NCIT Summer School 2012: GPU Computing Workshop
University Politehnica of Bucharest
6 June – 8 June 2012

Assignment 1: Parallel Reduction

Edit the provided source files **vector_reduction.cu** and **vector_reduction_kernel.cu** to complete the functionality of the vector reduction (addition) on the GPU. The vector reduction operation takes a vector (array) of values and reduces it to a single value via some two variable function, in this case addition. You may assume that the array will have exactly 1024 elements.

There are two modes of operation for the application:

- a) No arguments: The application will create a randomly initialized array to process. After the device kernel is invoked, it will compute the correct solution value using the CPU, and compare that solution with the device-computed solution. If it matches (within a certain tolerance), it will print out "**Test PASSED**" to the screen before exiting.

- b) One argument: The application will initialize the input array with the values found in the file provided as an argument (space delimited).

In either case, the program will print out the final result of the CPU and GPU computations, and whether or not the comparison passed.

Please provide your modified source files and a text file with your answers to the following questions:

- How many times does your thread block synchronize to reduce the array of 1024 elements to a single value?

- Over the life of the program, how many warps will be adversely affected by warp divergence? Warp divergence happens when threads within a warp take different code paths through the program.

These files should then be packaged (zip, tar, rar, etc) and emailed to: ncit-hw@andrewseidl.com

If you run into issues compiling, try the following:

- add the 'common' include directory from the SDK:
  - -I$NVSDKCOMPUTE_ROOT/C/common/inc
- add the 'sdk' library directory:
  - **-L$NVSDKCOMPUTE_ROOT/C/lib**
- link against libcutil_x86_64:
  - -lcutil_x86_64