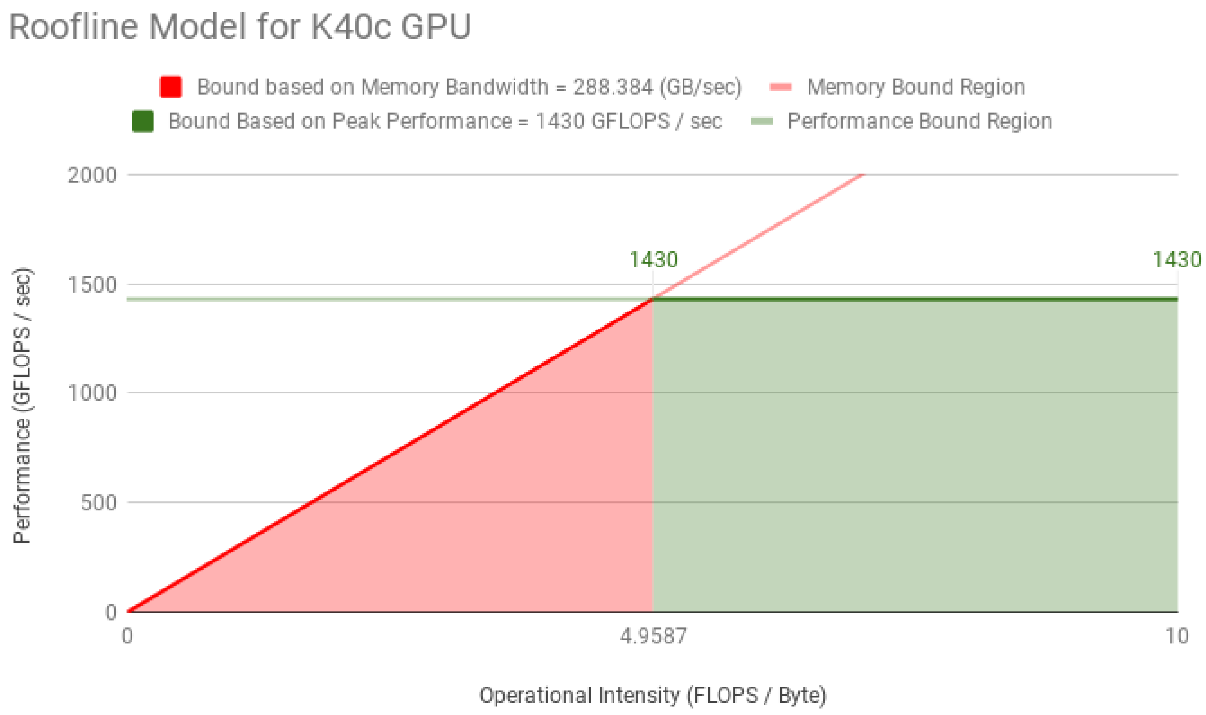
CS6023 Assignment 1 - Report

E Santhosh Kumar (CS16B107)

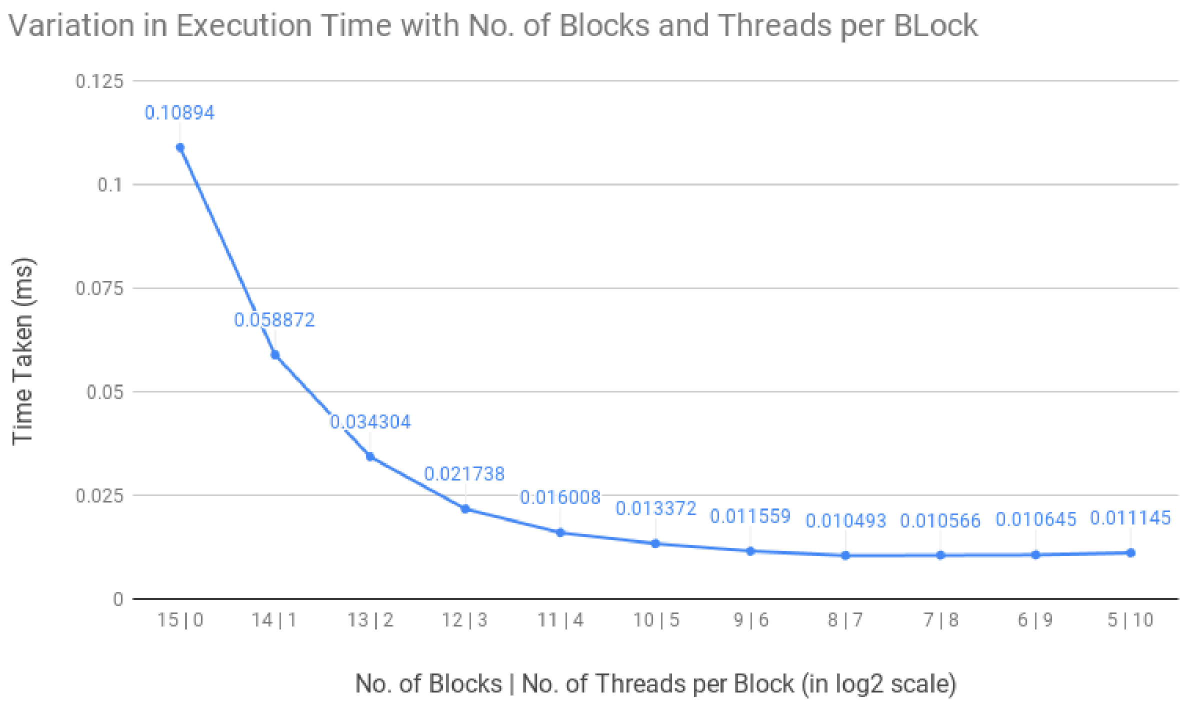
Q2



$$Memory\ Bandwidth = (Memory\ Bus\ Width) * (Memory\ Clock\ Rate) * 2 = 288.384\ GB/s$$

The factor of 2 is included as the GPU uses Double Data Ram.

Q4

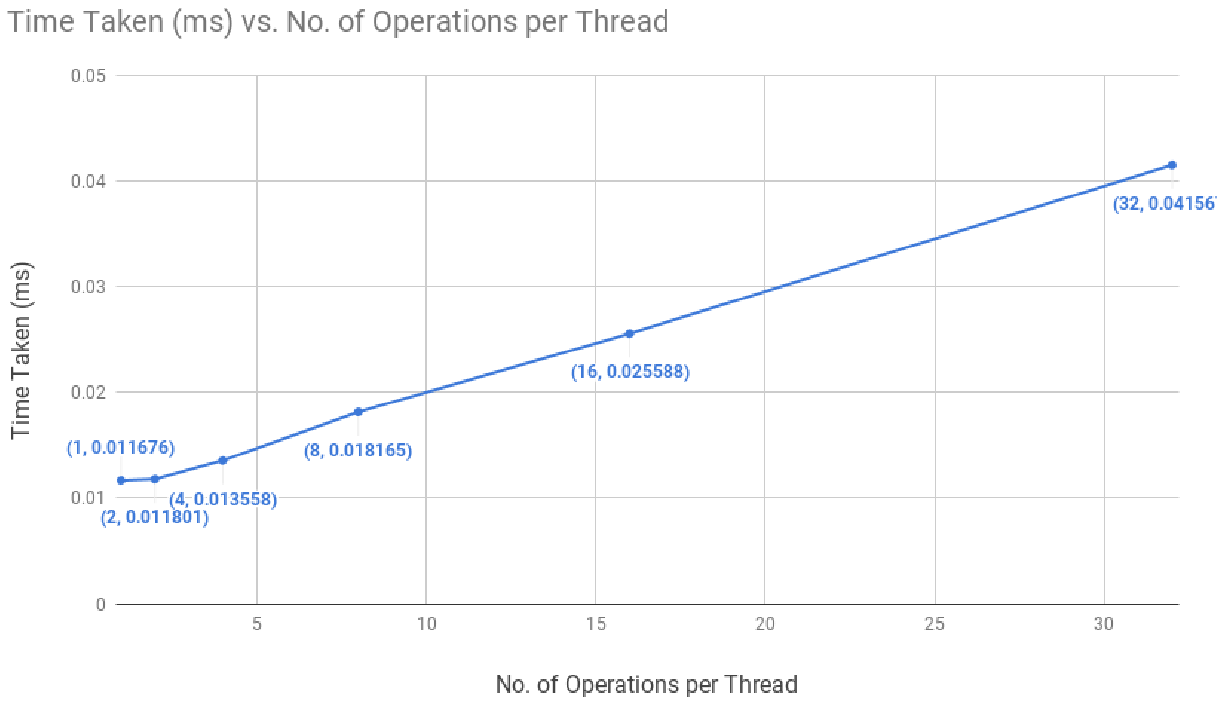


The optimal configuration is given by

$$No.\ of\ Blocks = 2^8 = 256$$

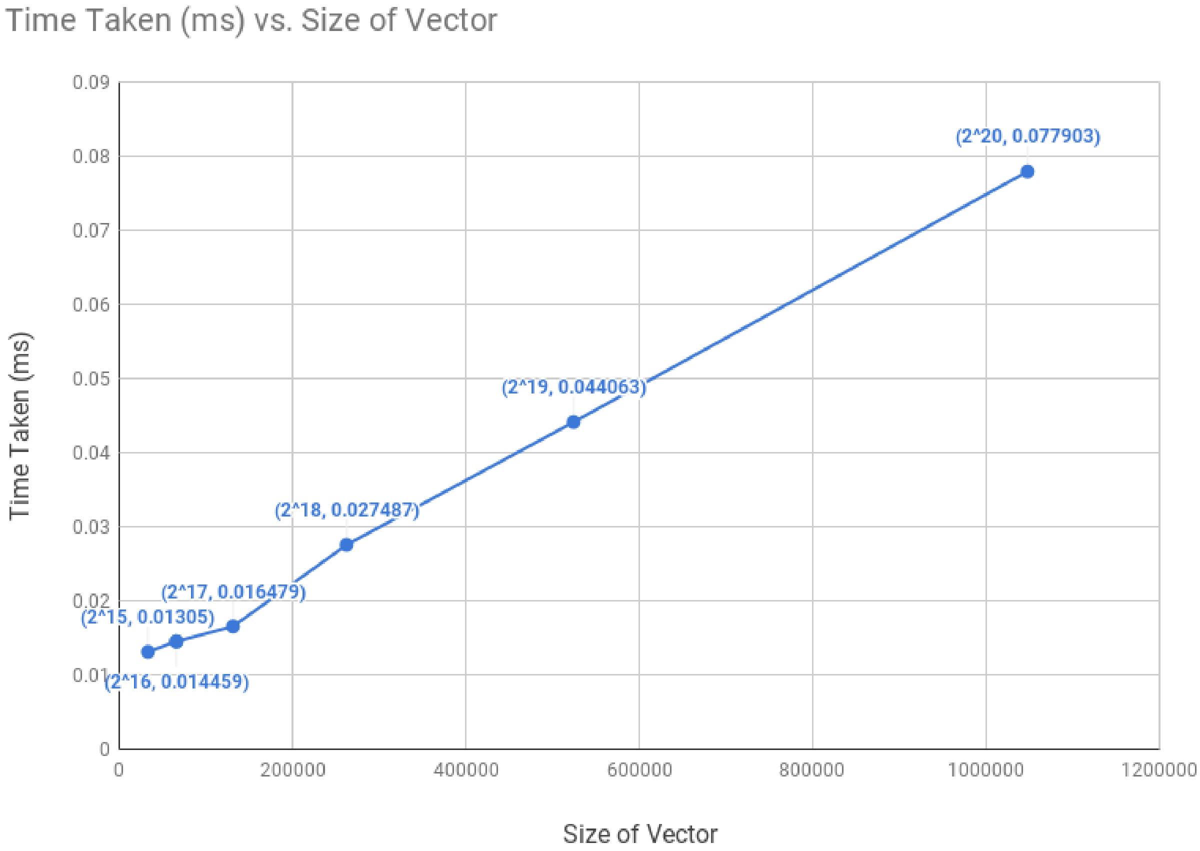
$$No.\ of\ Threads\ per\ Block = 2^7 = 128$$

Q5



The optimal configuration is when each thread performs exactly 1 operation.

Q6



Q7

Explanation for Q2

The theoretical peak memory bandwidth is given by the formula

$$\begin{aligned} Memory\ Bandwidth &= (Memory\ Bus\ Width) * (Memory\ Clock\ Rate) * 2 \\ &= (384\ bits) * (3004000\ kHz) * 2 \\ &= 288.384\ GB/s \end{aligned}$$

The factor of 2 is included as the Nvidia Tesla K40C GPU uses DDR (double data rate) RAM.
The lines denoting the memory bound and performance bound regions intersect at the point where

$$operational\ intensity = \frac{peak\ performance}{memory\ bandwidth} = \frac{1430\ GFLOPS/s}{288.384\ GB/s} \approx 4.9587\ FLOPS/Byte$$

Explanation for Q4

The warp size is 32. Hence, having less than 32 threads per block means that whenever the threads in such blocks get executed, they do not use all of the 32 SIMD ALUs present in the SM. There is under-utilization and hence performance improves by a lot as the number of threads per block increases in this region. Once threads per block is over 32, the time taken does not differ by a lot.

Explanation for Q5

we use No. of Blocks = 256 (optimal configuration from Q4). Hence, when we have more than 4 operations in every thread, we have less than 32 threads per block, which leads to relatively poorer performance (as explained in Q4). The best value of time taken is obtained when we have just 1 operation per thread. This is because in this case we have fewer loop operations and also make better use of the optimized GPU schedulers (the optimal configuration has the maximum number of threads that need to be scheduled).

Explanation for Q6

The size of the vectors are much greater than the number of SIMD units that work on the addition operation at any given cycle. Hence, each kernel call requires several clock-cycles to complete and the number required increases linearly with the size of the vector. Thus we get a linear graph.