

ASSIGNMENT 1

Zi Wang

03/08/2015

1 Problem 1

When running on my Macbook, the timing information is as follows. Let p be the number of processors, N be the number of iterations and T be the time in seconds. If $p = 50$, $N = 100$, then $T = 2.09$. If $p = 100$, $N = 100$, then $T = 9.87$. If $p = 100$, $N = 200$, then $T = 17.12$. So the time used per communication is about 0.3 - 0.9 ms.

When running on crunchy1 and crunchy3, the timing information is as follows. If $p = 50$, $N = 100$, then $T = 1.46$. If $p = 100$, $N = 100$, then $T = 8.85$. If $p = 100$, $N = 200$, then $T = 19.12$. Time per communication is also about 0.3 - 0.9ms.

I use a 0.25 million entry double array to test the bandwidth. When set number of cores to be 10 and $N = 50$, the total time is 0.48s. When set number of cores to be 10 and $N = 100$, the total time is 0.89s. When set number of cores to be 15 and $N = 50$, the total time is 0.99s. When set the number of cores to be 15 and $N = 100$, the total time is 1.94s. When set the number of cores to be 20 and $N = 50$, the total time is 1.88s. When set the number of cores to be 20 and $N = 100$, the total time is 3.55s. Based on these data, it is estimated that the bandwidth is about 1 - 2 GB/s.

2 Problem 2

Assume N is large enough, the algorithm is strongly scalable within a single iteration. However, there is dependencies between iterations. Thus, when each processor takes charge of only a few entries, the time consumption cannot be improved by increasing the number of cores.

Gauss-Seidel smoother is harder to parallelize since within each iteration the next core will need the previous one to compute the entries of the vector. While each processor in Jacobi smoother is independent within a single iteration.