CI853 Lab 1

CUDA Vector Add

(based on the GPU Teaching Kit -- Accelerated Computing)

Objective

The purpose of this lab is to introduce the student to the CUDA API by implementing vector addition. The student will implement vector addition by writing the GPU kernel code as well as the associated host code.

Prerequisites

None at this time

Instructions

Edit the code in the code tab to perform the following:

- Allocate device memory
- Copy host memory to device
- Initialize thread block and kernel grid dimensions
- Invoke CUDA kernel
- Copy results from device to host
- Free device memory
- Write the CUDA kernel

Instructions about where to place each part of the code is demarcated by the //@@ comment lines.

Local Setup Instructions

The most recent version of the template for this lab (source code and instructions) is at ~wagner/ci853/labs in the local machine.

You have to make a copy of each lan directory to your home account and work with your copy. The professor will give you instructions as to where/how to hand in your final solution.

The executable generated as a result of compiling your lab solution must run using the following command:

./VectorAdd -i <intput1.raw> <input2.raw> -o <output.raw>

where:

<input0.raw>,<input1.raw> is the input dataset, and <output.raw> is an optional path to store the results. Some datasets are already generated in the lab directory.

Questions

1. How many floating operations are being performed in your vector add kernel? EXPLAIN.
2. How many global memory reads are being performed by your kernel? EXPLAIN.
3. How many global memory writes are being performed by your kernel? EXPLAIN.
4. Describe what possible optimizations can be implemented to your kernel to achieve a performance speedup.
5. Name three applications of vector addition.

Code Template

The code template at the lab directory is suggested as a starting point for students. The code handles the import and export as well as the checking of the solution. Students are expected to insert their code is the sections demarcated with //@@. Students expected to leave the other code unchanged. The template uses a wblib (a file header included by the template). At this point the original wblib is not functional, so the professor is providing a “workarround” version of the lib.

Modified by W.Zola/UFPR