CSE 5441 Programming Assignment 3

Vineeth Thumma (thumma.6)

1) Avg. time for point-point communication & Performance for blocking MPI communication

Message Size	***time(within-node) in sec***	***time(across-node) in sec***	***Performance (within-node) in Mbytes/sec***	***Performance (across-node) in Mbytes/sec***
1	0.000001	0.000003	5.853432	2.359770
8	0.000003	0.000005	19.091854	11.751373
64	0.000003	0.000004	153.614503	114.859553
512	0.000007	0.000010	557.892611	409.716470
4096	0.000018	0.000027	1844.196933	1229.208491
32768	0.000064	0.000092	4111.908237	2853.892544
256K	0.000514	0.000690	4076.116826	3038.243738
1M	0.002302	0.002816	3644.616391	2978.540191

Avg. time for point-point communication & Performance for non-blocking MPI communication

Message Size	***time(within-node) in sec***	***time(across-node) in sec***	***Performance (within-node) in Mbytes/sec***	***Performance (across-node) in Mbytes/sec***
1	0.000002	0.000011	3.926752	0.743444
8	0.000002	0.000004	41.276414	17.137636
64	0.000003	0.000005	171.720222	111.613235
512	0.000004	0.000010	1115.236759	423.347997
4096	0.000021	0.000025	1577.83543	1321.41174
32768	0.000067	0.000094	3930.530393	2800.389990
256K	0.000528	0.000703	3974.679335	2983.22299
1M	0.002334	0.002837	3593.883627	2956.509494

From the above tables, we observe that the time taken for the point-point communication is less for the processors with-in the same node as compared to the processors across the nodes in both blocking and non-blocking communication. So performance will be better for with-in node communication. The reason for this is that there might be network congestion when we are communicating across nodes.

But if we compare across blocking and non-blocking communication, the time taken for non-blocking is expected to be less as compared to the blocking-communication as there is no immediate blocking on the send/receive buffer in non-blocking MPI. But sometimes non-blocking communication takes more time than blocking-communication. The reason for this might be that both blocking and non-blocking calls might be using different protocols for communication.

2)a)

1-D Blocking:

Time taken by the process in the middle (excluding the first & last) = $t_s + M*t_w$
where M = 2*N*2 = 4*N (as there will be two Send & two Recv for each process)
So communication cost: $t = t_s + (4N)*t_w$ for each process
Total communication cost = $P*(t_s + (4N)*t_w)$ for all 'P' processes (Upper Bound)

2-D Blocking:

Time taken by the process = $t_s + M*t_w$
where M = $4*(N/\sqrt{P})*2 = 8*(N/\sqrt{P})$ (as there will be four Send & four Recv for each matrix per process)
So communication cost: $t = t_s + ((8*N)/\sqrt{P})*t_w$ for each process
Total communication cost = $P*(t_s + ((8*N)/\sqrt{P})*t_w)$ for all 'P' processes (Upper Bound)

Hence the 2-D scheme takes less time for data transfer, though the volume of data transfer is high because there is more simultaneous Send/Receive of data as compared to 1-D.

b)

No. of processors	***Performance (in GFLOPS)***
1	1.7
4	4.2
8	5.8
12	7.1
24	9.1
48	14.4
96	12.6

There is a variation in the above observed performance values as we run the code different times.

As we can see from the above table, there is a Performance Gain as we increase the no. of processors (degree of parallelism). Also, the performance decreased for 96 processors as compared to 48 processors. This might be due to the increase in overhead of communication between different nodes.