Homework 1 HPC

Mayank Vadsola (300075960)

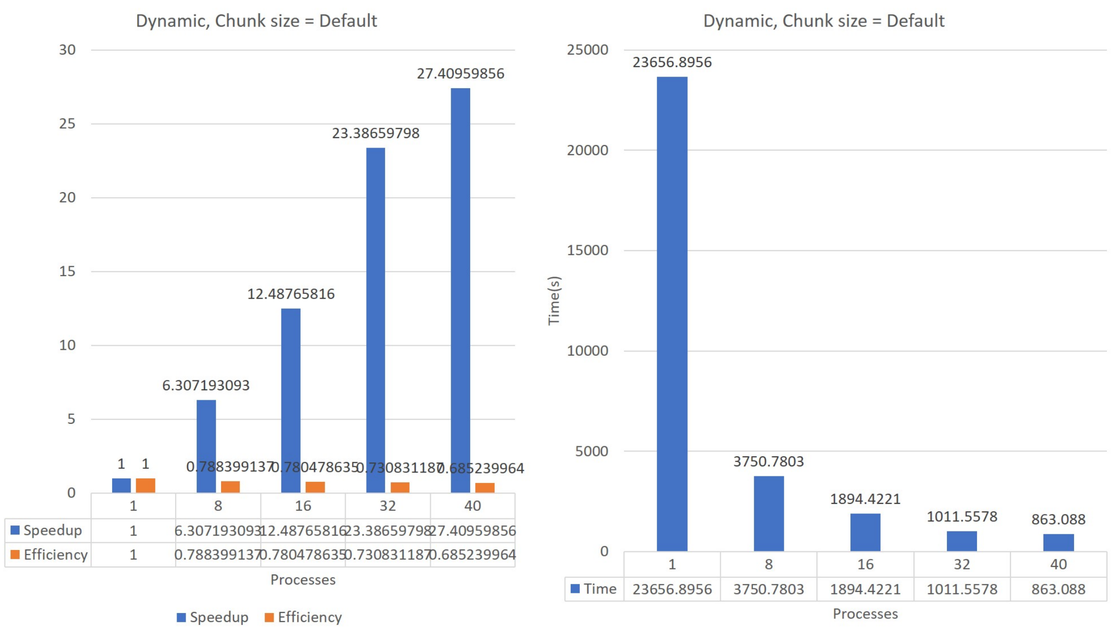
14/03/2019

Contents

List of Figures

1 Mandelbrot Set

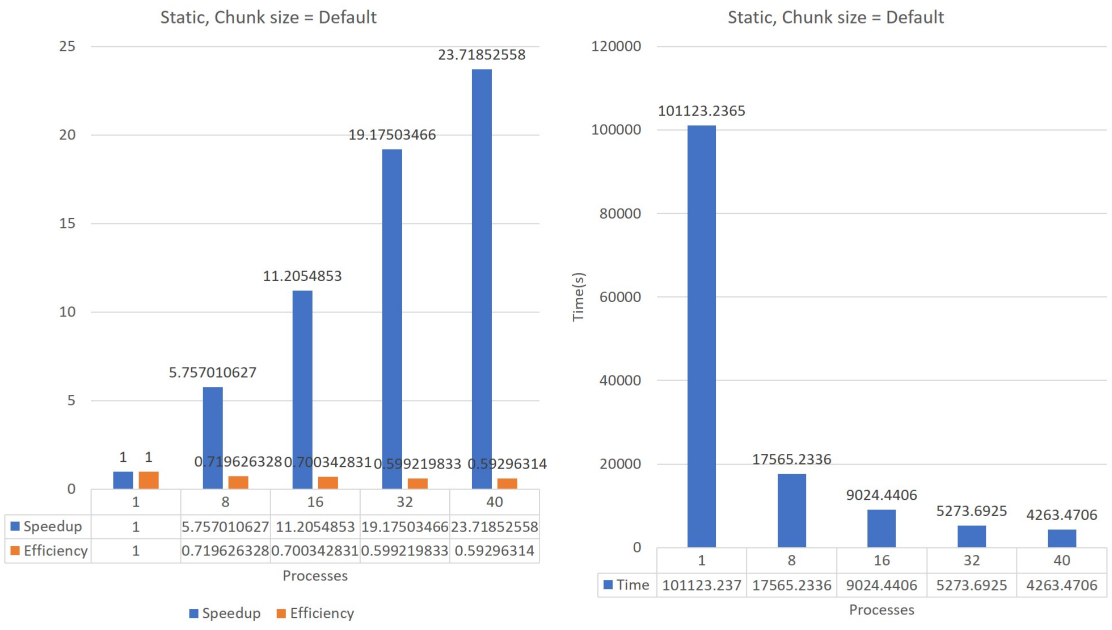
The computation for Mandelbrot Set was performed on SCINET Compute Canada Cluster. Each node has 40 cores which can be hyper-threaded to 80. Speedup and efficiency comparison is represented in figure 1, 2, and 3. Figure 4 describes the full Mandelbrot Set solution. It took approximately 863 secs on 40 cores using Dynamic Scheduler.



	1	8	16	32	40
■ Speedup	1	6.307193093	12.48765816	23.38659798	27.40959856
■ Efficiency	1	0.788399137	0.780478635	0.730831187	0.685239964

	1	8	16	32	40
■ Time	23656.8956	3750.7803	1894.4221	1011.5578	863.088

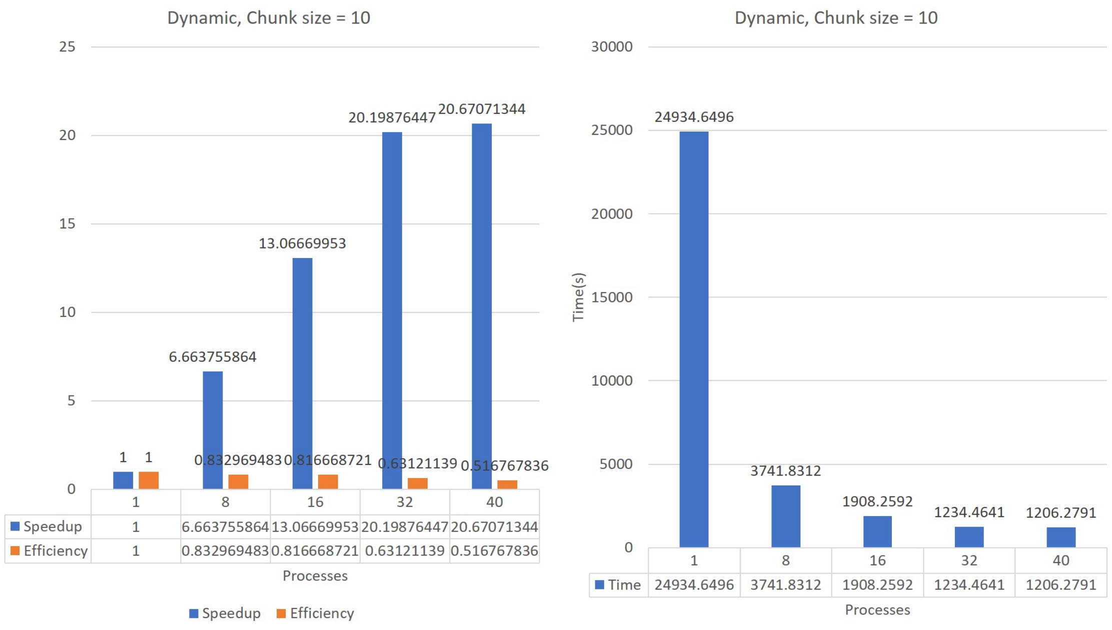
Figure 1: Speedup and Efficiency comparison for Dynamic Scheduler



	1	8	16	32	40
■ Speedup	1	5.757010627	11.2054853	19.17503466	23.71852558
■ Efficiency	1	0.719626328	0.700342831	0.599219833	0.59296314

	1	8	16	32	40
■ Time	101123.237	17565.2336	9024.4406	5273.6925	4263.4706

Figure 2: Speedup and Efficiency comparison for Static Scheduler



	1	8	16	32	40
■ Speedup	1	6.663755864	13.06669953	20.19876447	20.67071344
■ Efficiency	1	0.832969483	0.816668721	0.63121139	0.516767836

	1	8	16	32	40
■ Time	24934.6496	3741.8312	1908.2592	1234.4641	1206.2791

Figure 3: Speedup and Efficiency comparison for Dynamic Scheduler, Chunk Size = 10

It can be seen from the figure 1 and 2 that speedup increased with the number of processors where as trend for the efficiency is exactly opposite. Also, the dynamic scheduler gives comparatively higher speedup than the static scheduler as number of processors increases. Moreover, computation time for the dynamic scheduler is low than the static scheduler.

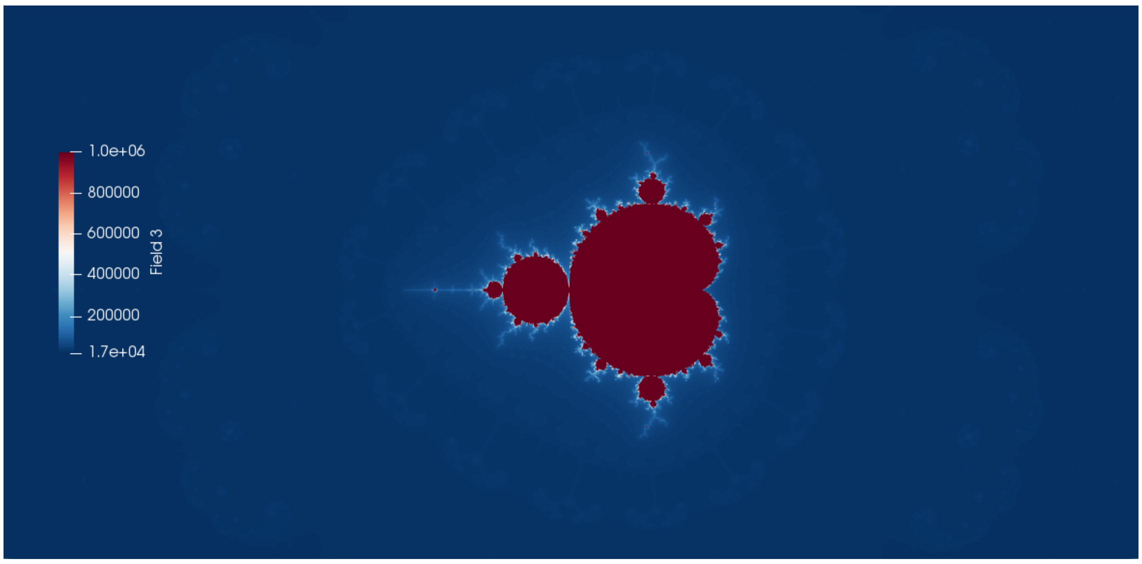
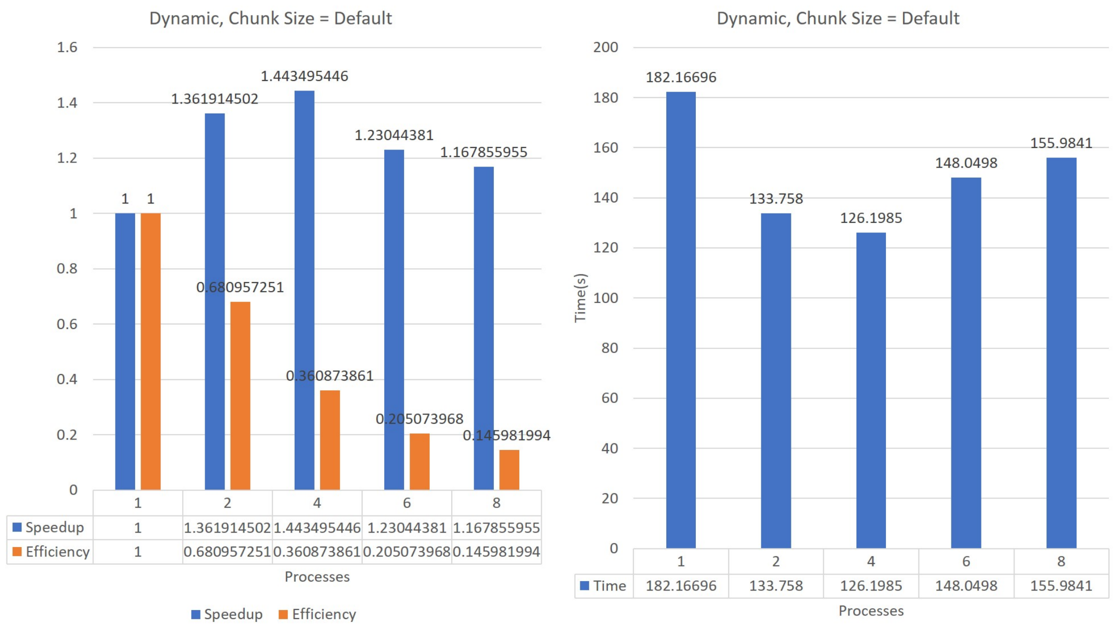


Figure 4: Mandelbrot Set

2 Gauss-Seidel

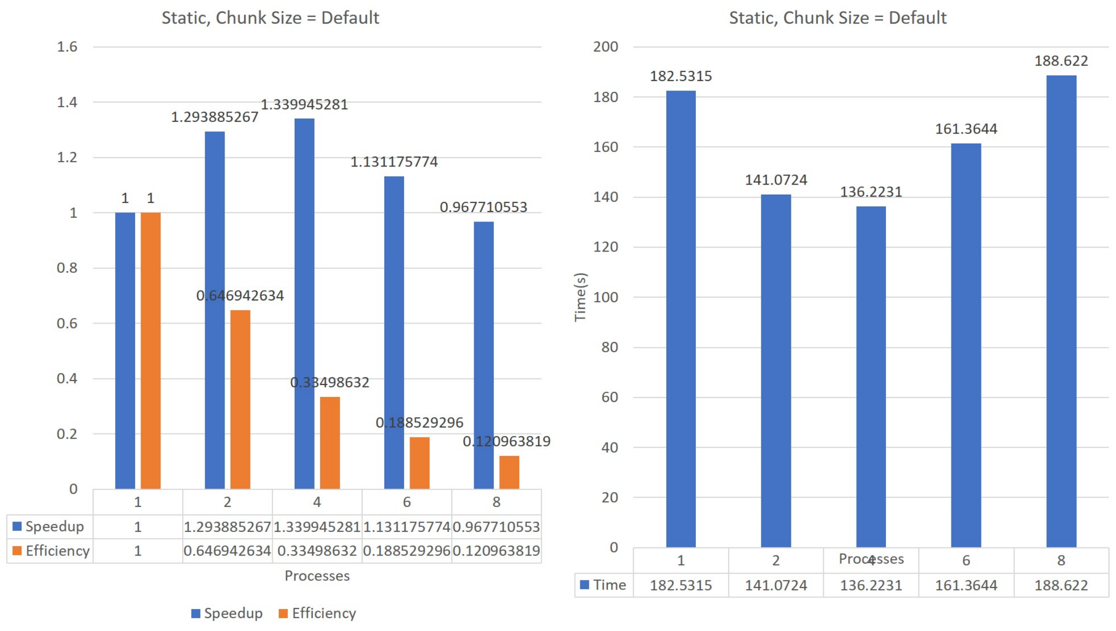
The computation for Gauss-Seidel was performed on INTEL 7700HQ (2.8 GHz) quad core processor that can be hyper-threaded to 8. It took approximately 182 secs on a single core for $100 \times 100 \times 100$ domain.



	1	2	4	6	8
Speedup	1	1.361914502	1.443495446	1.23044381	1.167855955
Efficiency	1	0.680957251	0.360873861	0.205073968	0.145981994

	1	2	4	6	8
Time	182.16696	133.758	126.1985	148.0498	155.9841

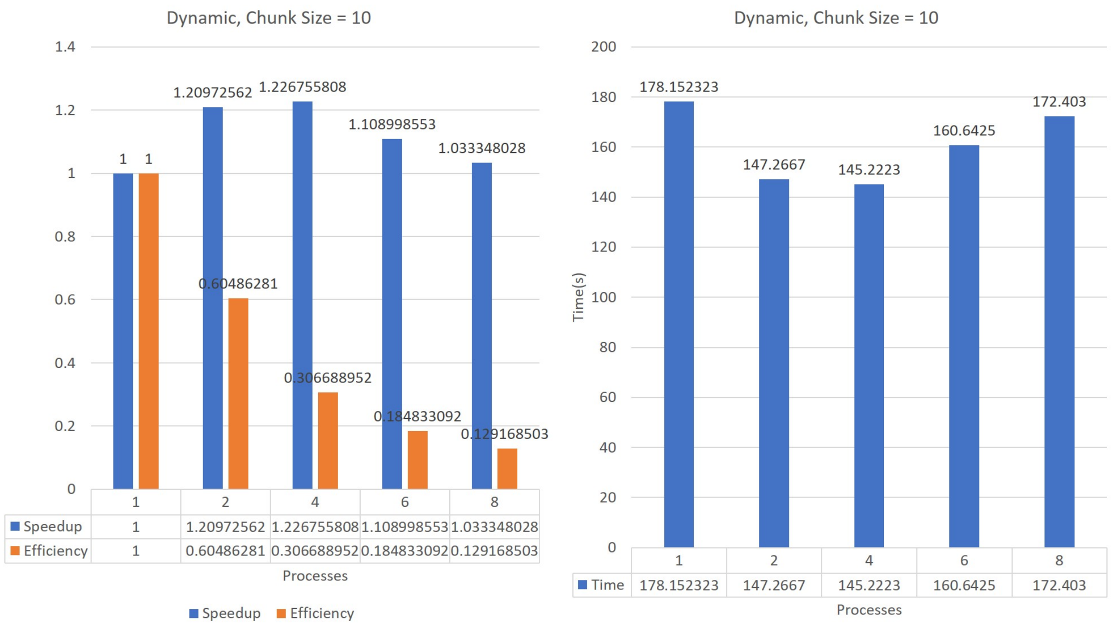
Figure 5: Speedup and Efficiency comparison for Dynamic Scheduler (Chunk Size = Default)



	1	2	4	6	8
Speedup	1	1.293885267	1.339945281	1.131175774	0.967710553
Efficiency	1	0.646942634	0.33498632	0.188529296	0.120963819

	1	2	4	6	8
Time	182.5315	141.0724	136.2231	161.3644	188.622

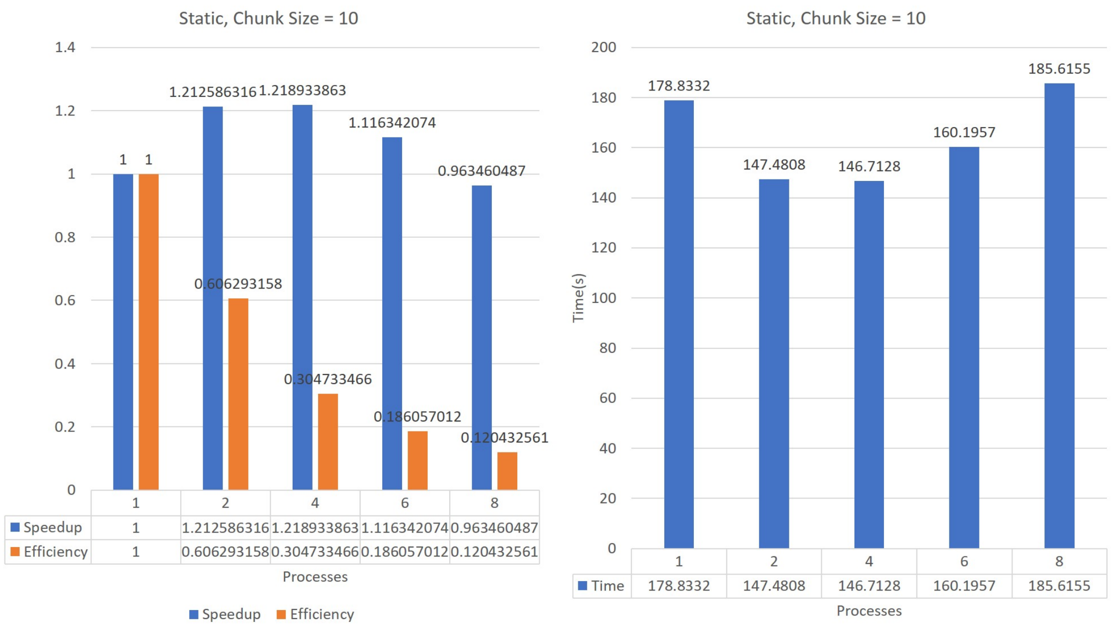
Figure 6: Speedup and Efficiency comparison for Static Scheduler (Chunk Size = Default)



	1	2	4	6	8
🟦 Speedup	1	1.20972562	1.226755808	1.108998553	1.033348028
🟧 Efficiency	1	0.60486281	0.306688952	0.184833092	0.129168503

	1	2	4	6	8
🟦 Time	178.152323	147.2667	145.2223	160.6425	172.403

Figure 7: Speedup and Efficiency comparison for Dynamic Scheduler (Chunk Size = 10)



	1	2	4	6	8
🟦 Speedup	1	1.212586316	1.218933863	1.116342074	0.963460487
🟧 Efficiency	1	0.606293158	0.3047334666	0.186057012	0.120432561

	1	2	4	6	8
🟦 Time	178.8332	147.4808	146.7128	160.1957	185.6155

Figure 8: Speedup and Efficiency comparison for Static Scheduler (Chunk Size = 10)

It can be seen from the figure 5, 6, 7, and 8 that the speedup increased till 4 processes after that it decreased because of hyper-threading, which means hyper-threading is not efficient in this case. Moreover, it is observed that the speedup and efficiency are better for default chunk size.

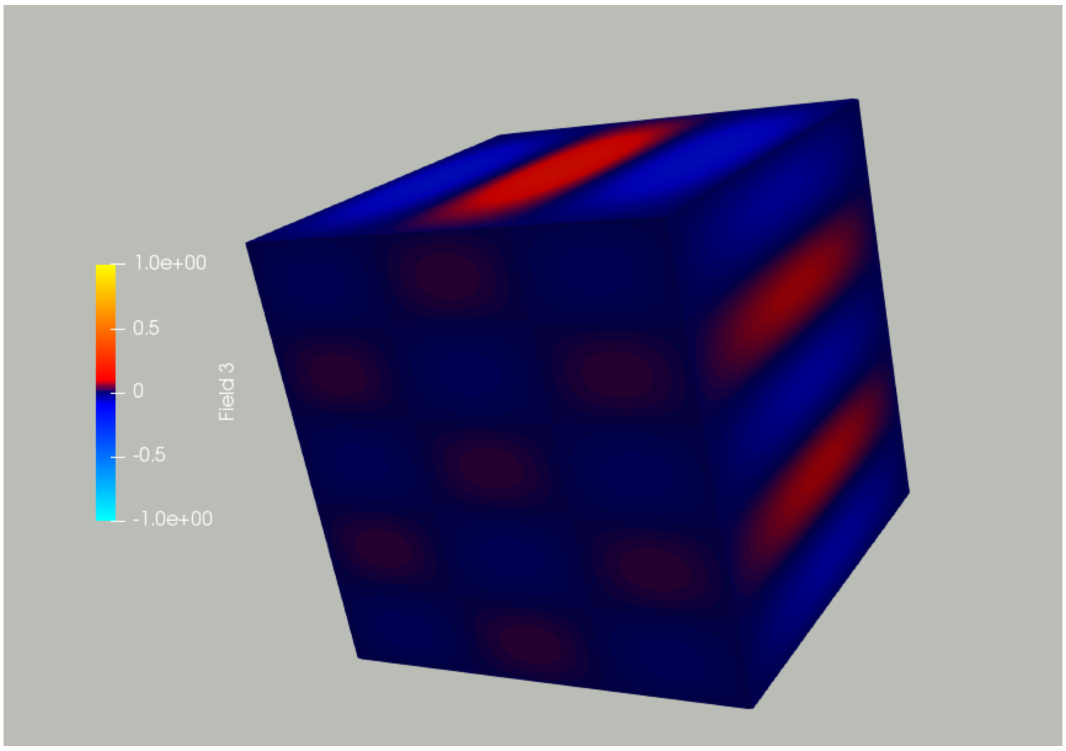


Figure 9: Gauss-Seidel solution

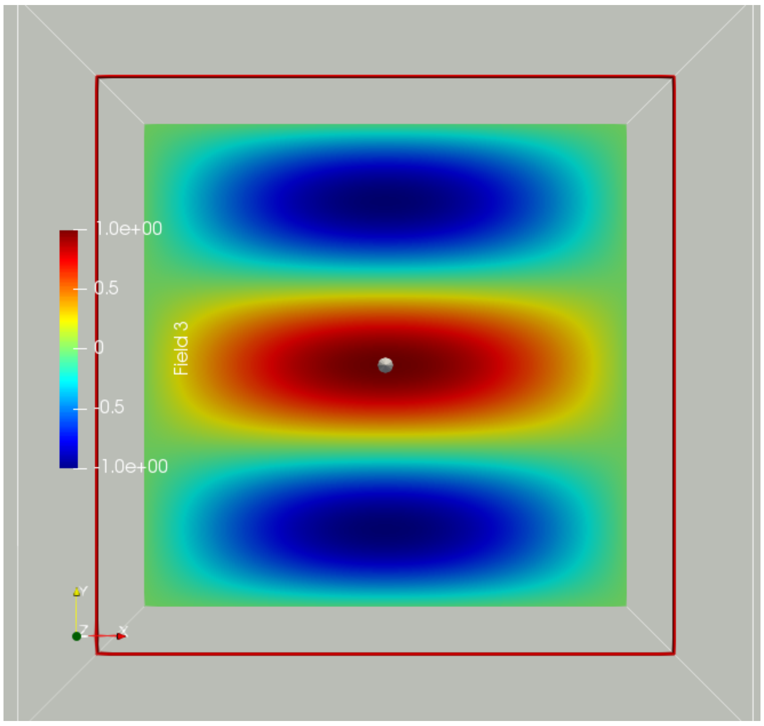
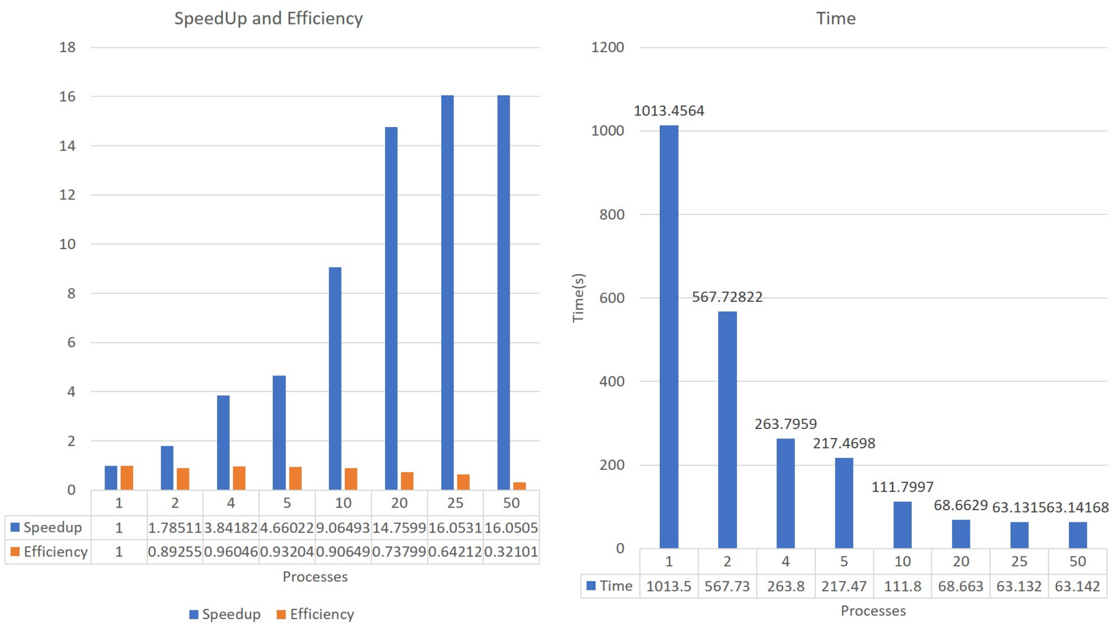


Figure 10: Gauss-Seidel solution cross section

3 Jacobi

The computation for Jacobi was performed on INTEL Xeon Gold 6130 (2.1 GHz) processor that has 32 cores. It took approximately 1013.5 secs on a single core for $100 \times 100 \times 100$ domain.



Processes	1	2	4	5	10	20	25	50
🟦 Speedup	1	1.78511	3.84182	4.66022	9.06493	14.7599	16.0531	16.0505
🟧 Efficiency	1	0.89255	0.96046	0.93204	0.90649	0.73799	0.64212	0.32101

Processes	1	2	4	5	10	20	25	50
🟦 Time	1013.5	567.73	263.8	217.47	111.8	68.663	63.132	63.142

Figure 11: Speedup,Efficiency, and Time comparison

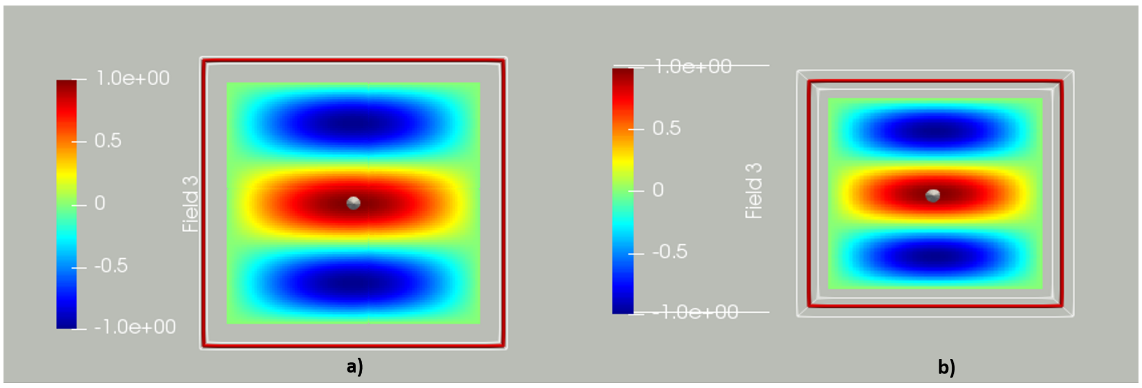


Figure 12: Jacobi cross section: a) Numerical Solution, b) Exact Solution

It can be observed from the figure 11 that the speedup increased till 25 processes and after that for 50 processes there is not a significant difference because of hyper threading.