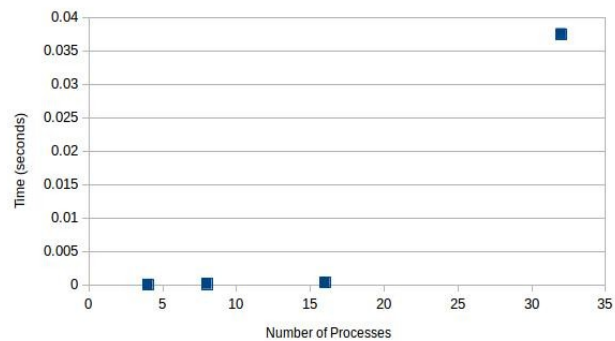


Allen Suner
CSCI 322
Assignment #3
02/10/14

The following is an analysis on three different solutions to finding the minimum and maximum of integer messages passed to different processes.

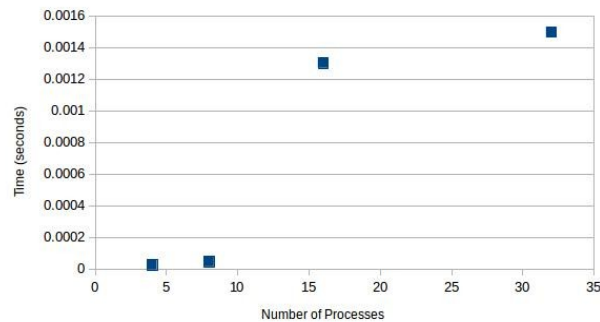
The first solution is performed like a broadcast and a reduce, where each process sends its value to process 0. Process 0 then computes the minimum and maximum and sends the values to each process. The following results were obtained when running with 'n' processes:

Number of processes	Time (seconds)
4	0.000024
8	0.000125
16	0.000341
32	0.037505



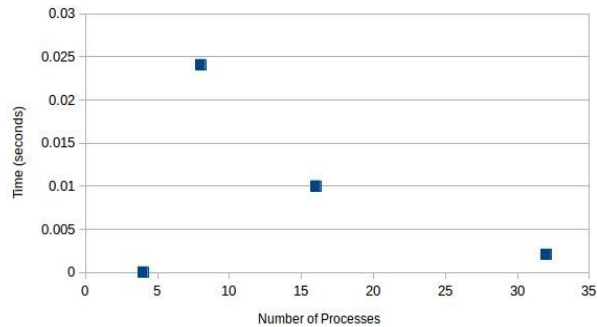
The second solution is performed like a broadcast and then another broadcast, where each process sends its value to every other process. Then each process computes the minimum and maximum values and then sends those to every other process. The following results were obtained when running with 'n' processes:

Number of processes	Time (seconds)
4	0.000026
8	0.000046
16	0.001302
32	0.001497



The third solution is performed in a ring-like fashion, where process 0 begins by sending two copies of its value to process 1, process 1 then computes the current maximum and minimum, then sends them to the next process. This procedure is repeated until we get to process 0 again, where we must go around the ring one final time to get the global max and min to each process. The following results were obtained when running with 'n' processes:

Number of processes	Time (seconds)
4	0.000033
8	0.024094
16	0.010017
32	0.002086



It is clear that no solution was clearly the best in all circumstances. All solutions performed relatively well when run with a limited number of processes ($< \sim 30$), but quickly began lagging when more processes were added. The third solution, however, performed better with more processes and slightly worse with a limited number of processes.

INSTRUCTIONS ON HOW TO COMPILE AND RUN

Each source file is labeled solution1.c, solution2.c, and solution3.c respective to the order presented in the assignment. To compile these programs run the following command in a unix shell:

`mpicc -o (name of executable) (name of source file)`

In order to run the executable run the following command in a unix shell:

`mpiexec -n (number of processes) ./(name of executable)`