Lab Assignment 1
Due Oct 7, 2023 at 11:59pm

1 Objective

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

2 Instructions

The code template in `template.cu` provides a starting point and handles the import and export as well as the checking of the solution. Students are expected to insert their code in the sections demarcated with `//@@`. Students are expected to leave the other code unchanged.
Edit the skeleton code to perform the following:

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

Compile the template with the provided `Makefile`. The executable generated as a result of compilation can be run using the following code:

```
./VectorAdd_template -e <expected.raw> -i <input1.raw>,<input2.raw> -o <output.raw> -t
vector
```

, where `<expected.raw>` is the expected output, `<input0.raw>,<input1.raw>` is the input dataset, and `<output.raw>` is an optional path to store the results. `README.md` has details on how to build `libgputk`, `template.cu` and the dataset generator.

When you work on the department's computing cluster, you need to request a computing node with NVIDIA GPUs using the following command:

```
srun -p titanxp -N 1 -n 6 --mem=32G --gres=gpu:2 --pty /bin/bash -l
```

Note that you need to run `ccmake` and compile your kernel on the master node because the required utilities are not installed on GPU computing nodes (We are working on getting them installed on GPU computing nodes). Then, you can execute your compiled program on a GPU computing node.

3 What to Turn in

Submit a report that includes the following:

1. Answers to the following questions.
   (a) How many floating point operations are being performed in your vector add kernel? Explain.
   (b) How many global memory reads are being performed by your kernel? Explain.
   (c) How many global memory writes are being performed by your kernel? Explain.
2. Your version of `template.cu`.
3. Execution times of the kernel with the input data generated by the dataset generator (in a table or graph). Please include the system information where you performed your evaluation. For time measurement, use `gpuTKTime_start` and `gpuTKTime_stop` functions (You can find details in `libgputk/README.md`).