Parallel Computing Lab1

Chuanyang Jin

April 12, 2024

Times (seconds):

N	1,000	10,000	100,000	1,000,000	10,000,000
# processes 1	3.6e-05	3.3e-04	4.1e-03	7.1e-02	1.3
2	7.3e-05	3.7e-04	3.1e-03	4.9e-02	8.7e-01
5	1.8e-04	5.7e-04	3.6e-03	4.1e-02	5.6e-01
10	3.2e-04	7.9e-04	4.2e-03	3.9e-02	4.4e-01

Speedup:

N	1,000	10,000	100,000	1,000,000	10,000,000
# processes 1	1.0	1.0	1.0	1.0	1.0
2	0.49	0.89	1.3	1.4	1.5
5	0.2	0.58	1.1	1.7	2.3
10	0.11	0.42	0.98	1.8	3.0

Efficiency:

N	1,000	10,000	100,000	1,000,000	10,000,000
# processes 1	1.00	1.00	1.00	1.00	1.00
2	0.25	0.45	0.66	0.72	0.75
5	0.04	0.12	0.23	0.35	0.46
10	0.01	0.04	0.10	0.18	0.30

a) What is the pattern that you see in speedup as the number of processes increases, for different problem sizes?

When the problem size is large, the speedup **increases** as the number of processes increases.

When the problem size is small, the speedup **decreases** as the number of processes increases.

b) Explain why we get the pattern you explained in the question above.

When the problem size is large, the speedup increases as the number of processes increases because the computational work can be effectively distributed among the processes, leading to faster execution times, while the overhead is relatively minor.

When the problem size is small, the speedup decreases as the number of processes increases because the communication and coordination overhead of multiple processes can outweigh the benefits gained from parallel computing. In this scenario, communication (e.g., MPI_Reduce) becomes costly.

c) As the number of processes increases, how is the efficiency affected, for different problem sizes?

For different problem sizes, the efficiency **decreases** as the number of processes increases.

When the problem size is small, the decrease is more significant.

d) Justify your answer to question c above.

As the number of processes increases for a fixed problem size, each process has less work to do, so the efficiency decreases due to increased communication and coordination overhead among processes. When the problem size is small, this overhead can become dominant, causing a more significant decrease in efficiency.