Algorithm Description

The algorithm found in syr2k is used to compute the sum of two products. The two products are computed from elements from two matrices. This is done using a triply nested for loop. Two of the for loops bounds are determined by user input, N. This leads to a complexity of O(n^2). These for loops give us the opportunity to test locality and its effect on speedup.

Parallelization Approach

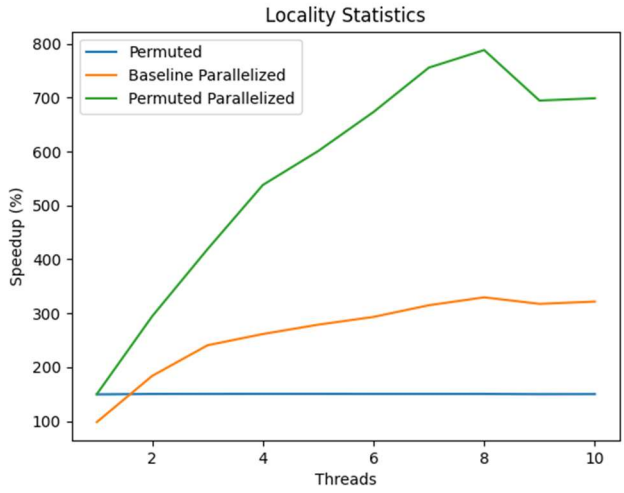
In addition to testing locality, we will be parallelizing the outermost for loop. This allows us to reduce complexity to O(n^2/p).

Experimental Setup

I used the Cheyenne state capital machine for this experiment. This machine has 8 cores. The L1d and L1i cache are 256 KiB, the L2 cache is 2 MiB, and the L3 cache is 20 MiB. I am the only user on this machine

Experimental Results

The experiment was ran by taking the average of eight tests, where the min and max of these results were removed. This data was then averaged. This process was repeated, each time with an increasing number of threads up to and including 10 threads. The plot of average execution time was then produced using pyplot.



Conclusion

From our results, we see that locality has a great impact on the execution time of this program. The permuted program has ~150% speedup. When we parallelize the program without permuting the for loops to optimize locality, we get more speedup. The best result is obtained by optimizing for loops for locality and parallelizing. This resulted in speedup of 787.9555% when using 8 threads! This is likely due to the hardware prefetching the data from the row we are computing from. This effect is compounded when using multiple threads.