1. Assignment 1: CUDA Vector Addition - Ilan Sela

1.1 Experiment Results

Table 0: Execution times for part 3

Number of Threads	Execution Time (ms)
1	0.017
2	0.007
4	0.006
8	0.006
16	0.007
32	0.006
64	0.006
128	0.006
256	0.006
512	0.007
1024	0.007
2048	0.009
4096	0.010
8192	0.014
16384	0.020
32768	0.047
65536	0.078
131072	0.170
262144	0.274
524288	0.586
1048576	1.276
2097152	2.397
4194304	4.645
8388608	9.652
16777216	19.827
33554432	36.296
67108864	74.559

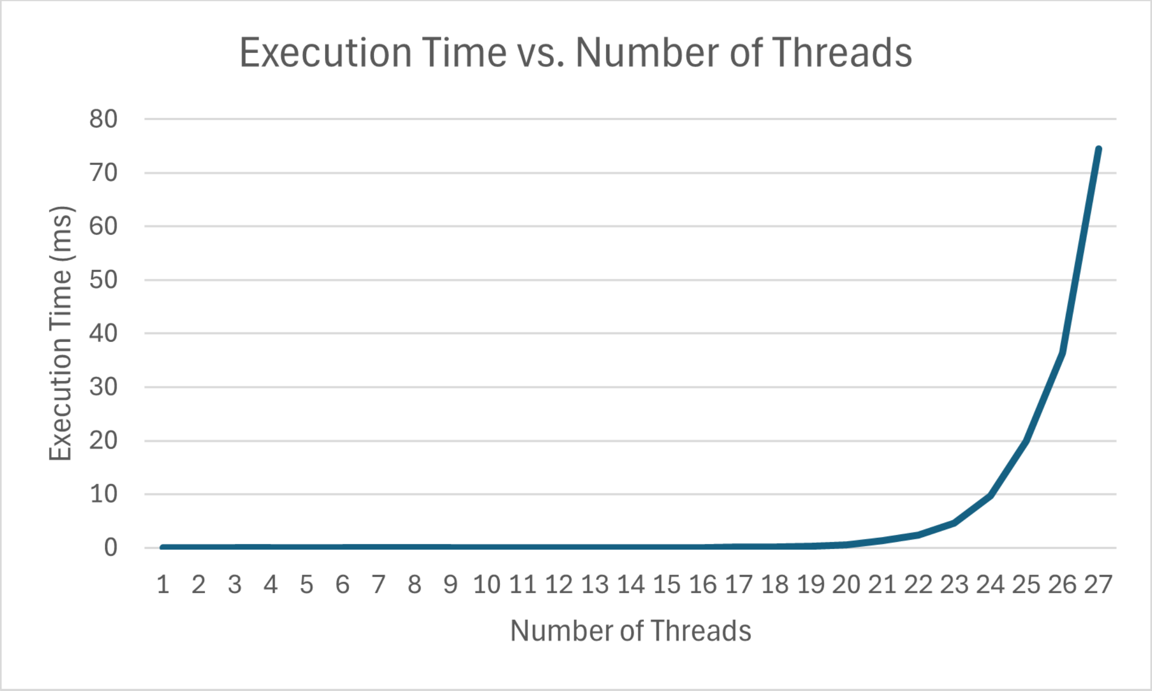

Figure 1: Execution times for part 3

1.2 Report

The CUDA program performs vector addition for two arrays. Vectors are allocated in host memory, assigned random values, and then copied to device memory. Using the vectors on the device, we launch a kernel with one thread per block and the number of blocks equal to the array size. Each thread performs an element addition and stores it in the resulting vector. The execution time for the kernel is measured via the CUDA runtime API. Once calculations are finished on the device, the resulting vector is copied to the host.

As we increase the number of threads, we observe an exponential increase in the execution time. Ideally, the execution time would be constant. However, one possible cause of the difference could be due to kernel launching overhead, as each block requires resources and we are launching blocks equal to the size of threads.