

Cuda Programming Hands-on Lab 2

Block Sizes with SumMatrix.cu (follows Lecture 6)

Instructions

Marks: 10*Due date: 12 noon on Wednesday 24th April 2024*

Please submit a pdf of the results worksheet and a **softcopy of changes made to the code for Tasks 4 and 5.**

By submitting your assignment via the RUConnected link, you declare that the assignment submitted is entirely your own work, unless noted otherwise.

Aim

Investigating the effect of changing block sizes on a simple Cuda program

Tasks

1. Download sumMatrix.cu on RUConnected (either MS or Linux version). Runtime arguments are: blockDim.x and blockDim.y. Note that you are using a LARGE data matrix of size $2^{12} \times 2^{12}$.
2. Run the kernel for different 2D thread block sizes as given in the table below: 32x32, 32x16, 16x32, 16x16, and four realistic sets of your own choice of dimensions. (Note that a 2D gridDim is automatically determined.) Also be careful NOT to exceed the max threads per block limitation (1024).
3. Note the execution times of the kernel. Also fill in the gridDim as determined automatically.
4. Rerun steps 2 and 3 with each block size configuration, but change the code to use a 1D gridDim. Be sure to fix the way the data is allocated to threads as well.
5. Finally, repeat step 4, but change the code to allow each thread to sum 16 data items instead of a single item. NB. It is up to you how you allocate the data to threads, but you should use **16x fewer threads to complete the task**. Fill in the final column in the table.
6. Tabulate the results as shown on the next page. Note that you should add in at least 4 of your own configurations for the 2D block size.
7. What can you conclude from your experiments in terms of kernel execution time?

Note that the sumMatrix.cu code uses the error helper function from the CUDA samples directory. Please ensure that you have the correct include path set up in your CUDA installation. Or use something like this (note **yellow highlight** is for 750ti devices):

nvcc -arch=sm_50 -I <includepath> myprog.cu -o myprog

MS: **nvcc -I**

"C:\ProgramData\NVIDIA Corporation\CUDA Samples\v11.8\common\inc"
myprog.cu -o myprog

Linux: **nvcc -I/usr/local/cuda/samples/common/inc myprog.cu -o myprog**