Kunal Shah(110974215) Gargi Saha(110923743)

Homework 3

Question 1

B. Running times for random initialization at every process for 1 process/node

Matrix size (n)	Node	Timings in seconds
1024	1	8.62421
1024	4	0.495257
1024	16	0.106541
2048	1	74.6548
2048	4	17.256
2048	16	0.989618
4096	1	724.02
4096	4	149.89
4096	16	34.504
8192	1	Beyond 2 hrs
8192	4	1557.96
8192	16	301.095
16384	1	Beyond 2 hrs
16384	4	Beyond 2 hrs
16384	16	3091.42

C. Running times for random initialization at every process for 16 process/node

Matrix size (n)	Node	Timings in seconds
1024	1	0.111397
1024	4	0.016273
1024	16	0.011333
2048	1	1.06056
2048	4	0.224833
2048	16	0.0392671
4096	1	34.7904
4096	4	2.11017
4096	16	0.441874
8192	1	305.329
8192	4	69.6595
8192	16	4.23398
16384	1	3314.46
16384	4	610.775
16384	16	140.358

E. Running times for random initialization at master node for 1 process/node

Matrix size (n)	Node	Scattering time (s)	Gathering time (s)	Total Running time (s)
1024	1	0.00727773	0.00376892	8.63526
1024	4	0.00689721	0.00476003	0.506914
1024	16	0.00966907	0.00653887	0.122749
2048	1	0.033217	0.0162468	74.7043
2048	4	0.0271478	0.0193529	17.3025
2048	16	0.0266609	0.0254152	1.04169
4096	1	0.142091	0.0704231	724.233
4096	4	0.114265	0.0811691	150.085
4096	16	0.0965581	0.109696	34.7102
8192	1	NA	NA	Running beyond 2hrs
8192	4	0.460106	0.317723	1558.73
8192	16	0.381746	0.426439	301.903
16384	1	NA	NA	Running beyond 2hrs
16384	4	NA	NA	Running beyond 2hrs
16384	16	1.641	22.0379	3115.1

F. Running times for random initialization at master node for 16 process/node

Matrix size (n)	Node	Scattering time (s)	Gathering time (s)	Total Running time (s)
1024	1	0.00496101	0.00398779	0.120346
1024	4	0.00797892	0.00577807	0.03003
1024	16	0.101708	0.0063262	0.119367
2048	1	0.0183971	0.0161271	1.09509
2048	4	0.0226288	0.022402	0.269864
2048	16	0.139276	0.0256219	0.204165
4096	1	0.0686281	0.072973	34.932
4096	4	0.0851009	0.108258	2.30353
4096	16	0.194483	0.116439	0.752796
8192	1	0.270096	0.236913	305.836
8192	4	0.327052	0.442186	70.4288
8192	16	0.464133	0.475664	5.17378
16384	1	1.06449	0.932691	3316.45
16384	4	1.27509	1.6322	613.683
16384	16	1.57876	5.46668	147.403

G. Explaining the findings

For 1B and 1E, we couldn't obtain the results for 8192 and 16384 as those running times would be beyond the 2 hours as for matrix multiplication those should be increased by factor of 8 at least as size increases by factor of 2, even though from observations it seems it might have taken more than 8 times the factor.

It seems scatter and gather has very less effect when compared to overall for computation time for large matrix and multiple nodes and processors. Thus, can be used for initializing at one node and then split across available resources.

But we can see that gathering time increases for large matrix compared. Thus, we can say that as matrix size will increase, considerable time might be spent on gathering.

Gathering time and scattering increases with the increase in nodes for same size. This is how should be the behavior. We also observe that gathering time is less for 16 process/node when compared to 1 process/node.

Question 2

B. Running times for random initialization at per process for 1 process/node and use 16 cores for shared memory using Open MP threads for shared memory.

Matrix size (n)	Node	Timings in seconds Shared memory algorithm - ikj	Timings in seconds Shared memory algorithm - kij
1024	1	0.296395	0.525063
1024	4	0.106234	0.318888
1024	16	**0.099359**	**0.146481**
2048	1	1.75709	2.86228
2048	4	0.639905	0.861353
2048	16	**0.139984**	**0.210625**
4096	1	12.9068	24.9924
4096	4	3.68398	5.63658
4096	16	**1.08013**	**0.343328**
8192	1	98.3091	204.056
8192	4	27.34	50.5202
8192	16	**4.13755**	**2.56383**
16384	1	747.46	1602.38
16384	4	218.304	415.089
16384	16	**23.2267**	**31.0943**

Matrix size (n)	Node	Scattering Time	Gathering Time	Timings in seconds Shared memory algorithm - ikj	Scattering Time	Gathering Time	Timings in seconds Shared memory algorithm - kij
1024	1	0.007689	0.00367284	0.307757	0.00767612	0.00376892	0.536508
1024	4	0.00708008	0.004673	0.117987	0.00707507	0.00471306	0.330676
1024	16	**0.00920296**	**0.01492**	**0.123482**	**0.00940681**	**0.00598097**	**0.161869**
2048	1	0.0335619	0.015758	1.80641	0.0341041	0.0161819	2.91257
2048	4	0.0273027	0.0185342	0.685742	0.0281219	0.018889	0.908364
2048	16	**0.0212979**	**0.018595**	**0.179877**	**0.0232651**	**0.0222018**	**0.256092**
4096	1	0.147224	0.070329	13.1244	0.142707	0.0678871	25.203
4096	4	0.112866	0.07515	3.872	0.11473	0.076478	5.82778
4096	16	**0.0679309**	**0.07199**	**1.22005**	**0.06881**	**0.0778859**	**0.490024**
8192	1	0.586938	0.297883	99.1939	0.601377	0.294034	204.952
8192	4	0.470758	0.329288	28.1401	0.463417	0.338682	51.3223
8192	16	**0.282869**	**0.305289**	**4.7257**	**0.271409**	**0.298202**	**3.13345**
16384	1	2.36919	1.18236	751.011	2.34159	1.17485	1605.89
16384	4	1.83334	1.39619	221.533	1.82345	1.28432	418.197
16384	16	**1.09873**	**1.08783**	**25.4133**	**1.09441**	**1.09047**	**33.2792**

Note: Readings in block are performed on comet to get rough idea as the stampede waiting time is more than 2 days and most of the readings were done on stampede in development queue except for 16 nodes.

C. Comparing the results

The time for executing task in shared + distributed (SD) with scattering and gathering (SG) is much better than the only distributed (D) in case where matrix size is more and more number of processors.

Thus we observe that SD is more expensive than the D when matrix size is 1024 and 2048. Even for 4096 and 4 nodes SD is more expensive. Thus, threading cost seems to be more than the message passing in case of less matrix size.

We can also observe that even though overall time for scattering and gathering is less in SD model against D model, but the time spent on scattering and gathering for SD has larger contribution to total time w.r.t. overall time when compared with only D.

We also observe that the reduction ratio in overall time for SD over D is dependent on the total number of MPI task and Nodes.

References:

1. http://stackoverflow.com/questions/5901476/sending-and-receiving-2d-array-over-mpi
2. https://portal.tacc.utexas.edu/c/document_library/get_file?uuid=2d346261-1555-45ae-b404-72ff289e2986&groupId=13601
3. http://cseweb.ucsd.edu/classes/fa12/cse260-b/Lectures/Lec13.pdf
4. http://mpitutorial.com/tutorials/mpi-scatter-gather-and-allgather/
5. Lecture Slides