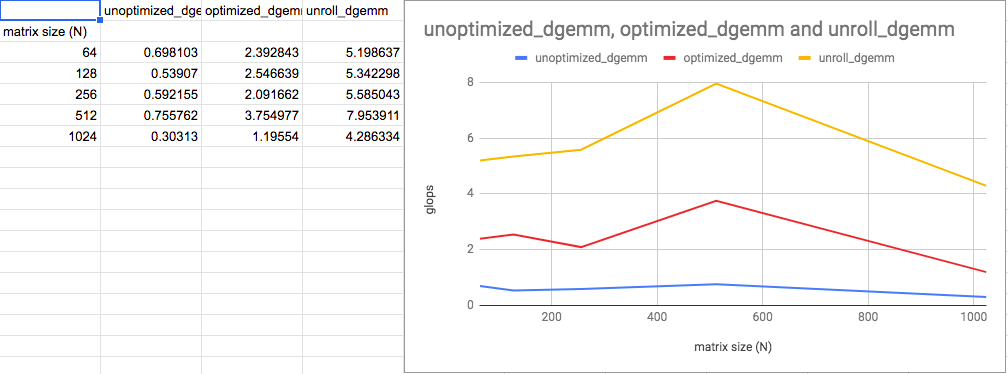
Nicholas Pierce

CSCI402

Lab4

Unoptimized\_dgemm vs Optimized\_dgemm vs Unroll\_Dgemm Analysis



From the data and graph above, we can see that the unroll dgemm is the fastest kernel of the 3 above. The unroll function performs roughly twice as fast as the optimized dgemm, and the optimized dgemm performs at roughly 4 times the speed of the unoptimized dgemm. Therefore, the unroll dgemm is approximately 8 times as fast as the original unoptimized dgemm. The unroll function overall is the most efficient and runs in the least amount of time. But why is this the case? Loop unrolling is a technique used to increase the performance of array accesses by increasing the parallelism of the instructions that are running. The for-loop overhead is removed and is replaced by the same array accesses written sequentially. The compiler will perform the actions in parallel, the key theme here is increasing instruction level parallelism of the array accesses. We can see objectively that the loop unrolling is a very sound technique for improving performance looking at the graphs of the functions. While the previous dgemm utilized intel intrinsics to perform a similar kind of parallel instructions, loop unrolling is a vastly superior method by a factor of 2 times.