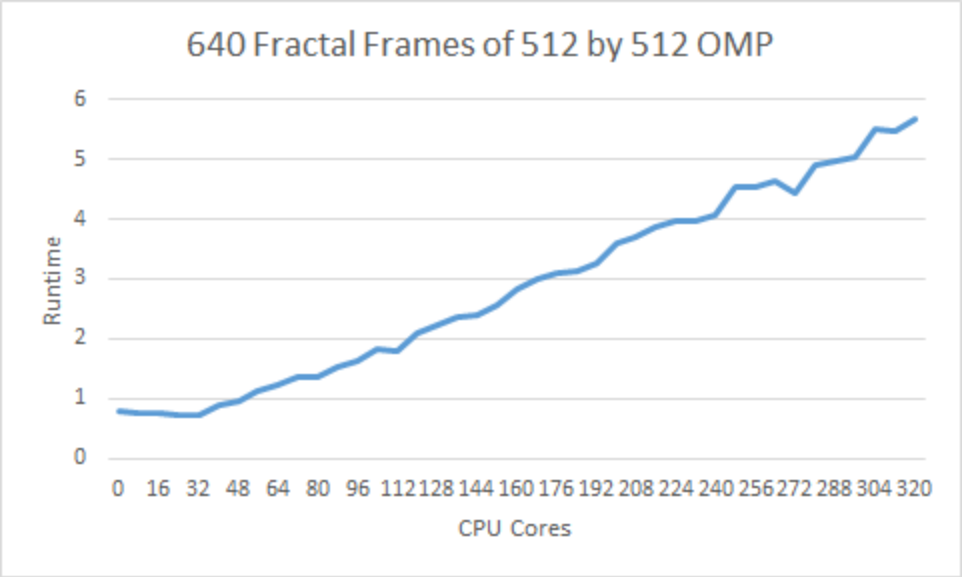
Project 6 Report
Blake Bordovsky / Joshua Galindo

Question 6.1a) Graphically present the runtimes in seconds in a single line chart (not a bar graph or a table) with the CPU frames along the x-axis and the runtime along the y-axis.

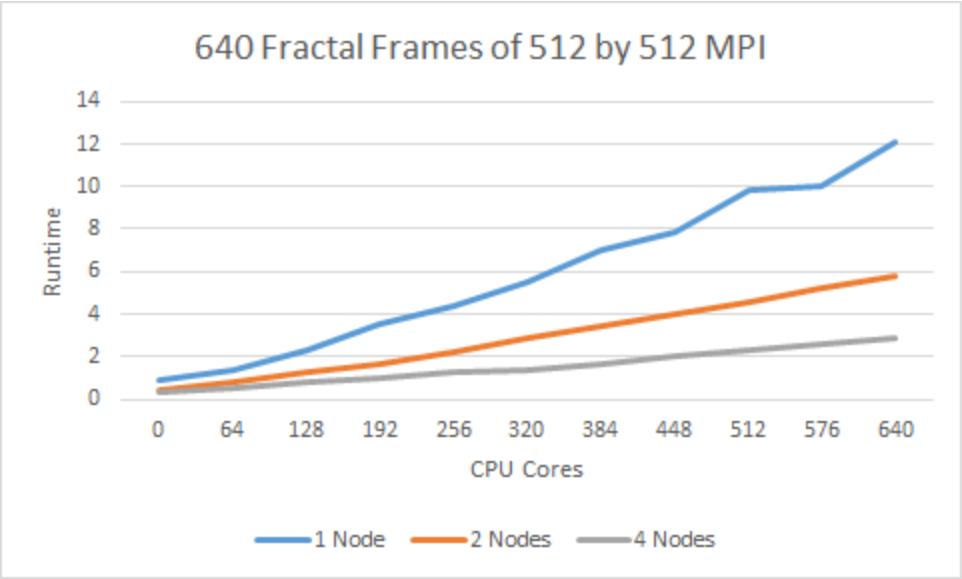


Question 6.1b) Explain the runtime behavior, especially the cases where the CPU or GPU gets all the frames and the general behavior of the runtime as a function of the CPU frames.
When the GPU gets more frames the performance is much higher because of the sheer throughput, but it needs enough data to process. When the CPU gets all the frames it loses performance because it's overloaded, but will outperform GPU on a small dataset.

Question 6.1c) Which CPU/GPU distribution of the frames results in the highest performance?
32 CPU Cores and 608 GPU Cores.

Question 6.1d) Using four digits after the decimal point, what is the runtime of just using the GPU and what is the best hybrid runtime (in seconds)? How much faster is the best hybrid execution relative to just using the GPU?
0.7249 seconds.
0.7714 seconds - 0.7249 seconds = 0.0465 seconds.

Question 6.2a) Graphically present the runtimes in seconds in a single line chart (not a bar graph or a table) with the CPU frames along the x-axis and the runtime along the y-axis. Use different lines styles or colors for the three different node counts.



Question 6.2b) Discuss the scaling and average speedup across node counts (in words).

Scaling is better when there are fewer CPU cores and more MPI nodes.

Question 6.2c) Comment on the usefulness of combining the three parallelization schemes (MPI, OpenMP, and CUDA). Which one(s) boost performance the most?

By utilizing all three schemes, the scalability of parallel computing is greatly increased. Each MPI node can have OMP CPU and CUDA GPU parallel code.

MPI and CUDA increase the performance.