Drake Seifert

CS 475 - Project 1

Tell what machine you ran this on

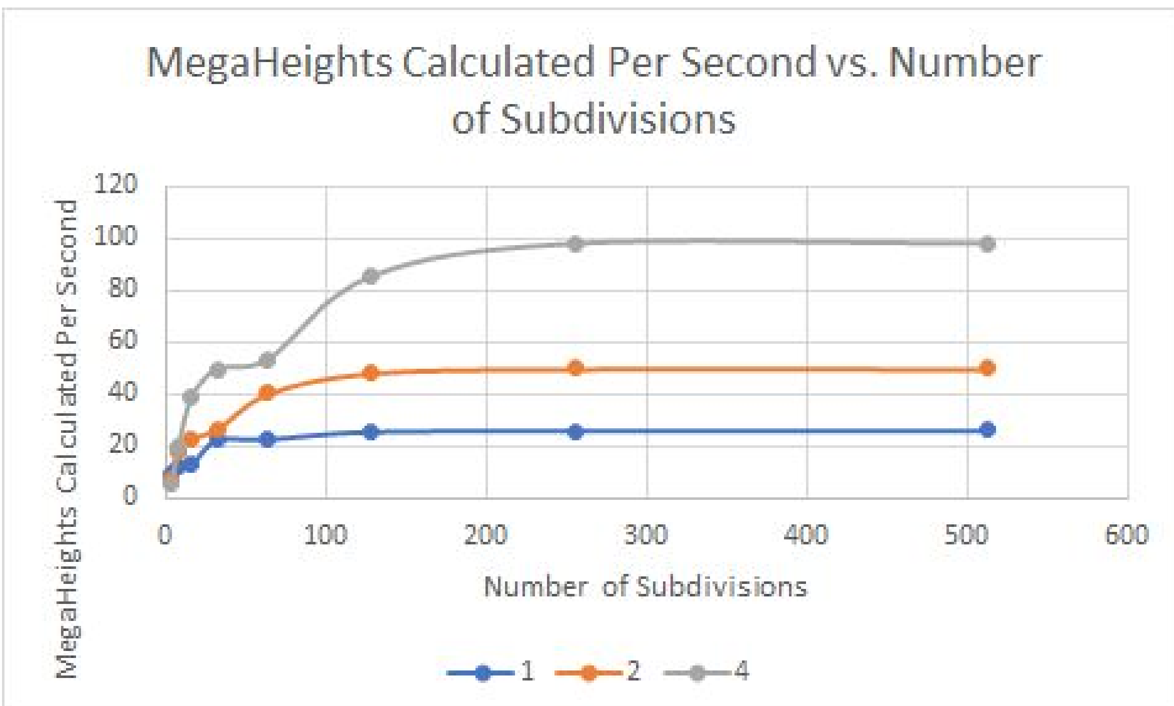
All tests were run on the OSU FLIP servers.

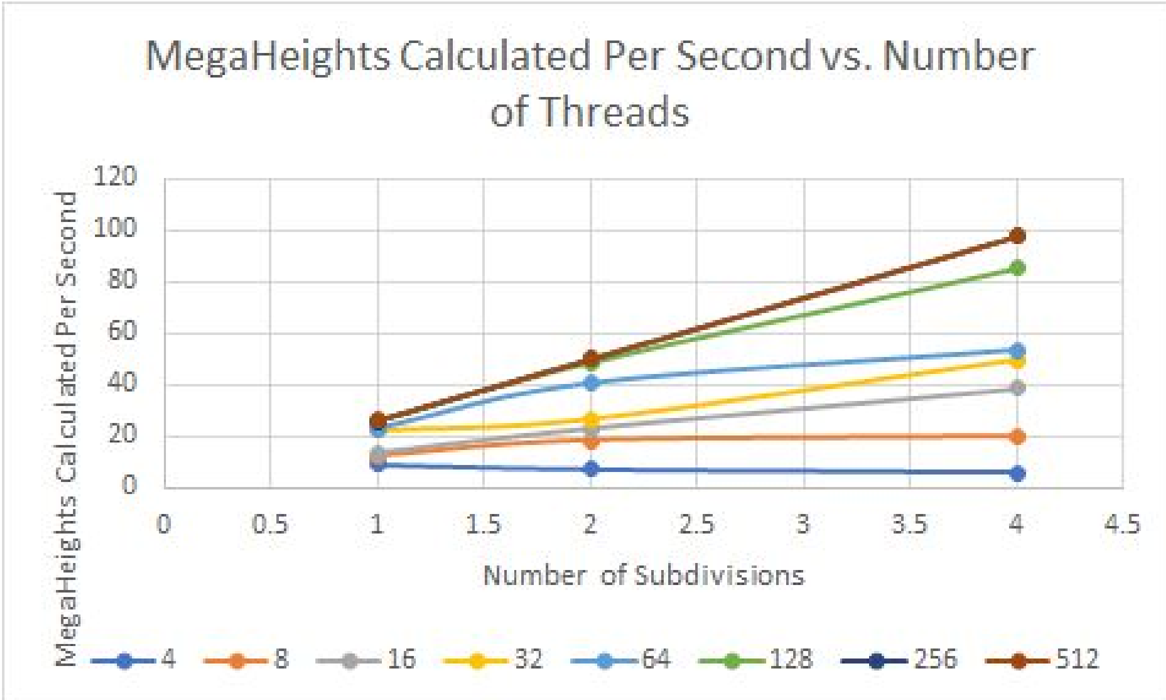
What do you think the actual volume is?

25.31 units.

Show the performances you achieved in tables and graphs as a function of NUMNODES and NUMT

Note: There were 100,000 trials for each run. Average performance from each was used in chart data if there were no large deviations in peak performance versus average performance.





What patterns are you seeing in the speeds?

The larger number of threads shows a much greater amount of performance given a fairly big subdivision. The performance seems to correspond with the number of threads; that is, performance gains are linear relative to the number of threads that have been spawned. However, the performance gains quickly taper off as each thread reaches maximum capacity. Also, a low number of subdivisions shows that a greater number of threads is slower than a smaller number.

Why do you think it is behaving this way?

Each thread has a maximum number of resources, and the chart shows at about what time each resource is used to its max. This can be seen when the MegaHeights Calculated Per Second begins to flatline. As expected, two threads can do about twice as much work as one thread, and four threads can do about twice as much work as two. This is because the work is distributed between each thread evenly, so as the number of threads increases, the overall workload per thread is reduced. There is a limit to this however, as seen in the below equations regarding maximum speedup and Amdahl's equation.

A low number of subdivisions results in poor performance for a large number of threads due to the overhead needed to spawn and manage multiple threads at one time. The greatest gains can be seen after about one hundred subdivisions in this specific scenario.

What is the Parallel Fraction for this application, using the Inverse Amdahl equation?

S = (Performance with four threads) / (Performance with one thread)
$Fp = (4 / 3) * (1 - (1 / S))$
$F_{Parallel} = 0.9792$

Given that Parallel Fraction, what is the maximum speed-up you could ever get?

$Speedup_n = 1 / ((F * (1 - n) / n) + 1)$
$Speedup_4 = 1 / ((0.9792 * -3 / 4) + 1) = 3.7649$

Max Speedup = $Speedup_n$ as $n \rightarrow$ infinity $= 1 / (1 - F_{parallel}) = 1 / (1 - 0.9792) = 48.0442$