

Homework 10

Josh Bradt

April 28, 2016

Rather than running the provided code, I chose to write my own version of it. I ran this code using a matrix of dimension 16384 by 16384 with striping factors of 0, 16, and 32, as instructed. I ran on Blue Waters, compiled my code with the default Cray compiler, and requested 64 nodes to run on, for a total of 1024 processes (16 per node). The three cases were run as separate jobs, so it is likely that they ran on different sections of the machine.

The results of running the code are shown in Figure 1 and Table 1. For all cases, the write times for the collective method were 2 to 3 orders of magnitude faster than the independent method. The close times were 5 orders of magnitude faster for the collective method than the independent method, though that result is slightly suspect since I was getting some I/O errors during the calls to the independent methods. (That being said, they still appeared to write the output to disk, so I'm not sure what was going on.)

Striping the files improved the write speeds by between 0.5 and 1 order of magnitude, so this is clearly an important optimization to make in an I/O-heavy application. Striping did, however, increase the amount of time taken to close the file after the collective write operation, but this increase is trivial when compared to the amount of time spent writing to the file, so it doesn't seem like a big problem.

Num. Stripes	Method	Write Time [s]	Close Time [s]	Rate [B/s]
0	Collective	6.123	2.172×10^{-3}	3.507×10^8
0	Independent	9.209×10^2	9.070×10^2	2.332×10^6
16	Collective	5.403×10^{-1}	2.227×10^{-3}	3.975×10^9
16	Independent	2.057×10^2	1.755×10^2	1.044×10^7
32	Collective	5.747×10^{-1}	4.449×10^{-3}	3.737×10^9
32	Independent	3.194×10^2	3.009×10^2	6.723×10^6

Table 1: Output of the code for the given parameters.

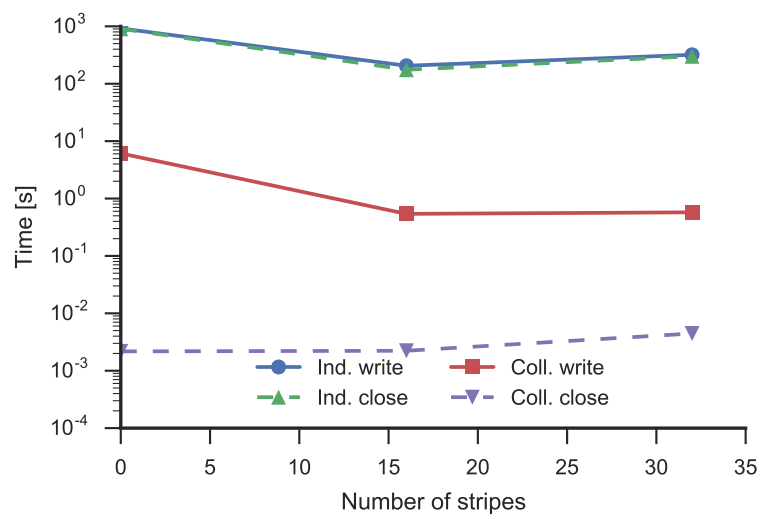


Figure 1: The time taken for each set of parameters. Both the time to write to the file and the time to close the file are shown.