Tiled Matrix Multiplication (Download ZipFile)

- 1) Unpack the lab template into your SDK projects directory
- 2) Edit the source files matrixmul.cu and matrixmul_kernel.cu to complete the functionality of the matrix multiplication on the device. The two matrices could be any size, but the resulting matrix is guaranteed to have a number of elements less than 64,000.
- There are several modes of operation for the application. Note that the file interface has been updated to allow the size of the input matrices to be read.

No arguments: The application will create two randomly sized and initialized matrices such that the matrix operation M * N is valid, and P is properly sized to hold the result. After the device multiplication is invoked, it will compute the correct solution matrix using the CPU, and compare that solution with the device-computed solution. If it matches (within a certain tolerance), if will print out "Test PASSED" to the screen before exiting.

One argument: The application will use the random initialization to create the input matrices, and write the device-computed output to the file specified by the argument.

Three arguments: The application will read input matrices from provided files. The first argument should be a file containing three integers. The first, second and third integers will be used as M.height, M.width, and N.height. The second and third function arguments will be expected to be files which have exactly enough entries to fill matrices M and N respectively. No output is written to file.

Four arguments: The application will read its inputs from the files provided by the first three arguments as described above, and write its output to the file provided in the fourth.

Note that if you wish to use the output of one run of the application as an input, you must delete the first line in the output file, which displays the accuracy of the values within the file. The value is not relevant for this application.

4) Answer the following questions:

In your kernel implementation, how many threads can be simultaneously executing on a GeForce GTX 280 GPU, which contains 30 Streaming Multiprocessors. Use nvcc --ptxas-options="-v" matrixmul_kernel.cu to see the resource usage of your kernel (although compilation will fail, it will only do so after compiling the kernel and displaying the relevant information.)

Grading:

Your submission will be graded on the following parameters.

Demo/knowledge: 25%

- Produces correct result output files for provided inputs

Functionality: 40%

- Runs correctly when executed on a G80 device.
- Shared memory is used in the kernel to mask global memory access latencies.

Report: 35%

- Answer to provided question.