ECE574: Cluster Computing – Homework 8
CUDA

Due: Tuesday 18 April 2017, 3:30pm

1. **Background**

   - In this homework we will take the sobel code from earlier homeworks and parallelize it using CUDA.
   - You may work in a group for this Homework.

2. **Setup**

   - For this assignment you will need to log into the "quadro" machine that has the NVIDIA GPU. First log into the same Haswell-EP machine we used in previous homeworks. As a reminder, use the username handed out in class and ssh in like this
     `ssh -p 2131 username@weaver-lab.eece.maine.edu`
     From there, type "ssh quadro"
   - Download the code template from the webpage. You can do this directly via
     `wget http://web.eece.maine.edu/~vweaver/classes/ece574_2017s/ece574_hw08_code.tar.gz`
     to avoid the hassle of copying it back and forth.
   - Decompress the code
     `tar -xzvf ece574_hw08_code.tar.gz`
   - Run `make` to compile the code.
   - You may use your own code from a previous assignment as a basis for this assignment. (Alternately some really poorly-optimized sample code is provided).

3. **Moving "combine" to the GPU (8 points)**

   We will first convert the "combine" routine to run on the GPU.

   Edit the file `sobel_coarse.c`

   Be sure to comment your code!

   A suggested first implementation is this:

   (a) Allocate device buffers for sobelx, sobely, and the output using `cudaMalloc()`

   (b) Copy the local `sobel_x.pixels` and `sobel_y.pixels` to the device using `cudaMemcpy()`

   (c) Run the kernel

   (d) Copy the results back into `new_image.pixels` using `cudaMemcpy()` (be sure to get the direction right)

   (e) Add calls to PAPI so you can print separately the total time taken by the cudaMalloc() and cudaMemcpy calls and that taken by the combine kernel.

(f) Some hints: to debug that your kernel works, you can first set all output to 0xff and verify you get an all-white image back.
Next make the output just be a copy of the sobel_x input and make sure you get back what you passed in.
When you call `sqrt()` inside the kernel, be sure to cast to double before taking the sqrt, otherwise CUDA might complain about you trying to use a host version of the function.
nvcc uses C++ to compile things, so be sure you aren't using C++ reserved words (such as "new") as variable names

slurm is not installed on this machine, just run the program at the command line.

Run on the `butterfinger.jpg` input

Report the combine time taken by the C version (`sobel_before`) and the combine and alloc/cpy time for teh CUDA version (`sobel_coarse`).

4. **Fine Grained (2 points)**

Modify the code so that the convolves are done on the GPU.

First copy your code to `sobel_fine.c` and edit it.

Here are some hints. You don't have to do it this way, but it helps.

- The hardest part here is getting the grid/block/thread count right.

```
int blockId = blockIdx.y* gridDim.x+ blockIdx.x;
        int i = blockId * blockDim.x + threadIdx.x;
```

- You might first want to convert your previous combine code to this and see that it works on the larger cake image.
- For the generic convolve, you will also need to upload the sobelx and sobely matrices to the device. A simple array of 9 ints is probably best.
- For each point "i" add in the 9 scaled values.
- Remember there are separate RGB colors so you will need to add in points -3, 0, +3 for example.
- Also make sure you have code that skips the first and last rows, as well as the first and last columns (which is three columns, remember RGB).
- Again it might be helpful to output the sobel_x output and run on the butterfinger input and get that to match exactly before running with both sobel_y and combine hooked up.

Report the time output from `sobel_before` and compare it to `sobel_fine` when run on the `IMG_1733.JPG` input. Did it run faster on the CPU or GPU?

5. **Submitting your work.**

- Be sure to edit the README to include your name, as well as the timing results, and any notes you want to add about your something cool.
- Run `make submit` and it should create a file called `hw08_submit.tar.gz`. E-mail this file to me.
- e-mail the file to me by the homework deadline.