Ian Myers
Assignment 4 Tables

Execution Time

	Serial	4 processes (4x1)	8 processes (4x2)	16 processes (4x4)
N = 512	1.160868	0.550249	0.362250	0.357801
N = 1024	10.418317	3.702494	2.406324	1.967154
N = 2048	236.921295	80.492209	43.445049	24.907083

Speedup

	Serial	4 processes (4x1)	8 processes (4x2)	16 processes (4x4)
N = 512	1	2.10971397	3.20460455	3.24445152
N = 1024	1	2.81386465	4.32955703	5.29613696
N = 2048	1	2.94340655	5.45335546	9.51220562

Efficiency

	Serial	4 processes (4x1)	8 processes (4x2)	16 processes (4x4)
N = 512	1	0.52742849	0.40057557	0.20277822
N = 1024	1	0.70346616	0.54119463	0.33100856
N = 2048	1	0.73585164	0.68166943	0.59451285

Analysis

The execution time increased for every amount of processes as N increased. The execution time also was consistently lower for the more processes there were to complete the task. This is to be expected as the higher the N the more computation time is required as the matrices are larger and as the amount of processes increases, each process has to do a fewer amount of rows to complete the job in a quicker time. The speed up table reflects this as the speedup increases as more processes are added and they linearly improve as the N increases. The best speedup times were for N = 2048, which took a lot of time for the serial program and significantly less time for the multi process programs. This led to an efficiency in that row of up to .74, since the speedup for 4 processes at N=2048 was 2.94 and there were only 4 processes. This makes sense, since the more computationally exhaustive the N is, the better the higher process times will be compared to the serial time. This also results in the best efficiencies being at the bottom since that's where the best speedup times are.