Assignment 1 : CS6023 GPU Programming

by CS15B049

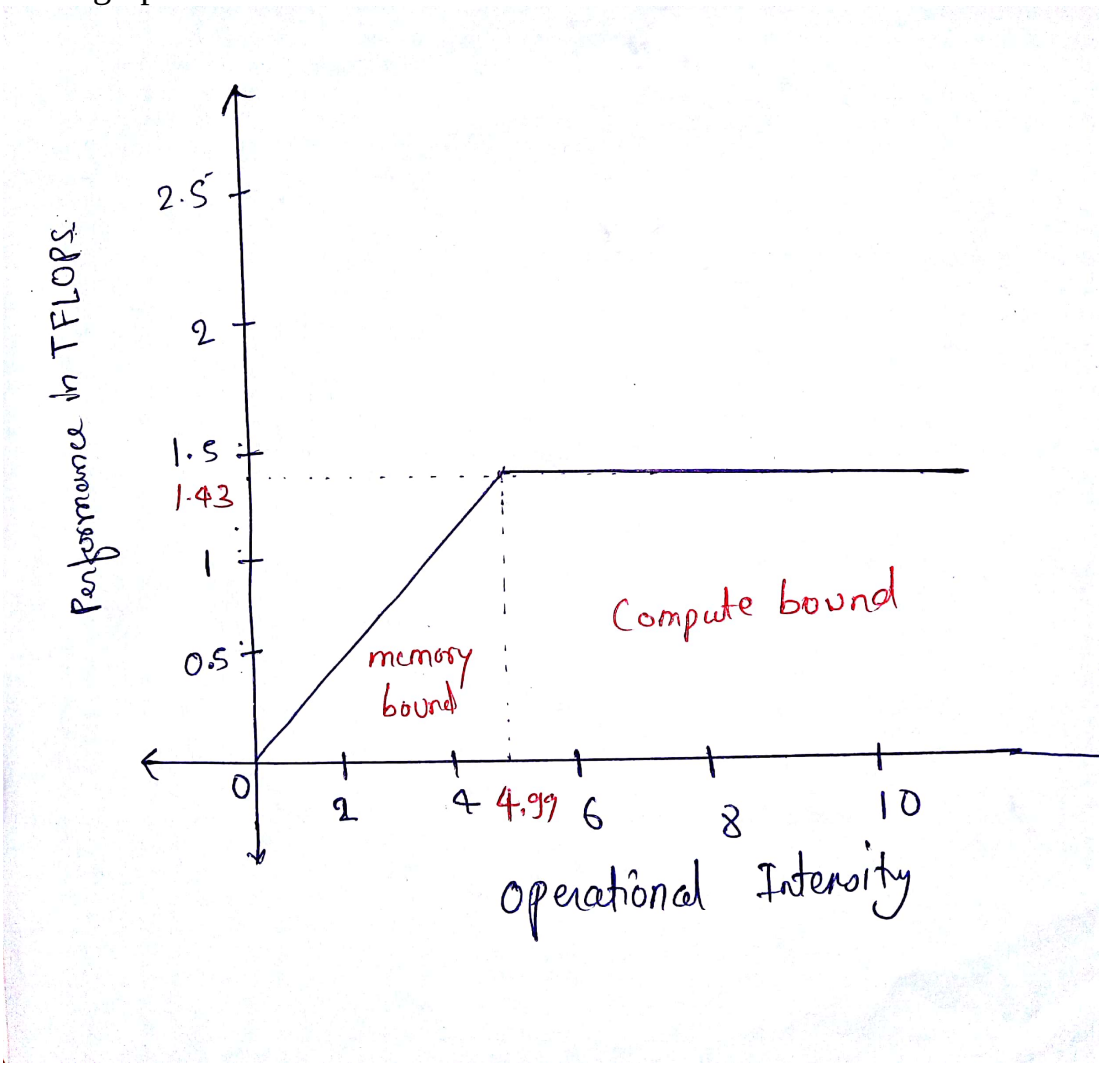
Q1 : The value of the following device properties are :

L1 cache supported locally : TRUE
L1 cache supported globally : TRUE
L2 cache size (in bytes) : 1572864
maximum allowed number of threads per block : 1024
registers are allocated per block : 65536
registers are available in a Streaming Multiprocessor : 65536
warp size : 32
total amount of memory available in the GPU (in bytes) : 11995578368

Q2: The roof line model graph is :

calculation of peak performance = (number of SM**wrap_size)*2*2*clock_rate
= (15*32)*2*2*745000000 = 1.43 TFLOPS
memory bandwidth = 288 GB/sec

The graph :



Q3: The code is submitted along with this report.

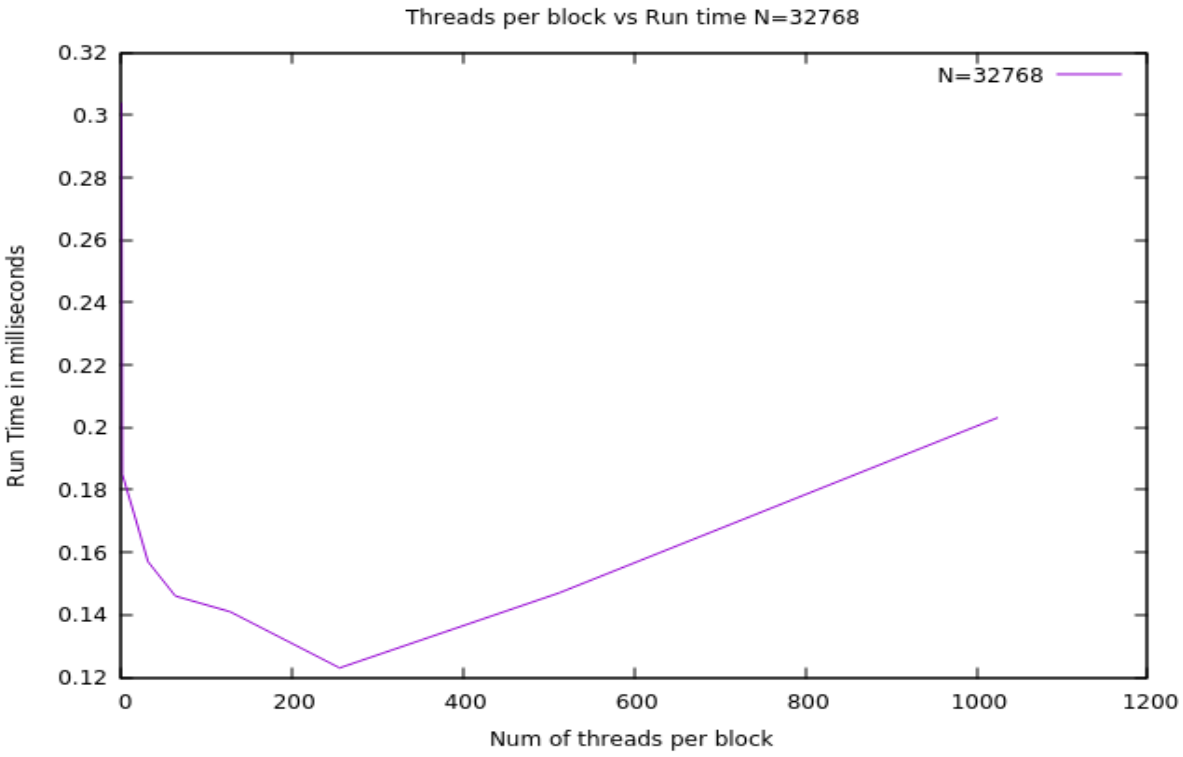
Q4: The runtime configuration :

size of vector (N) = 32768
num of ops per thread = 1

Optimal Configuration for N=32768 :

Threads per block = 256
Blocks per grid = 128

Graph :



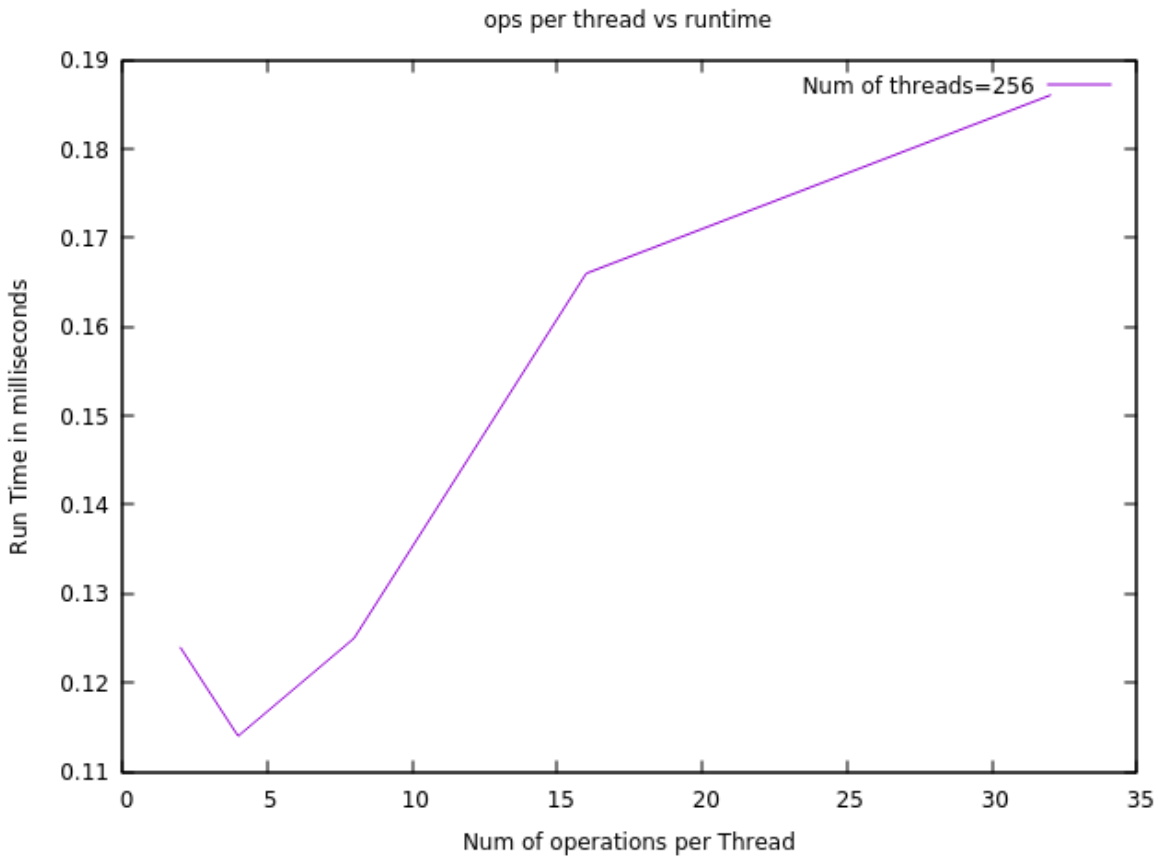
Observation : The run time decreases as num of threads per block increases and till it saturates the hardware and after it increasing number of threads increase runtime.

Q5: The runtime configuration :
size of vector (N) = 32768

Using optimal Configuration from Q4:

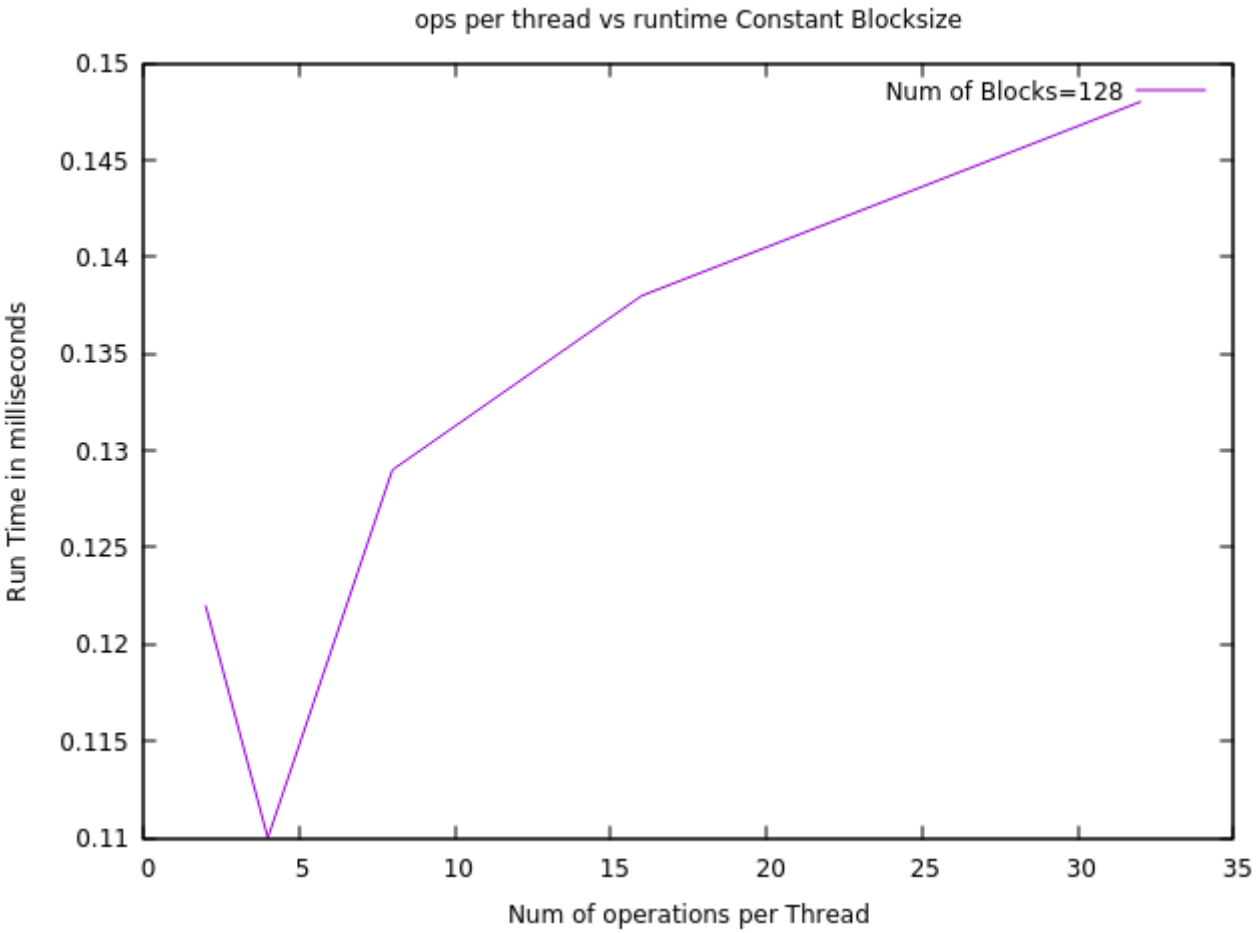
(i) Num of threads per block = 256 and varying block per grid wrt num of ops per thread as N remains constant.

Which gives optimal runtime at number of ops per thread = 4;

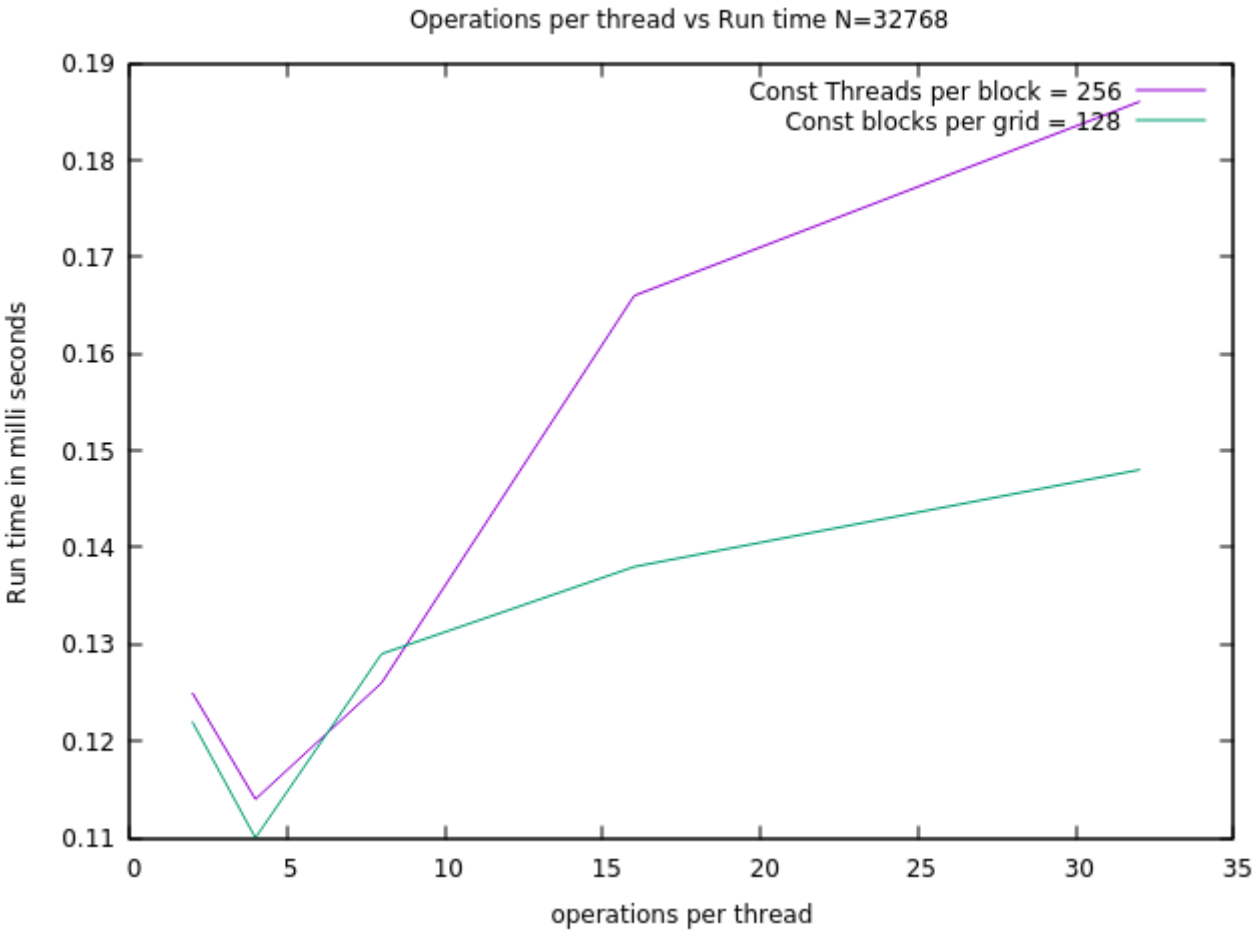


(ii) Kepping number of blocks per grid = 128 and varying number of threas as number of operations per thread increases.

Which gives optimal value = 4 .

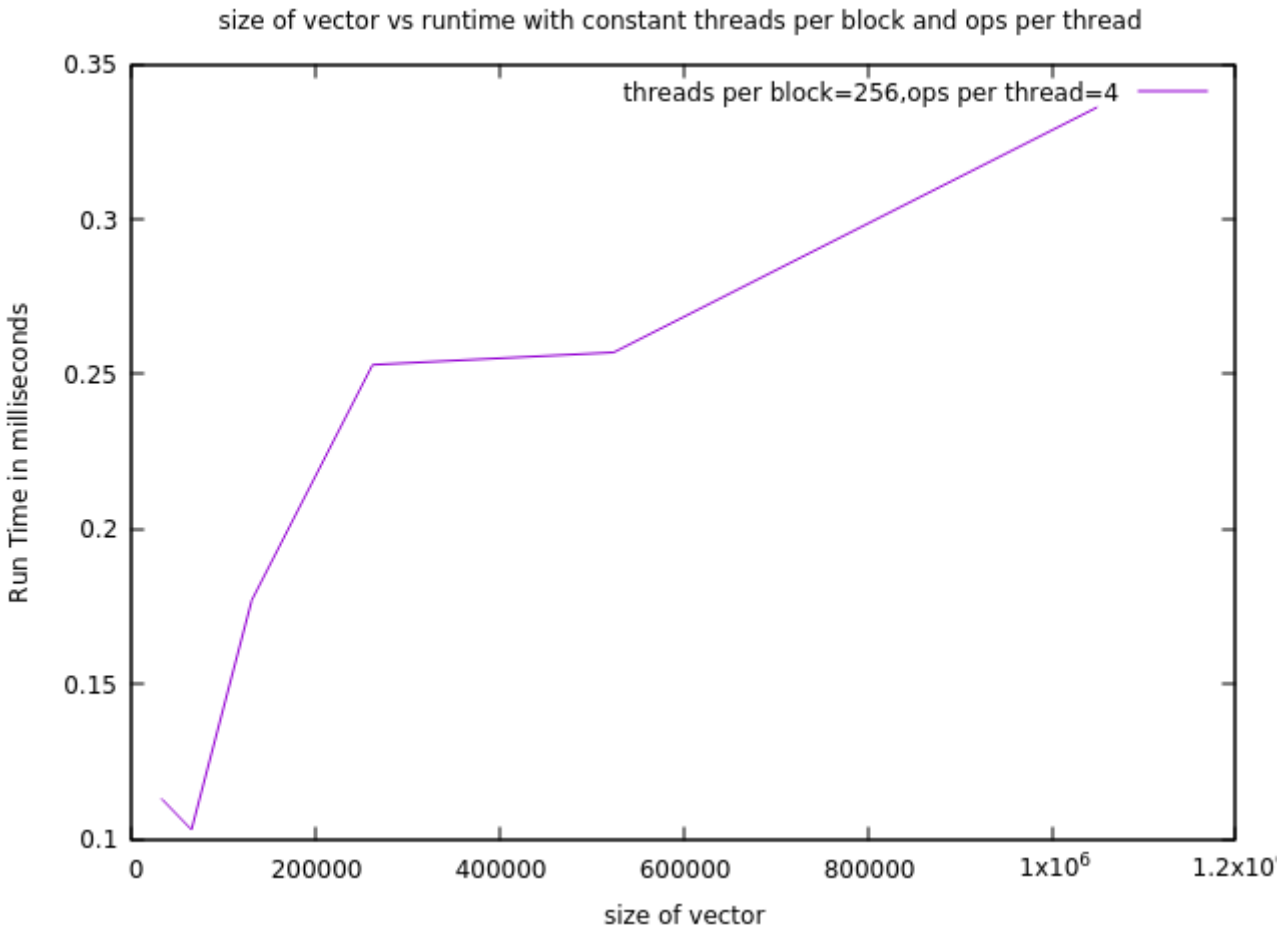


Combining both graphs into one : Both gives optimal at 4

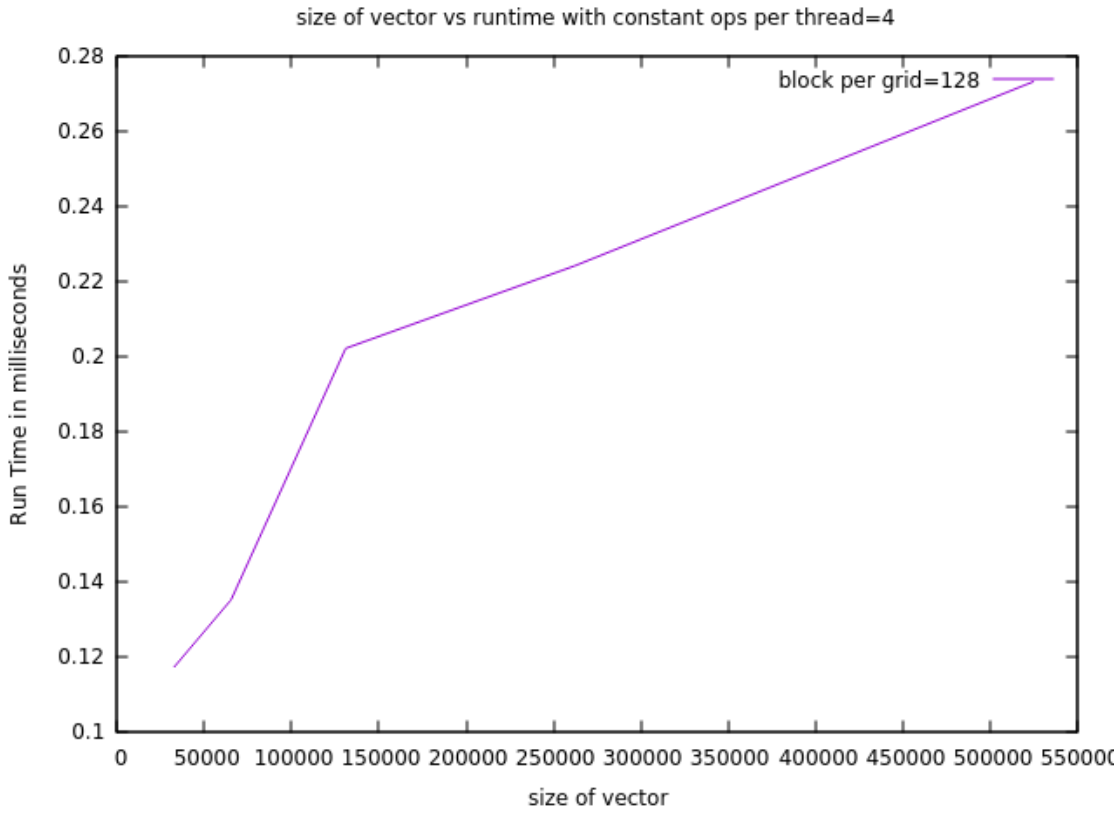


Q6 : The runtime configuration (using optimal from above):
num of ops per thread = 4

(i) Keeping Number of threads per block to optimal =256 , and increasing N from 2^15 to 2^20



(ii) Keeping Number of block per grid to optimal = 128 , and increasing N from 2^15 to 2^20



Combining both graphs in one :