GPGPU PROGRAMMING USING CUDA

Prac 4

Block Sizes with SumMatrix.cu

(follows Lecture 9)

1. Download sumMatrix.cu on RUConnected - runtime arguments are: blockDim.x and blockDim.y. Note that you are using a data matrix of size 10^{12} x 10^{12} !
2. Run the kernel for different 2D thread block sizes as given in the table below: 32x32, 32x16, 64x32, 64x64, and four realistic sets of your own choice of dimensions. (Note that a 2D gridDim is automatically determined.)
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. Now, repeat step 4, but change the code to allow each thread to sum 16 data items instead of a single item (it is up to you how you allocate the data to threads). 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?
8. Submit hardcopy of results by 8:30am on Thursday, 26th April to obtain the relevant course credit.

Prac 4: Block Sizes

Name: **Marks: 20**

2D Block size	2D Grid size (automatically determined) Step 3	Kernel execution time (ms) Step 3	1D Grid size Step 4 / Step 3	Kernel execution time (ms) (1 datum per thread) Step 4 / Step 3	Kernel execution time (ms) (16 data per thread) Step 5
32x32					
32x16					
64x32					
64x64					

Conclusion: