Jeremia Muhia
jm5763
Lab 1
Parallel Computing

System Info:

- Crunchy1
- 64 Cores
- 64 Threads

Table 1:

64 Cores	1	2	4	8	16	32	64
1	0.756	0	0	0	0	0	0
2	0.774	0.742	0	0	0	0	0
4	7.13E-01	0.748	7.94E-01	0	0	0	0
8	0.722	0.75	7.57E-01	0.841	0	0	0
16	0.722	0.736	7.48E-01	0.813	0.896	0	0
32	0.734	0.733	7.46E-01	0.795	0.902	1.37	0
64	0.731	0.737	7.56E-01	0.82	0.901	1.394	2.57
128	0.725	0.771	7.73E-01	0.813	0.878	1.387	2.925
256	0.744	0.754	8.13E-01	0.827	0.898	1.392	2.352
512	0.801	0.911	1.05E+00	0.883	0.972	1.455	2.403
1024	1.102	1.039	1.22E+00	1.082	1.241	1.404	2.988
2048	2.631	2.21	2.68E+00	1.971	2.077	2.526	4.638
4096	9.349	7.985	6.60E+00	6.76	7.011	7.994	13.708

Table 2:

64 Cores	1	2	4	8	16	32	64
1	1	0	0	0	0	0	0
2	1	1.043126685	0	0	0	0	0
4	1	0.953208556	0.897984887	0	0	0	0
8	1	0.962666667	0.953764861	0.858501784	0	0	0
16	1	0.980978261	0.965240642	0.888068881	0.805803571	0	0
32	1	1.001364256	0.983914209	0.92327044	0.813747228	0.535766423	0
64	1	0.991858887	0.966931217	0.891463415	0.811320755	0.524390244	0.284435798
128	1	0.940337224	0.937904269	0.891758918	0.825740319	0.522710887	0.247863248
256	1	0.986737401	0.915129151	0.899637243	0.828507795	0.534482759	0.316326531
512	1	0.879253568	0.764312977	0.907134768	0.824074074	0.550515464	0.333333333
1024	1	1.060635226	0.901800327	1.018484288	0.887993554	0.784900285	0.368808568
2048	1	1.190497738	0.983183857	1.334855403	1.266730862	1.041567696	0.567270375
4096	1	1.170820288	1.416729808	1.382988166	1.333475966	1.169502127	0.682010505

Below 512 unknowns, speedup is rare. Using 64 cores leads to no speedup at all. Because I have only tested each combination of cores and unknowns once, this is likely lead to the results listed above. Had I completed more tests, the data would likely include more combinations where speedup was achieved. Nonetheless, because MPI_Scatter is dependent on one process sending data to all the other processes, this load imbalance could decrease speedup when the data is not large enough. That is, scattering a small amount of data requires a computationally expensive communication and when this data is small, the benefits of parallelization and co-working processes are not fully realized.

Further, MPI_Allgatherv is another computationally expensive communication. Speedup is achieved in larger data because as the size of the data increases, not only does the benefit of communication outweigh the cost, but the it also becomes more likely that all the cores are being fully utilized to capacity.