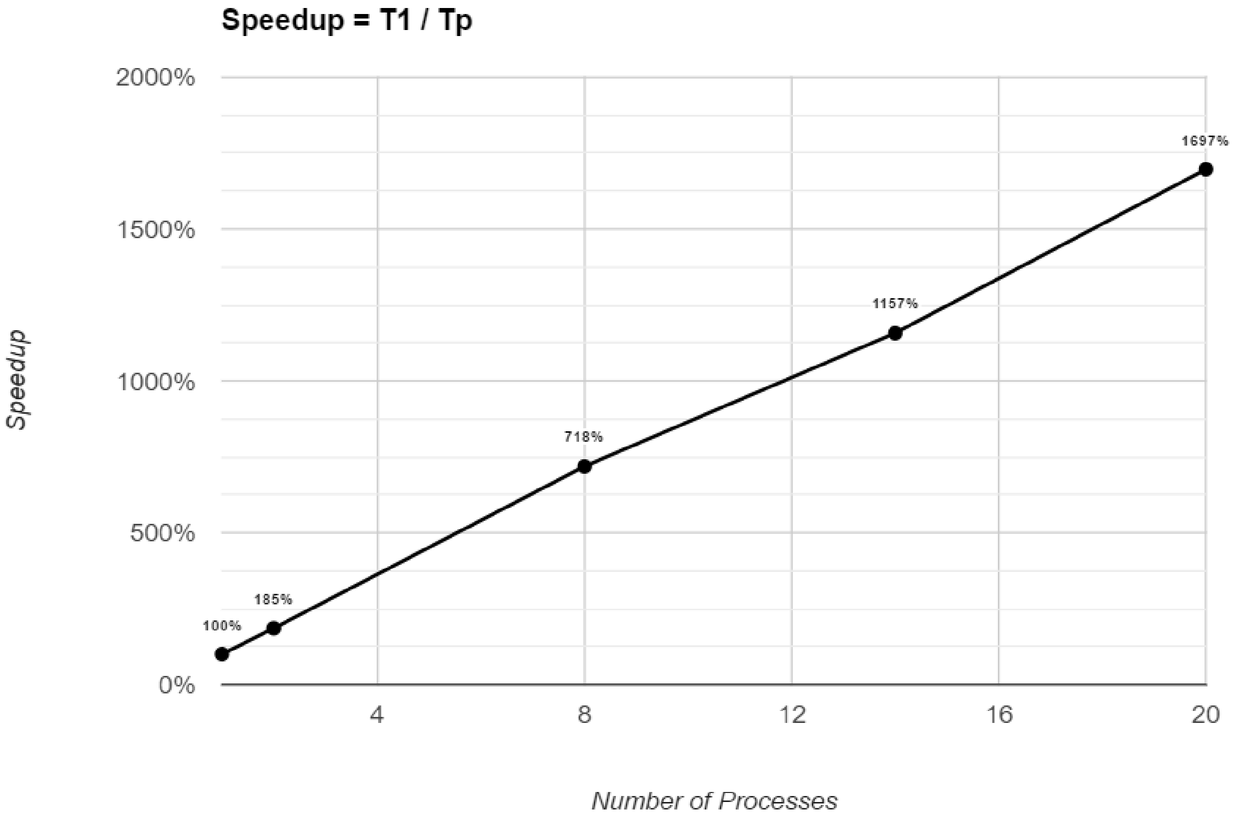
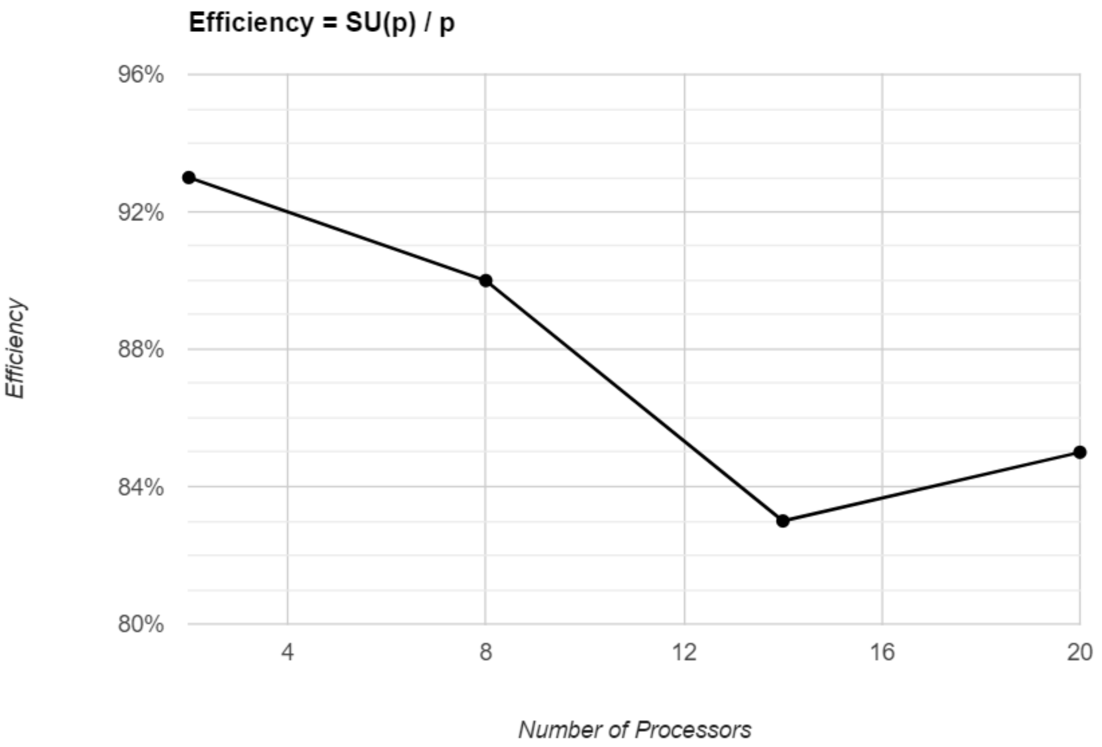
Value of n used: 10,500,411

Absolute relative true error for n trapezoids: 4.99313522824601016660e-15

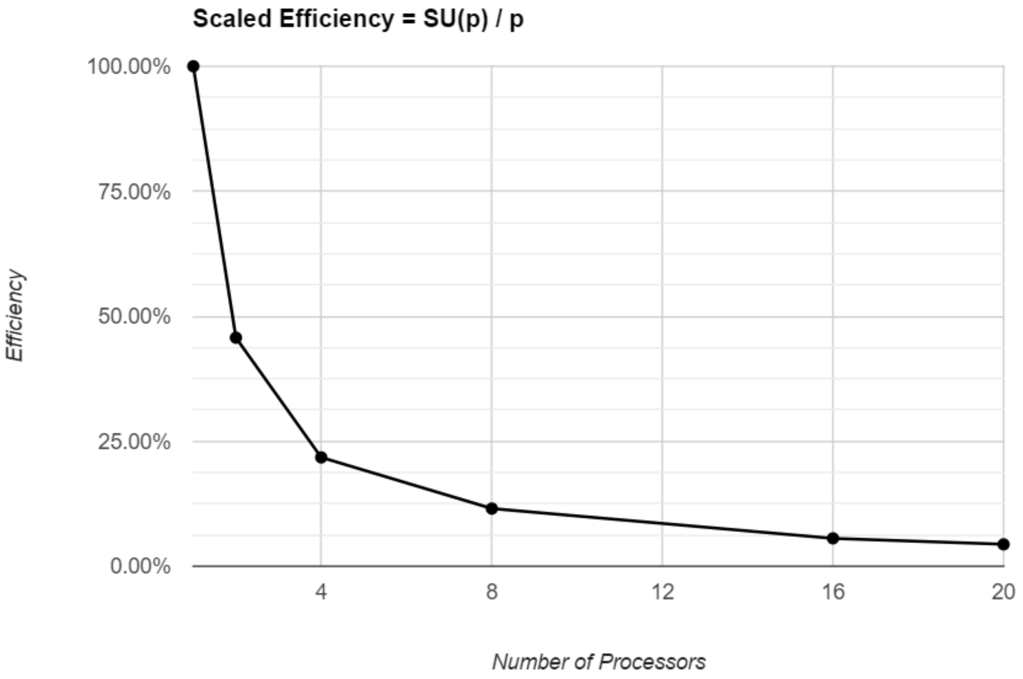
Number of Processors	Size of n
	10,500,411
1	5.738964e+00
2	3.098850e+00
8	7.992330e-01
14	4.958529e-01
20	3.382280e-01

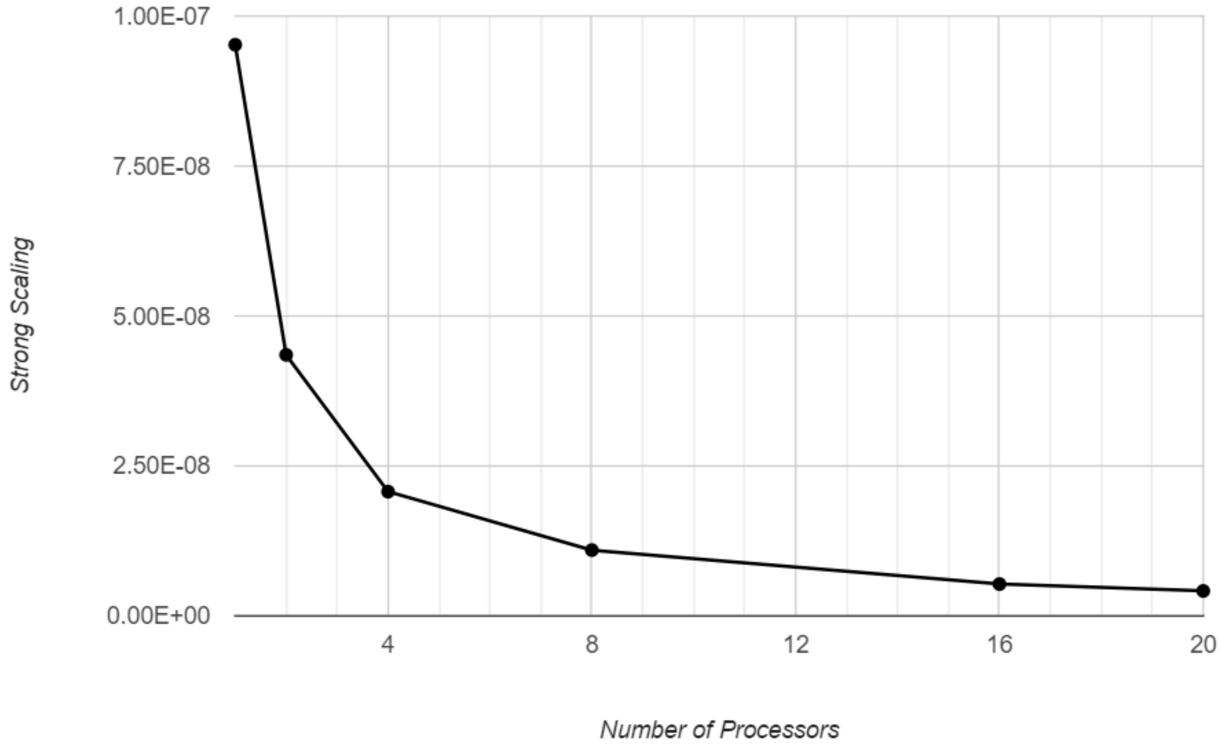
Number of Processors	Run Time	Estimated Integral	Absolute Relative Error	Speedup	Efficiency
1	5.738964e+00	4.0037209001513e+03	4.99313522824601016660e-15	100%	100%
2	3.098850e+00	4.0037207437570e+03	3.90622506863231722400e-08	185%	93%
8	7.992330e-01	4.0037204309777e+03	1.17184387934999383410e-07	718%	90%
14	4.958529e-01	4.0037201182112e+03	1.95303359731934977550e-07	1157%	83%
20	3.382280e-01	4.0037191799875e+03	4.29641282891690502370e-07	1697%	85%



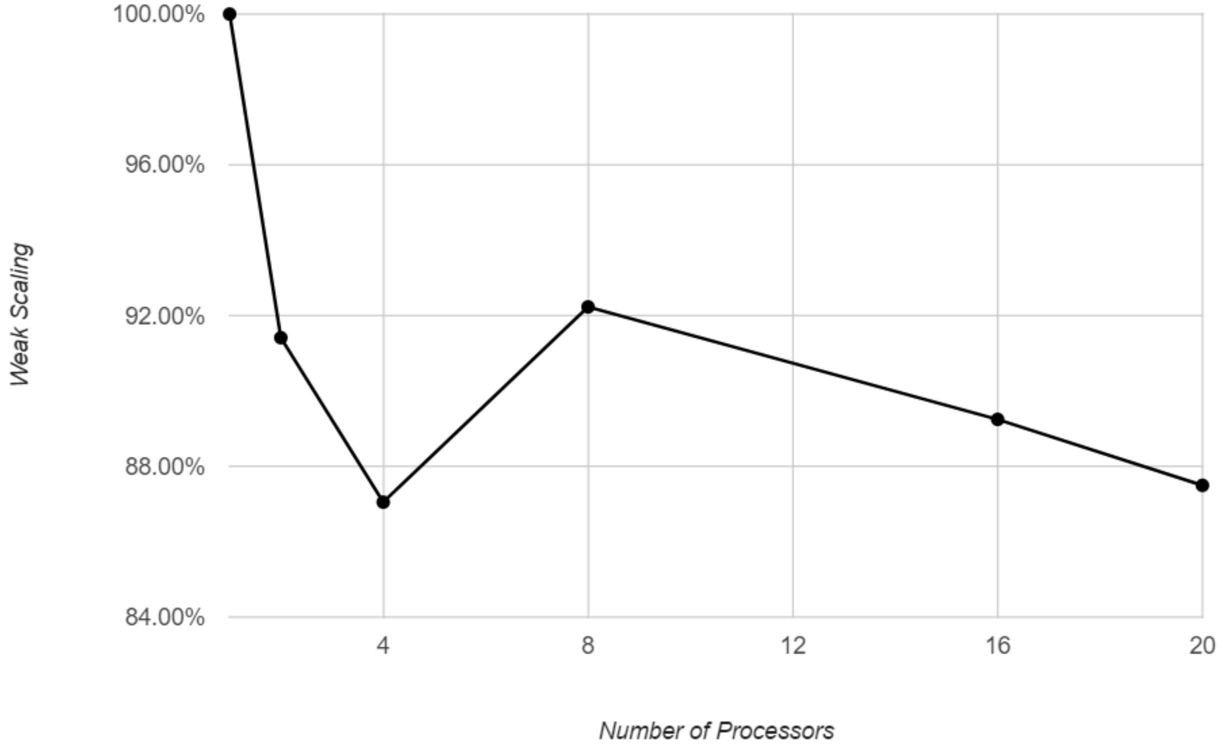

Efficiency = SU(p) / p

Number of Processors	Run Time	Size of n	Absolute Relative Error
1	5.719051e+00	10,500,411	4.9931352282460101666e-15
2	6.256325e+00	21,000,822	1.2734624479183500388e-15
4	6.569931e+00	42,001,644	2.8062538047499571244e-16
8	6.200814e+00	84,003,288	3.9154450319238528829e-17
16	6.408228e+00	168,006,576	2.1185552439021414139e-17
20	6.536430e+00	210,008,220	4.8804413995651427178e-17


Scaled Efficiency = SU(p) / p



Strong Scaling



Weak Scaling = (t1 / tN) * 100%

Conclusion & results:

Throughout this assignment, we took our previous lab where we found the minimum number of trapezoids needed to get below a certain error. We transformed it in that we used MPI to calculate the integral over multiple different processes. We also gave it n instead of having it find n. During this assignment, I learned a lot about MPI and how it worked as well as giving me some good hands on experience with MPI. I was able to conclude that this program got good speed up and that it had very good efficiency when ran on multiple cores.