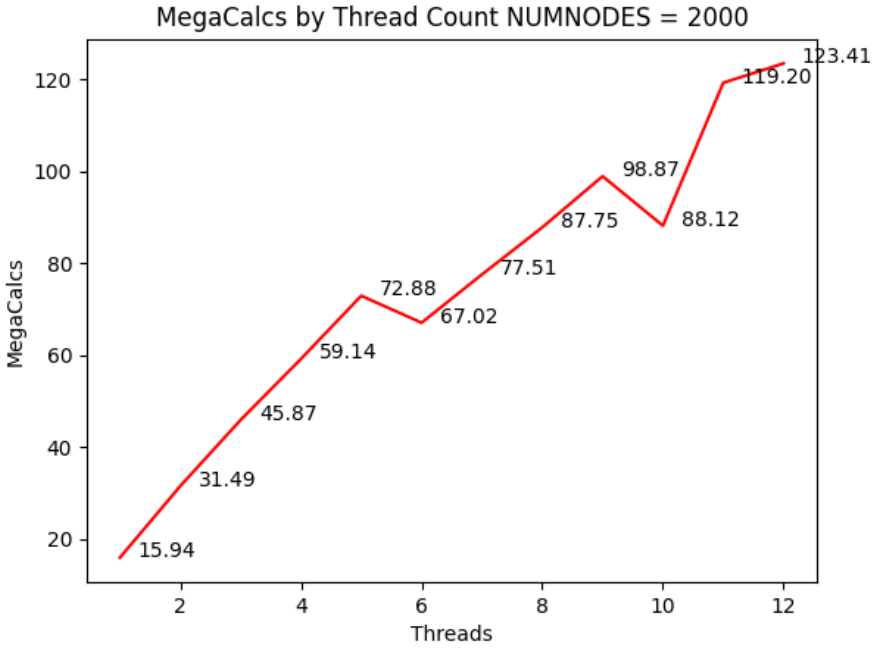
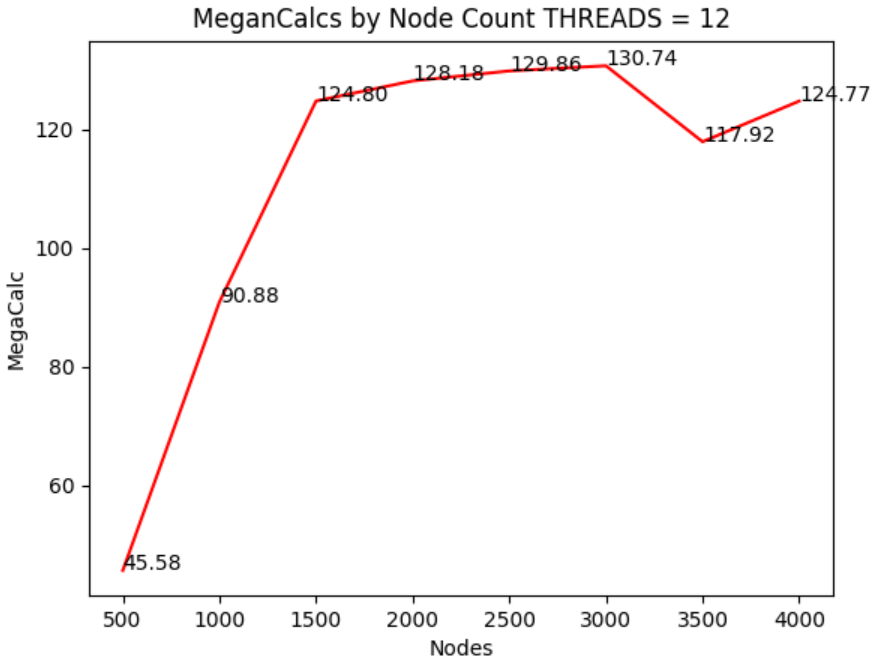
Numeric Integration w/ OpenMP Reduction

By Logan Saso

1. Machine Details
    a. Intel I7 8086k
        i. 14 nm process
        ii. 4ghz base (5ghz single-core boost speed)
        iii. 6 cores, 12 threads
    b. 32 GB Corsair Dominator Platinum RAM
        i. Clocked at 2600 MHz
2. I think the actual volume is around 6.48 units^3. It never really specifies what the unit is, be it inches, meters, or whatnot. But the output of the program is 6.48
3. Tables and Graphs

Nodes	Threads	Area	MegaCalcs
500	12	6.480511	45.578371
1000	12	6.481231	90.876284
1500	12	6.481898	124.799111
2000	12	6.481750	128.180919
2500	12	6.477538	129.864906
3000	12	6.491700	135.739703
3500	12	6.490125	117.920008
4000	12	6.473911	124.772917
2000	12	6.481750	123.407253
2000	11	6.482069	119.200387
2000	10	6.481351	88.121184
2000	9	6.481717	98.865719
2000	8	6.483333	87.753830
2000	7	6.483784	77.505802
2000	6	6.488411	67.016114
2000	5	6.482276	72.876927
2000	4	6.490974	59.142802
2000	3	6.480195	45.867334
2000	2	6.460046	31.494325
2000	1	6.427982	15.941469





4. As far as what I'm seeing in the speeds, it looks like what I'd expect. When the nodes are too small, we spend too much time organizing and updating shared variables that we lose significant performance. We essentially degrade to the point where using 2 threads would be just as if not more effective. As far as calcs per thread count, with sufficient size, I'm also unsurprised that the performance would increase linearly with threads, with the expected dropoff of organizing that many threads.

5. If the nodes are too small, the CPU is spending processing time organizing what it's going to work on next or waiting for other reductions to occur before a thread finishes, so we lose time. We also lose some speed when we go up in cores as certain counts may not be as well optimized as others or spend too much time waiting. Typically, it's better when the thread count can nicely divide the problem set.
6. Calculating the Parallel Fraction
    a. S_12 = 135.739703 / 15.941469 = 8.5149
    b. F = (12 / 11) * (1 - (1 / 8.5149)) = 0.96279142764
7. Given the speedup we got, I doubt we could get a significant bonus. Maximum I'd say we could get a speedup of around 9 on my PC, but I'd have to make sure that nothing intensive was running aside from this program before doing so.