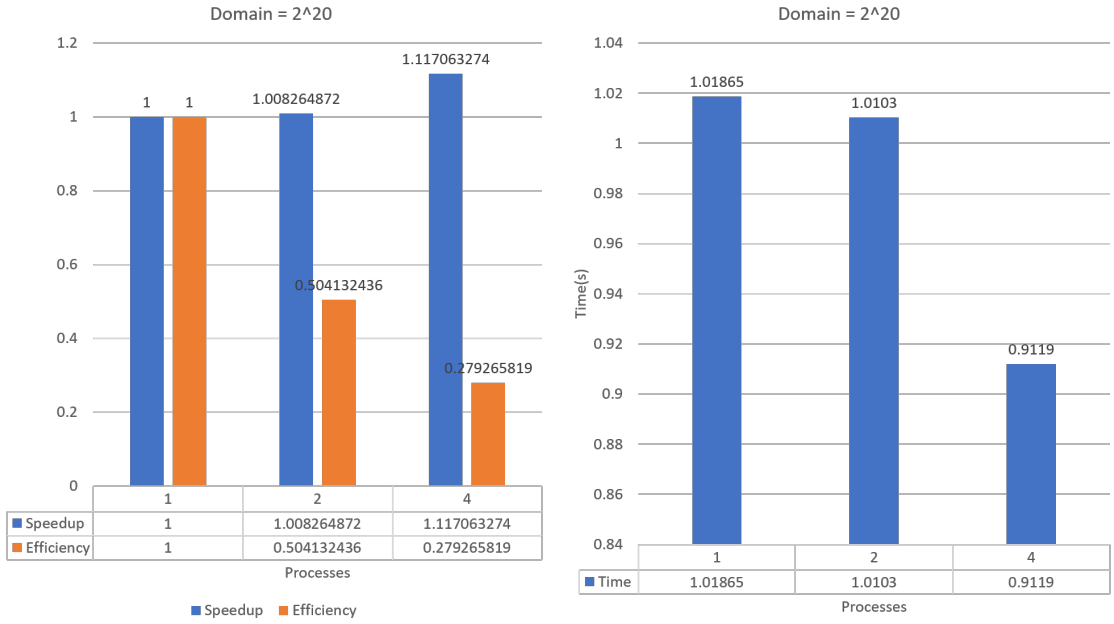

Figure 10: Speedup,Efficiency, and Time comparison for 2^{20} domain

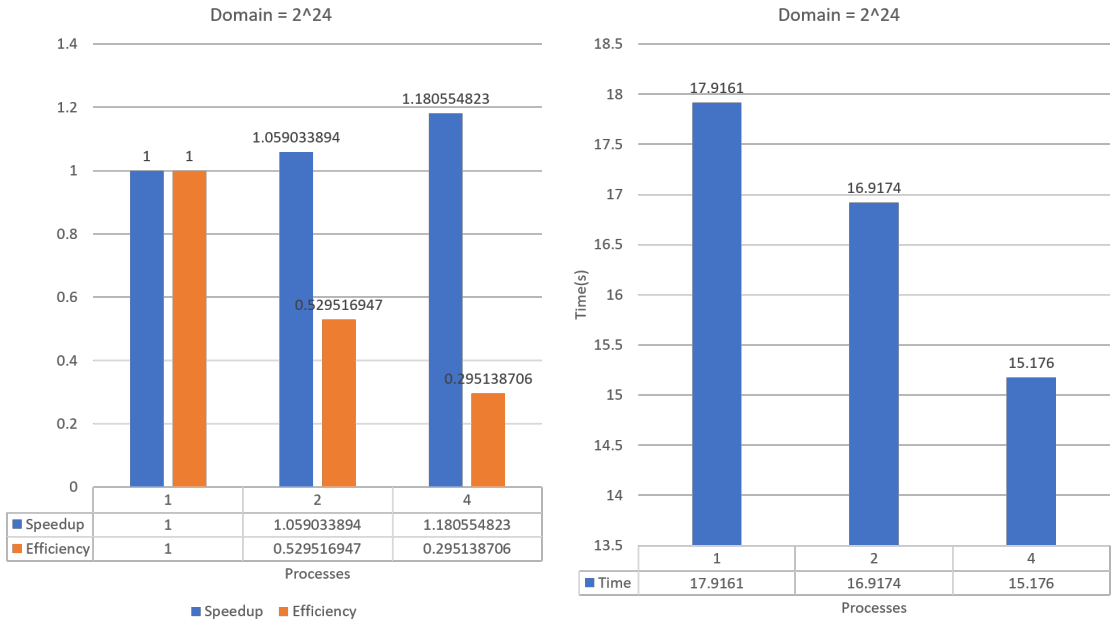

Figure 11: Speedup,Efficiency, and Time comparison for 2^{24} domain

To measure parallel efficiency two different domains of size 2^{20} and 2^{24} were considered. The value of X_{459339} was found to be $1.27725 + 2.28332i$ for the original domain.

It can be observed from the figures 10 and 11 that the speedup was observed with increase in a number of threads although not efficient.