Sarah Anderson
Dr. Ligon
ECE 4730: Parallel Programming
December 1, 2020

Cannon's Algorithm

Purpose: The purpose of this project is to create the serial and parallel versions of Cannon's O(n^3) matrix multiply.

Results:

Table 1: Results of timing for number of processors (rows) vs. matrix size (columns)

	100x100	200x200	300x300	400x400	600x600	800x800	1200x1200	1600x1600
1	0.005015	0.042815	0.146011	0.448656	2.184803	8.406328	30.070262	72.907367
4	0.003652	0.011989	0.034395	0.081045	0.294256	0.793104	3.502784	9.058486
9			0.044128		0.185265		1.217336	
16	0.061830	0.058202	0.172123	0.130762	0.127405	0.342354	0.612720	1.931492
25	0.087202	0.168568	0.120624	0.192047	0.208661	0.250382	0.471839	1.020361

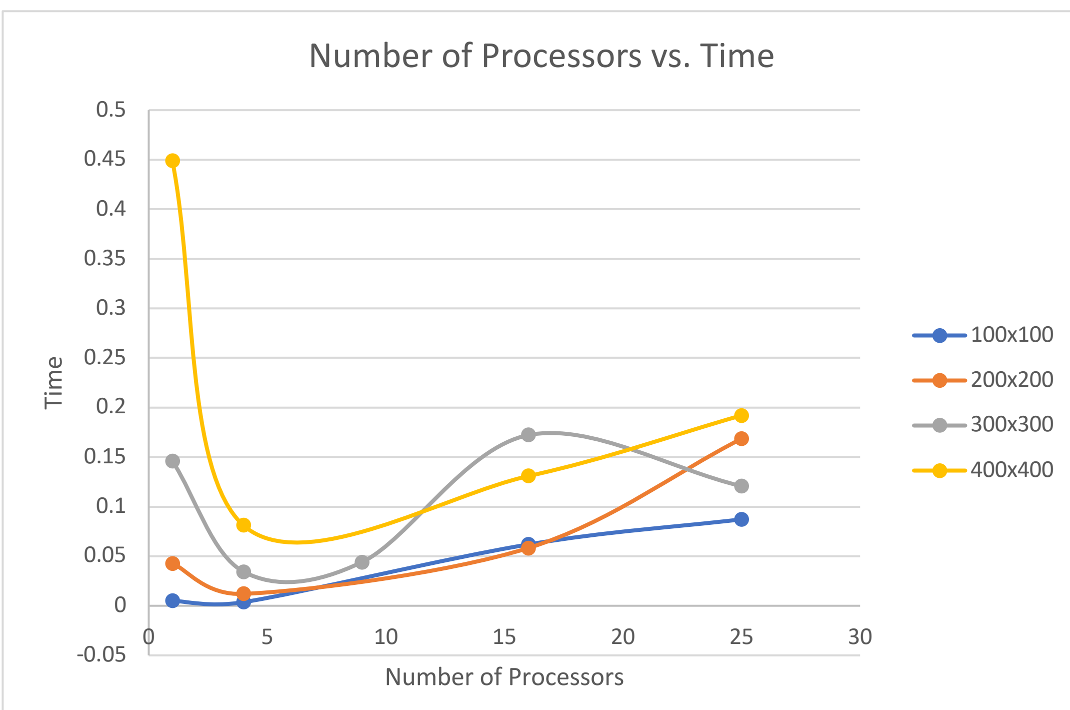

Figure 1: Timing for 100-400 sized matrices

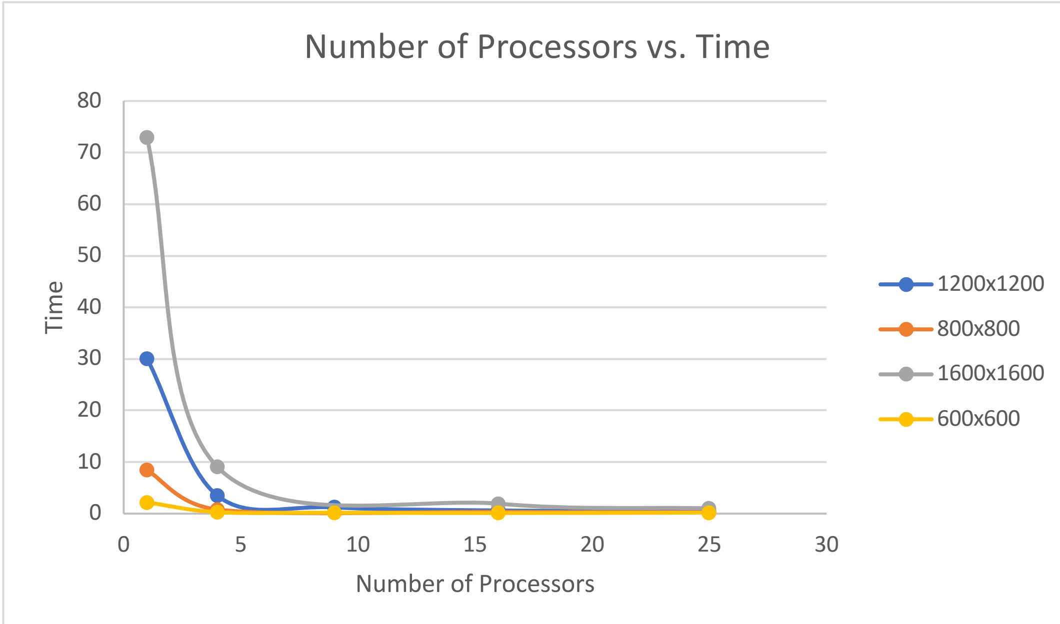

Figure 2: Timing for 600, 800, 1200, 1600 sized matrices

Table 2: Speedup Calculations (Current Block time – sequential time (i.e. 1 processor))

	100x100	200x200	300x300	400x400	600x600	800x800	1200x1200	1600x1600
4	-0.001363	-0.030826	-0.111616	-0.367611	-1.890547	-7.613224	-26.567478	-63.848881
9			-0.101883		-1.999538		-28.852926	
16	0.056815	0.015387	0.026112	-0.317894	-2.057398	-8.063974	-29.457542	-70.975875
25	0.082187	0.125753	-0.025387	-0.256609	-1.976142	-8.155946	-29.598423	-71.887006

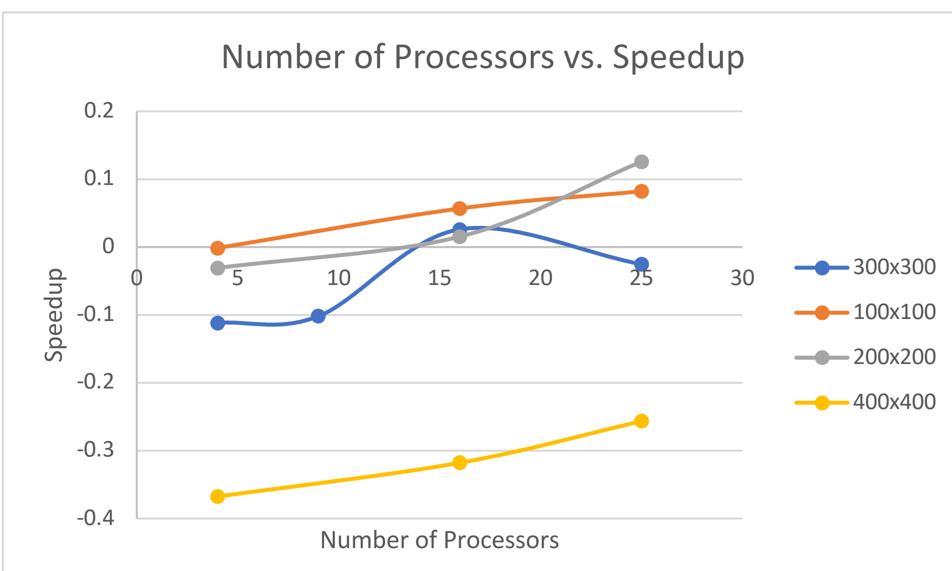

Figure 3: Speedup Graph for 100-400 matrices

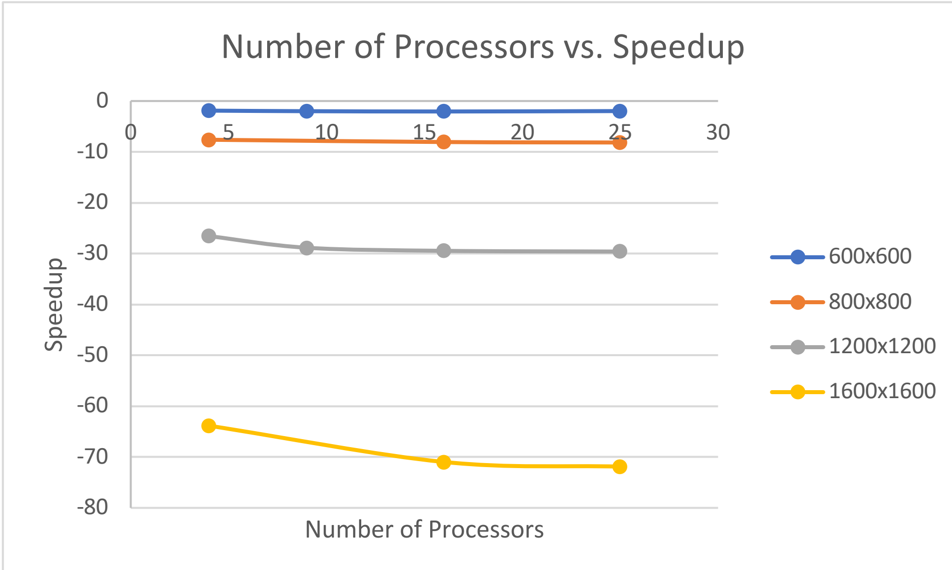

Figure 4: Speedup Graph for 600, 800, 1200, and 1600 matrices

Table 3: Percentage of Efficiency (Speedup / Number of Cores (or Processors) used)

	100x100	200x200	300x300	400x400	600x600	800x800	1200x1200	1600x1600
4	-0.034075	-0.77065	-2.7904	-9.190275	-47.263675	-190.3306	-664.18695	-1596.222
9			-1.1320333		-22.217089		-320.58807	
16	0.35509375	0.09616875	0.1632	-1.9868375	-12.858738	-50.399838	-184.10964	-443.59922
25	0.328748	0.503012	-0.101548	-1.026436	-7.904568	-32.623784	-118.39369	-287.54802

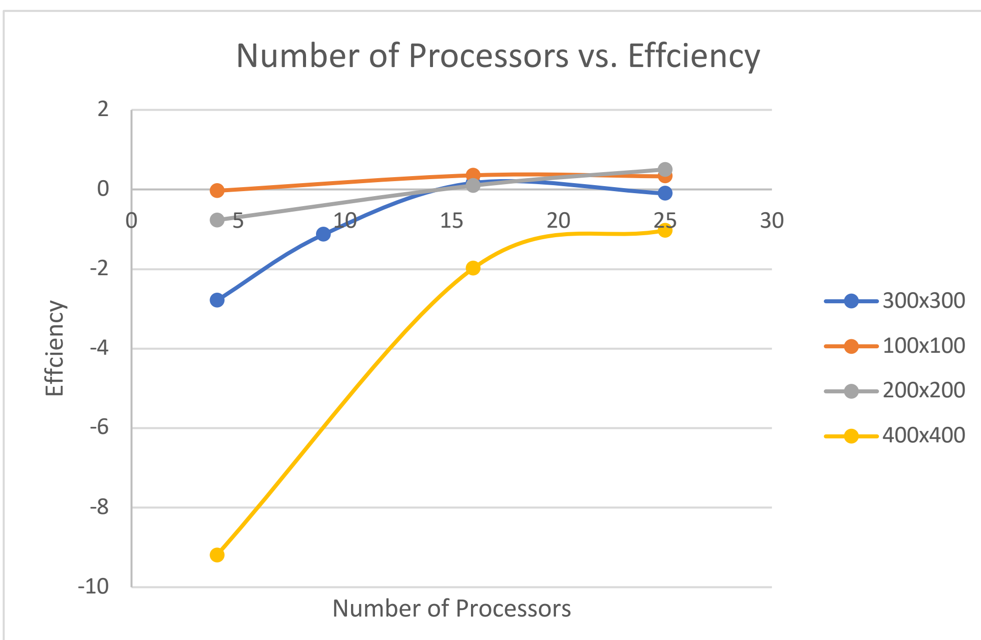

Figure 5: Efficiency Graph for 100-400 matrices

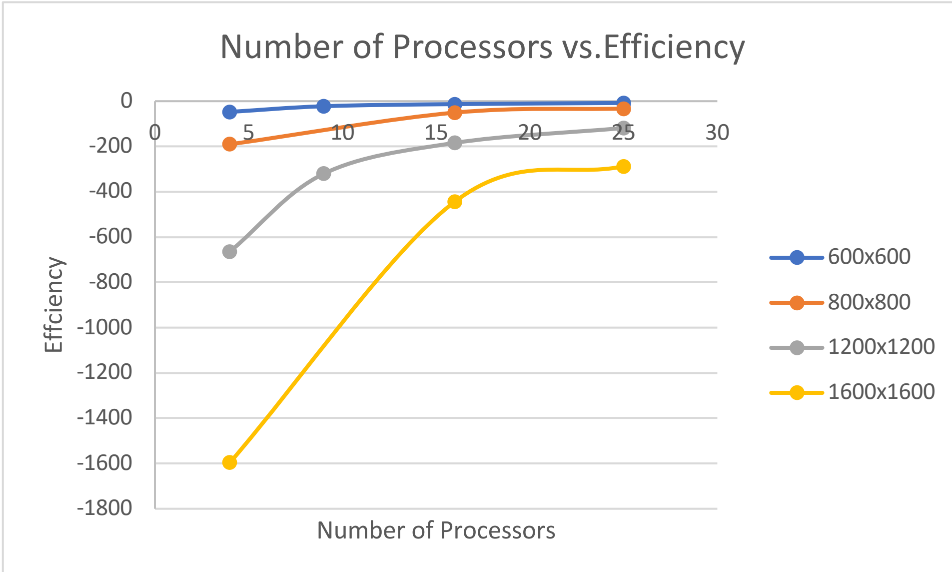

Figure 6: Efficiency for 600, 800, 1200, and 1600 matrices

Conclusion:
The purpose of this lab is to create a serial and parallel version of Cannon's O(n^3) matrix multiply algorithm. The apparent scalability and speedup for the different sizes of matrices is to be expected that is for a smaller sized matrix, it would speedup less than that of a larger sized matrix. Also, the efficiency of the matrices looks to be about what to expect as well. As the matrices get larger, the efficiency goes down exponentially. It has the same concept as that of speedup.