ECEC-413: Introduction to Parallel Computer Architecture
CUDA Programming Lab 2: Parallel Vector Reduction

Prof. Naga Kandasamy, ECE Department, Drexel University

January 27, 2011

The lab is due on February 4, 2011. You may work on the problems in teams of up to two people.

Develop a GPU-based program to sum-reduce a large array of floating-point numbers to a single value.

Edit the source files `vector_reduction.cu` and `vector_reduction_kernel.cu` to complete the functionality of the parallel addition reduction on the GPU. The size of the array is guaranteed to be equal to $5,000,000$ (five million) elements for this assignment. Note that you may need to define a 2D grid of thread blocks to process an array this large.

Your program should accept no arguments. The application will create a randomly initialized array to process. After the GPU kernel is invoked, it will compute the correct solution value using the CPU, and compare that solution with the GPU-computed solution. If the solutions match (within a certain tolerance), it will print out "Test PASSED" to the screen before exiting.

You must e-mail me all of the files needed to run your code as a zip file called `lab_2.zip`.

This lab will be graded on the following parameters:

- *Correctness*: 50 points.
- *Performance*: 25 points. This includes the efficient and clever use of shared memory. Also, for best performance, multiple invocations of the kernel will be necessary. For example, during the first step, each thread block will reduce the set of values assigned to it to a single value (using shared memory) and store these partial sums in GPU global memory. The kernel is invoked again to reduce these partial sums further, and so on.
- *Report*: 25 points. A two/three page report describing how you designed your kernel (use code or pseudocode to clarify the discussion) and the amount of speedup obtained over the serial version.