

Edwin Tomy George

CS 1190 Parallel Computing

Instructor: David Pruitt

Friday, March 4th, 2022

Assignment IV Report: Map Reduce using MPI

Issues, Solutions & Time Management

During the completion of the assignment, there were issues with recording the time. This was solved by using `time.clock_gettime(time.CLOCK_MONOTONIC_RAW)` instead of `time.time()`. There was also some confusion present on which section of the program would go on the *if rank = 0*: section, causing a never-ending program, but that was quickly identified.

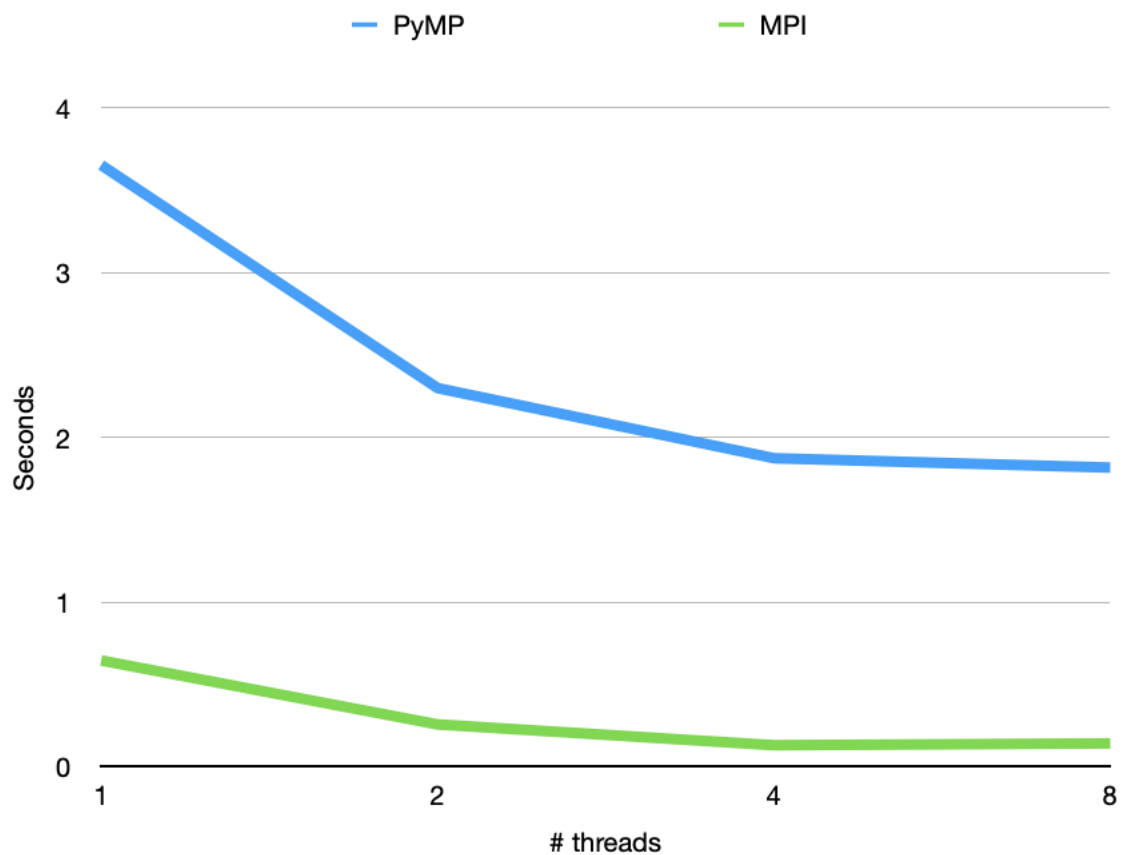
Overall, it took me around 3 hours to complete the assignment, from which 1 hour was spend debugging and half an hour on the report.

Number of Threads-Based Performance

The performance increased greatly from one-thread to two-threads, and from two-threads to four. Once we add more threads, the performance actually decreases. The increase time may be due because of increased communication and data sending between the threads. This conforms a more significant portion of the time, since the algorithm has improved total time spent in comparison to the last rendition of the project. The improvement between MPI and PyMP may be due more to algorithm change, since now we divide the tasks first by files instead of words.

# Threads	Time with PyMP (s)	Time with MPI (s)
1	3.6521	0.6453

2	2.2988	0.2572
4	1.8730	0.1309
8	1.8158	0.1420



Conclusions

In my opinion, MPI is cleaner and more intuitive than PyMP, and seems to avoid race conditions easier. With more cleaner tools, building more efficient algorithms is easier. This is shown in this assignment, where part of the improvement was due to a better algorithm.

Output of cplInfoDump.sh

model name : Intel(R) Core(TM) i5-1038NG7 CPU @ 2.00GHz

2 18 11