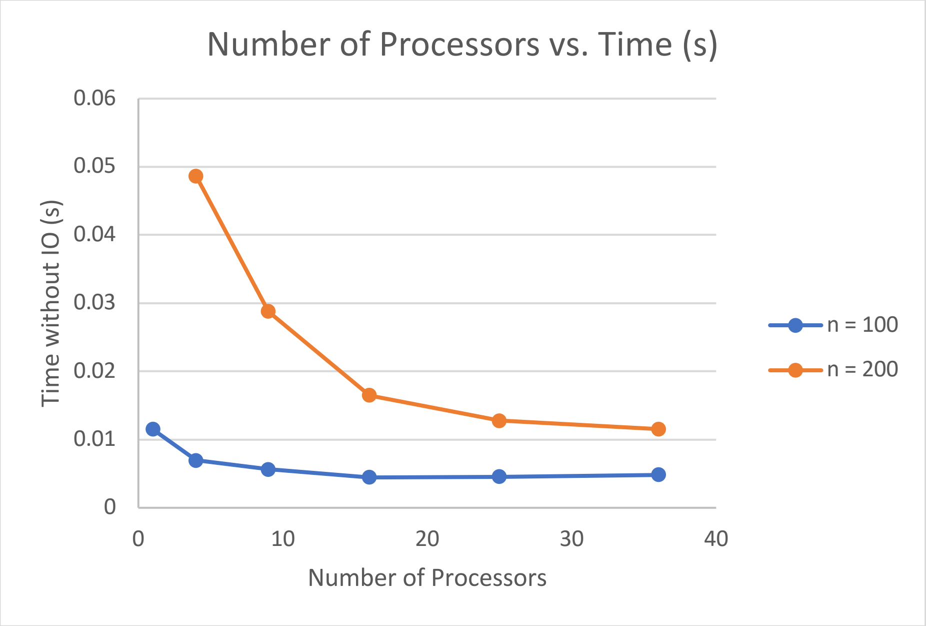
Sarah Anderson
Dr. Ligon
ECE 4730: Parallel Systems
November 7, 2020

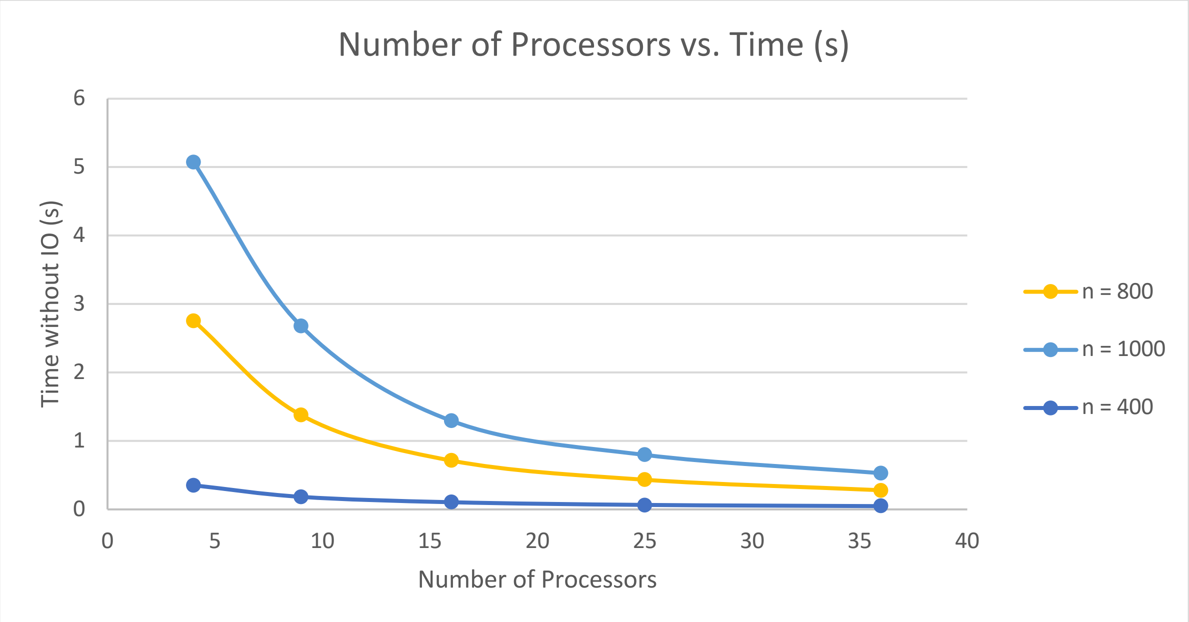
Project 2: Floyd's Algorithm

Purpose:

The purpose of this project is to create the parallel version of Floyd's algorithm using 2D block checkboard allocation of tasks and to compare the timing of multiple different sized matrices.

Results:





Note: I have two graphs for the different sizes of n because it made it hard to read the smallest n

n = 100		
Number of Processors	Total Time (w/ IO)	Floyds Time (w/o IO)
1	0.04437	0.0115
4	0.04169	0.00697
9	0.26832	0.00562
16	0.05076	0.00445
25	0.03573	0.00457
36	0.02979	0.00485
n = 200		
Number of Processors	Total Time (w/ IO)	Floyds Time (w/o IO)
1	0.16721	0.09166
4	0.15272	0.04865
9	0.0975	0.02877
16	0.08467	0.0165
25	0.06521	0.01276
36	0.05988	0.01154
n = 400		
Number of Processors	Total Time (w/ IO)	Floyds Time (w/o IO)
1	0.96117	0.68435
4	0.62335	0.35232
9	0.37302	0.18323
16	0.32362	0.10464
25	0.23651	0.06548
36	0.15516	0.04747
n = 800		
Number of Processors	Total Time (w/ IO)	Floyds Time (w/o IO)
1	5.76713	5.24254
4	3.18752	2.75656
9	1.70195	1.37865
16	1.06426	0.71256
25	0.76321	0.43298
36	0.49492	0.27985
n = 1000		
Number of Processors	Total Time (w/ IO)	Floyds Time (w/o IO)
1	11.46326	10.61746
4	5.64387	5.07661
9	3.16286	2.68015

16	1.75791	1.29429
25	1.0472	0.79612
36	0.79714	0.52869

Conclusion:

To conclude this project., in parallel, I found that as the bigger the matrix got, the longer it would take to run the program in parallel. Also, I found that as each matrix is running in parallel on bigger numbers of processors, the time tends to decrease and vice versa. Both of these things that I mentioned are to be expected when running a parallel program. I also compared the one processor numbers to the serial output time and the parallel time was less time when ran on one processor. I found that to make to best time out of running this program for any size matrix, it was best to use 36 processors running in parallel.