

EECE5640 HW3

Quinn Arbolante

March 2025

Question 4

Part A

My code can be found as `blas_simple.c`. `blas_simple.c` takes 0.6764117ms to run (averaged over 10 runs). This is on a Intel(R) Xeon(R) Gold 5318Y CPU @ 2.10GHz. The minimum frequency is 800 MHz, the maximum frequency is 3400 GHz, the number of cores is 96 (2 threads per core, 24 cores per socket, and 2 sockets), and the memory size is 135.8MB (2.3MB for the L1d cache, 1.5MB for the L1i cache, 60MB for the L2 cache, and 72MB for the L3 cache). The operating system is Rocky Linux 9.3 (Blue Onyx). On the other hand, my implementation for Question 3 takes 588.819626ms (averaged over 10 runs). I believe the OpenBLAS implementation takes significantly less time, because it utilizes multiple cores, whereas my implementation only utilizes one core.

Part B

The results in Part A were on a CPU in Explorer. For this part, I will also use an Intel(R) Xeon(R) CPU E5-2680 v4 @ 2.40GHz. The minimum frequency is 1200 MHz, the maximum frequency is 3300 GHz, the number of cores is 28 (1 thread per core, 14 cores per socket, and 2 sockets), and the memory size is 78.79MB (896KB for the L1d cache, 896KB for the L1i cache, 7MB for the L2 cache, and 70MB for the L3 cache). The operating system is Rocky Linux 9.3 (Blue Onyx). This CPU took 0.835944ms (averaged over 10 runs). I believe this CPU took more time on average than the other CPU, because the cache is smaller, so there is less room for exploiting fast memory, and there are less cores, so there is less room for parallelization.