

Laboratory Assessment 2: Parallel Computing

Student 1076589

August 14, 2017

1 Vector dot product:

1.1) Parallelization is useful in the problem of the dot product as it contains independent operations ($a_i * b_i$) that are repeated over the vector. In this case, we parallelize the multiplication of the value pairs in the vectors, each thread adding independently, and summing the partial sums.

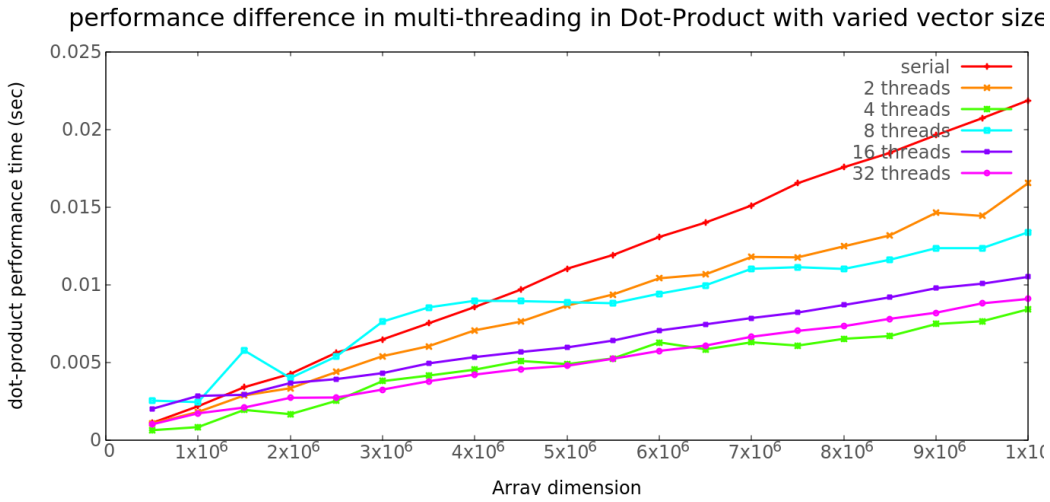


Figure 1: Line plot of array dimension vs operation time, comparing serial and threaded dot-product

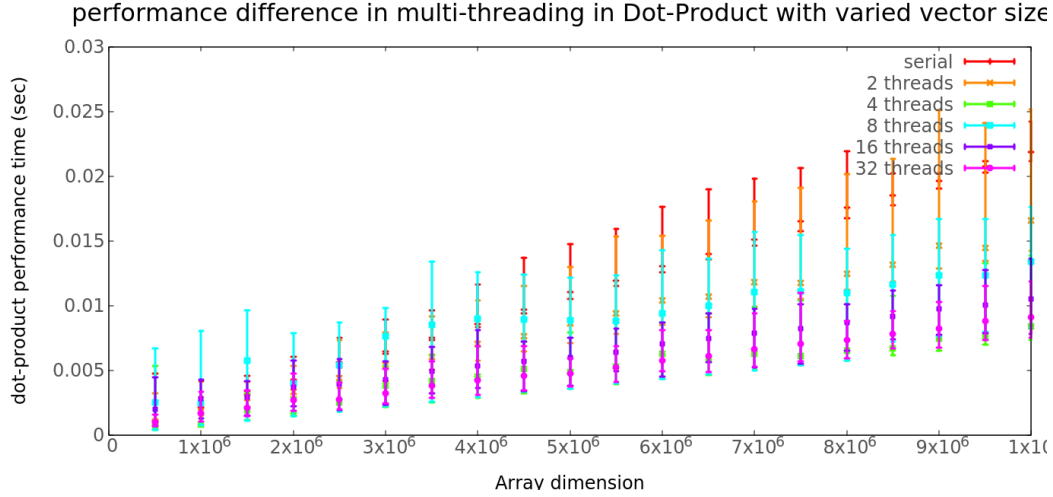


Figure 2: scatter plot with error bars, using min, max and average

1.2) For small vector sizes, increasing the number of threads has unpredictable and erratic affects, as can be seen in figure 1, showing no improvement in the 1st sector of the graphs.

For larger vector sizes, the observed performance stabilizes with the threaded programs generally out-performing the serial one. The 4 threaded program performs the best while the 2 threaded program performs the worst after the serial version.

This observation makes some sense since it can be explained, in part, by the overheads incurred when attempting to parallelize code. One would expect parallelization to improve performance immediately, but would only be able to overcome the overheads and show a significant improvement with larger data-sets. 4 threaded program performs best because it seems to optimally take advantage of resources (cores and such) while minimising overheads.

1.3) For smaller array sizes, no performance improvement was observed. For larger array sizes, a more positive stable linear relationship was observed, with 4 the threaded program having the lowest gradient