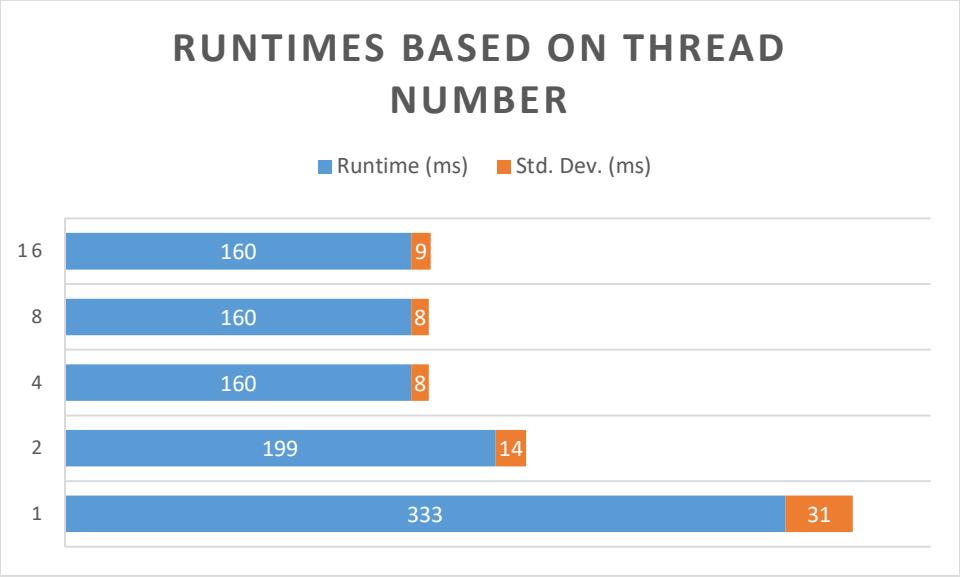
Daniel Oliveros, A02093272

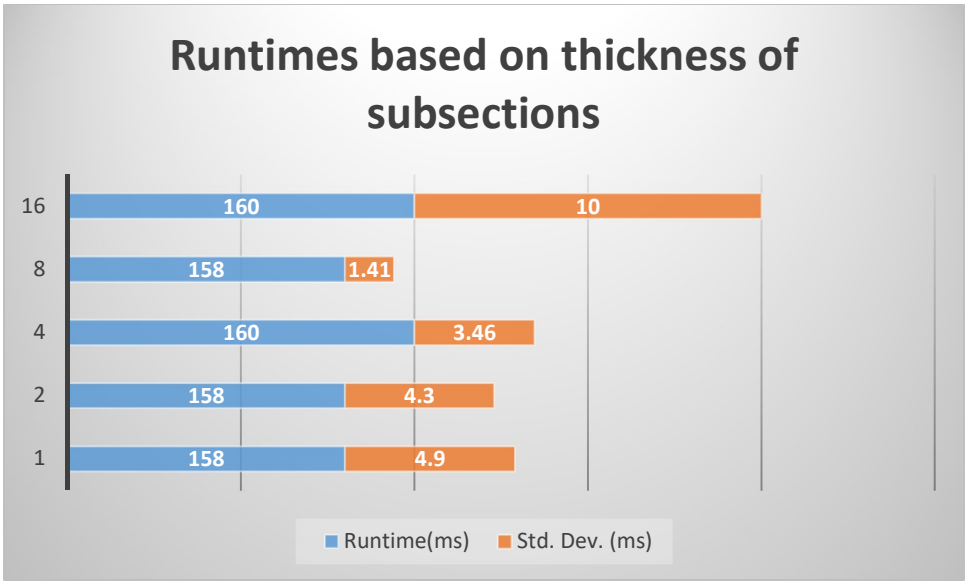
CS 3100

Threadpool Mandelbrot

Firstly, we will evaluate the changes in runtime and standard deviation as we go up in threads. In this setup, the datasets that were being worked on had a thickness of 4 rows. As expected, a higher number of threads lead to a shorter runtime. Interestingly enough, after reaching a higher number of threads I ended up reaching a time I could not surpass, and my standard deviation increased slightly.

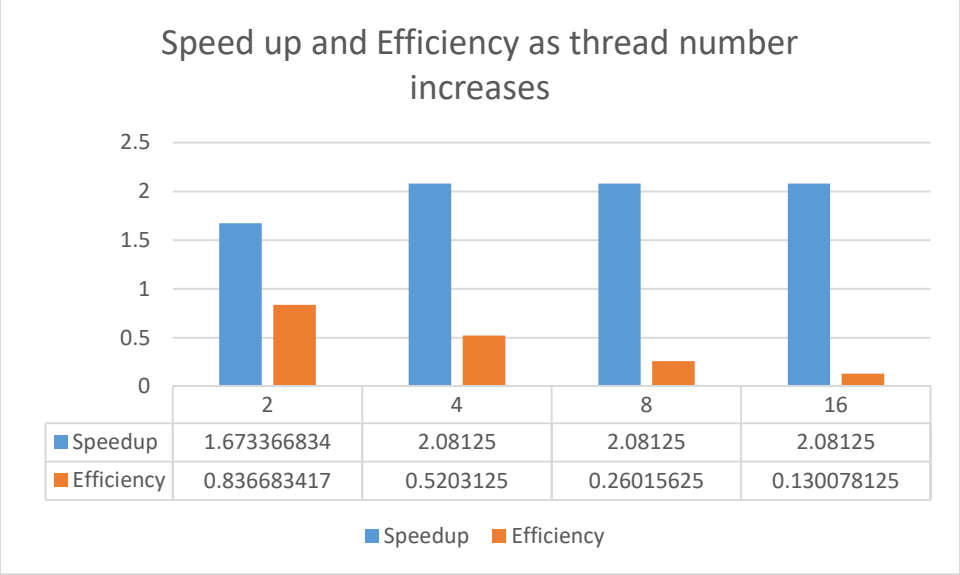


On my second analysis, I left the thread number at 4 and only varied the size of each dataset that was added onto the queue. The entire image remained just as big.



The standard deviation varied wildly in these trials, during multiple different attempts it seemed to just go everywhere. The runtime, however, remained around the same.

In these experiments, there was noticeable speedup when moving up in threads at lower numbers, but it was clear after a certain point that we'd hit our wall time. In these trials our wall time was 160 milliseconds. We will now refer to the following graph to display these things in better detail.



As our thread number increased, our speedup compared to the runtime with 1 thread did not increase. This could signify that we'd reached our wall time. This also meant that our efficiency went down considerably as well. Because our runtime varied at a negligible rate, adding more threads soon became much less worthwhile.