CSIS7303 High-Performance Computing

Assignment 1

Part 1: Benchmarking Jacobi OpenMP on Belief

Fill in the blanks in yellow below with your measured execution times. Run 2 to 3 trials for each data point.
For p =1, run the sequential program for taking absolute speedup.

Measured Execution Times (Sec)

n \ p	1	2	4	5
200	14.811	8.932	6.280	5.995
	14.830	8.931	6.305	
		8.904	6.299	
300	58.825	33.429	20.102	18.139
	58.474	33.333	20.196	
400	153.828	83.152	46.912	40.552
	154.033	83.159	47.013	

I don't have a dedicated platform and only 5 CPU cores are available during my testing period. So I used 5 threads only. You should change it to 8 in your measurement.

Average Execution Times (Sec)

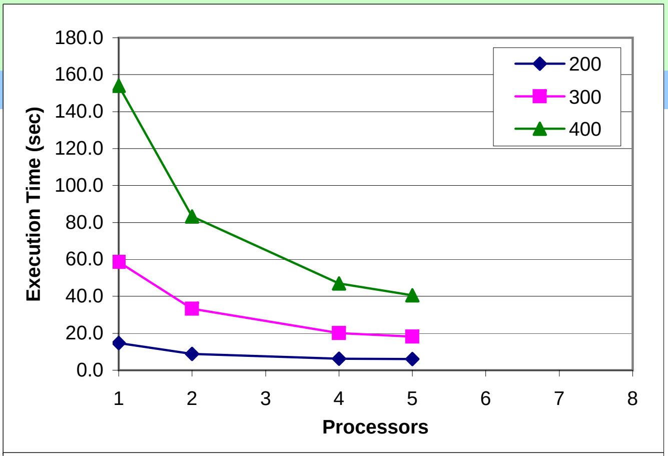
n \ p	1	2	4	5
200	14.820	8.922	6.295	5.995
300	58.649	33.381	20.149	18.139
400	153.930	83.155	46.963	40.552

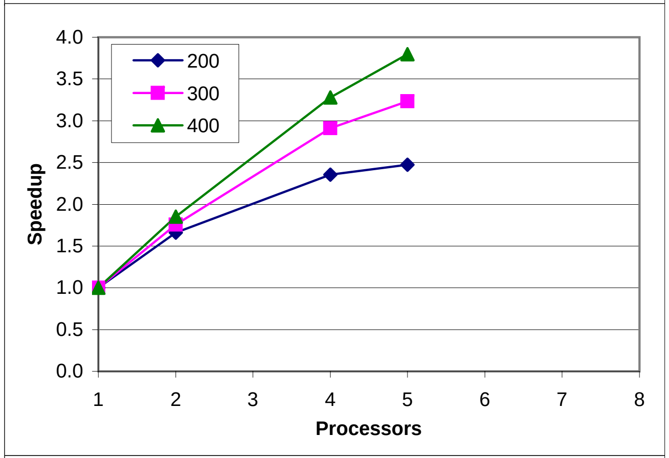
Speedups

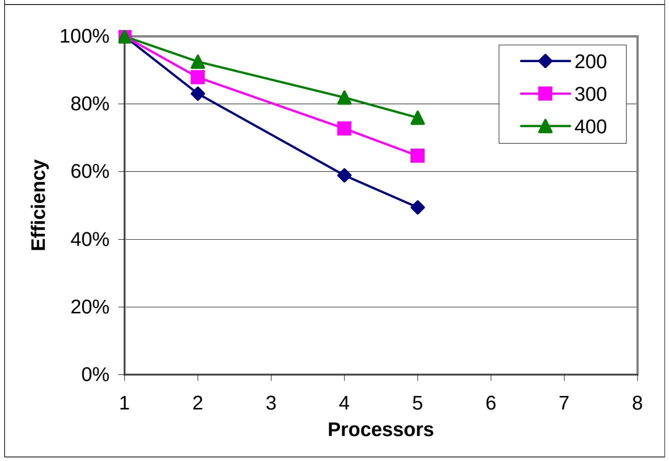
n \ p	1	2	4	5
200	1.000	1.661	2.354	2.472
300	1.000	1.757	2.911	3.233
400	1.000	1.851	3.278	3.796

Efficiencies

n \ p	1	2	4	5
200	100.00%	83.05%	58.86%	49.45%
300	100.00%	87.85%	72.77%	64.67%
400	100.00%	92.56%	81.94%	75.92%







CSIS7303 High-Performance Computing

Assignment 1

Part 2: Benchmarking Jacobi MPI on Gideon

Fill in the blanks in yellow below with your measured execution times. Run 2 to 3 trials for each data point.
For p =1, run the sequential program for taking absolute speedup.

Measured Execution Times (Sec)

n \ p	1	2	4	8	16
200	35.376	36.129	24.134	30.223	32.841
	34.155	37.807	23.957	31.654	32.492
300	208.089	182.082	102.744	70.399	68.379
	209.022	183.350	105.324	73.633	68.968
400	439.940	471.913	303.977	150.299	123.490
	442.994	471.721	294.990	154.223	127.731

This is an average-quality implementation, using standard send/recv and the simplest coding. It is not quite scalable. The best speedup achieved is just about 3.5.

Average Execution Times (Sec)

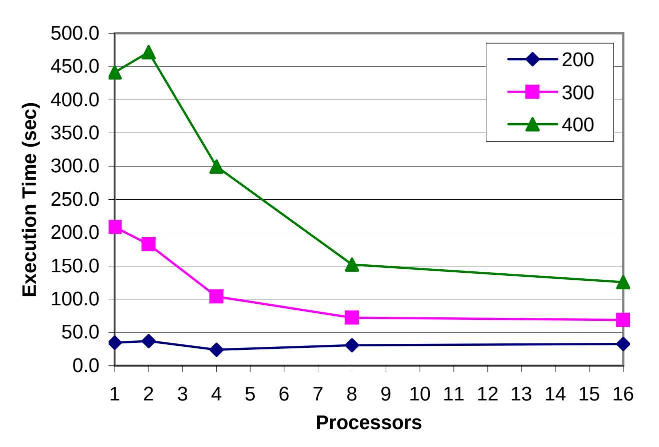
n \ p	1	2	4	8	16
200	34.765	36.968	24.045	30.938	32.667
300	208.556	182.716	104.034	72.016	68.673
400	441.467	471.817	299.483	152.261	125.610

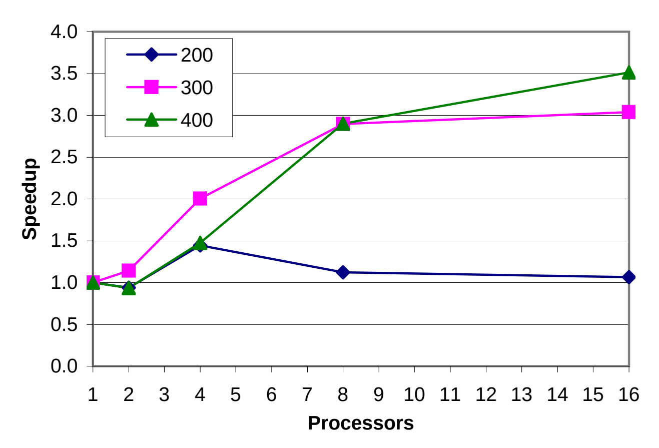
Speedups

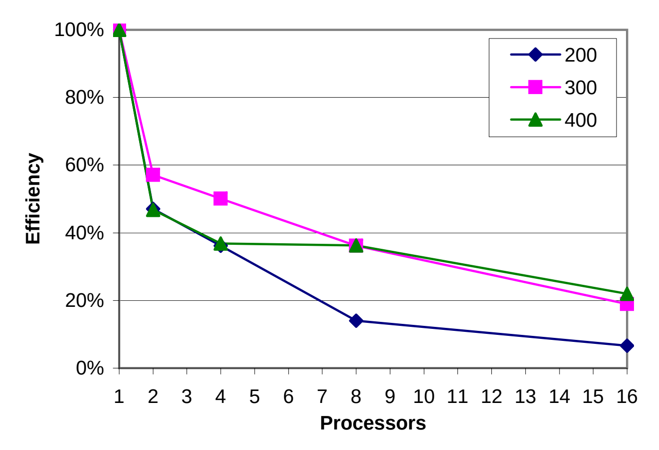
n \ p	1	2	4	8	16
200	1.000	0.940	1.446	1.124	1.064
300	1.000	1.141	2.005	2.896	3.037
400	1.000	0.936	1.474	2.899	3.515

Efficiencies

n \ p	1	2	4	8	16
200	100.00%	47.02%	36.15%	14.05%	6.65%
300	100.00%	57.07%	50.12%	36.20%	18.98%
400	100.00%	46.78%	36.85%	36.24%	21.97%







CSIS7303 High-Performance Computing

Assignment 1

Part 2: Benchmarking Jacobi MPI on Gideon

Fill in the blanks in yellow below with your measured execution times. Run 2 to 3 trials for each data point.
For p =1, run the sequential program for taking absolute speedup.

Measured Execution Times (Sec)

n \ p	1	2	4	8	16
200	35.376	44.647	19.474	16.411	16.429
	34.155	45.859	18.340	16.373	16.562
300	208.089	185.621	100.288	43.062	36.146
	209.022	181.584	98.634	41.216	35.942
400	439.940	461.129	287.509	124.802	63.898
	442.994	455.427	281.752	124.483	64.007

This is an optimized implementation, using non-blocking send/recv and more coding. It is more scalable. The best speedup achieved is just about 6.9.

Average Execution Times (Sec)

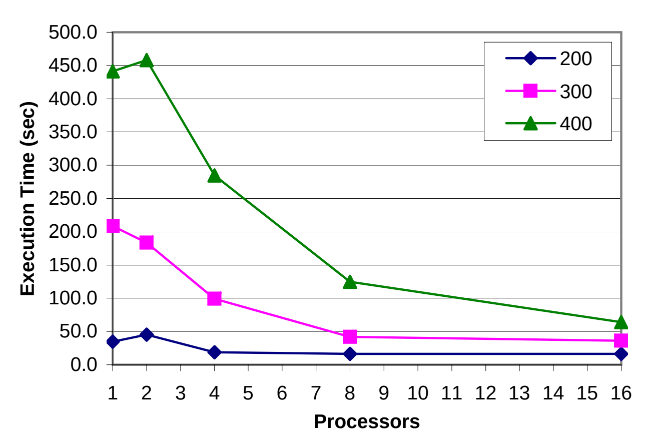
n \ p	1	2	4	8	16
200	34.765	45.253	18.907	16.392	16.496
300	208.556	183.603	99.461	42.139	36.044
400	441.467	458.278	284.631	124.642	63.953

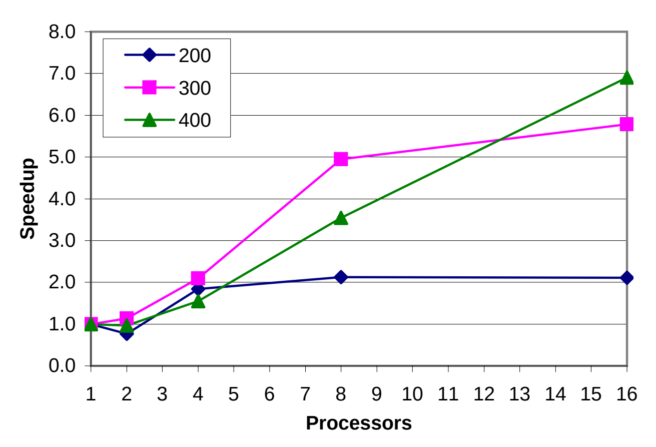
Speedups

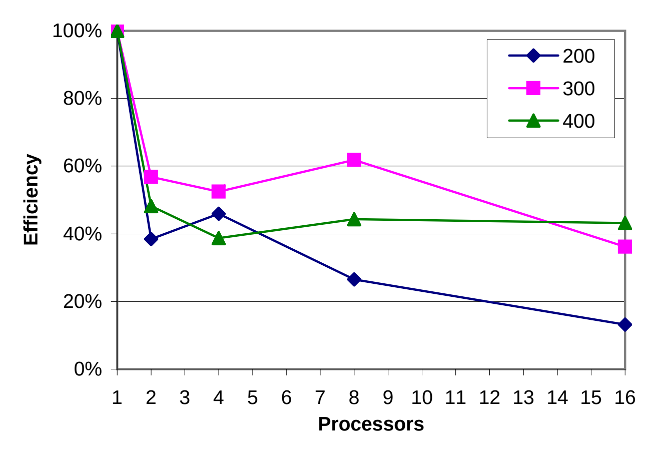
n \ p	1	2	4	8	16
200	1.000	0.768	1.839	2.121	2.108
300	1.000	1.136	2.097	4.949	5.786
400	1.000	0.963	1.551	3.542	6.903

Efficiencies

n \ p	1	2	4	8	16
200	100.00%	38.41%	45.97%	26.51%	13.17%
300	100.00%	56.80%	52.42%	61.87%	36.16%
400	100.00%	48.17%	38.78%	44.27%	43.14%





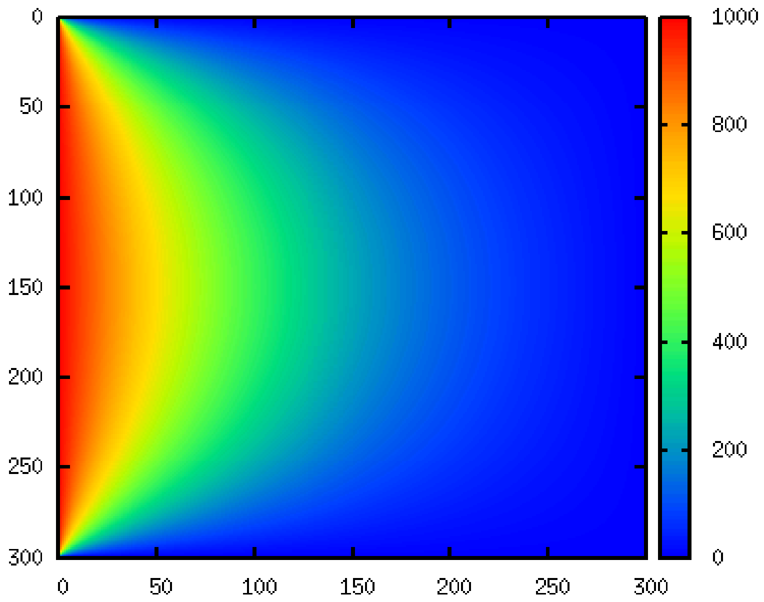


CSIS7303 High-Performance Computing

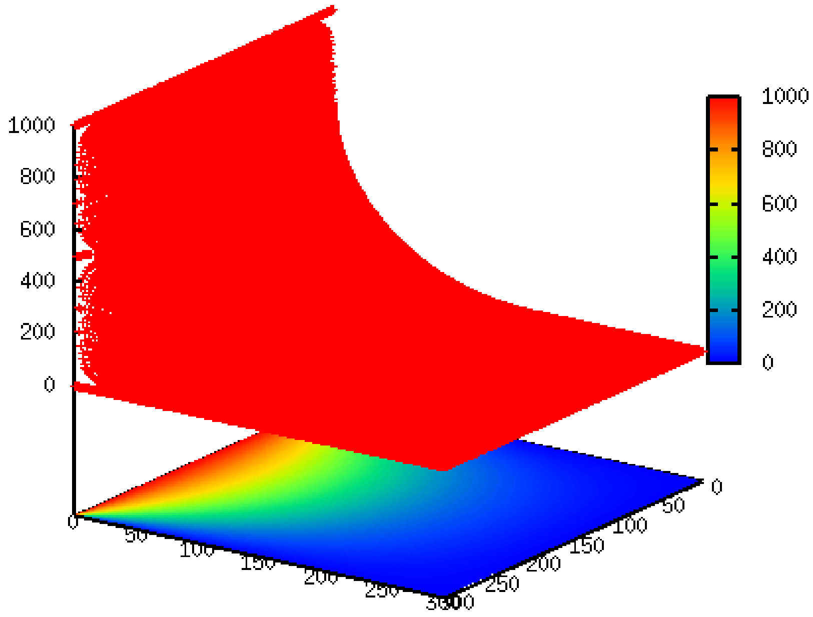
Assignment 1

Sample Color Map or 3D Plot by Gnuplot

You need not include plots in your hand-in. This is just for your reference.



0
50
100
150
200
250
300
0
50
100
150
200
250
300
1000
800
600
400
200
0



1000
800
600
400
200
0
0
50
100
150
200
250
300
0
50
100
150
200
250
300
1000
800
600
400
200
0