

CSCI-UA.0480-003

Parallel Computing

Lab 3

Before you start:

- To calculate the time of each program in this lab use the Linux command *time*. Report the **REAL** part (neither usr nor sys).
- After you login to your CIMS account, you need to ssh to one of the following: cuda1, cuda2, cuda3, cuda4, or cuda5
- The source code, containing both device and host code, has extension .cu
- You compile with **nvcc** progname.cu
- Don't forget to `#include <cuda.h>`
- A very useful API is **cudaGetDeviceProperties()** check it up.

You will be provided with a sequential program maxseq.c that can be compiled and run with:

```
gcc -o maxseq maxseq.c
```

Then you execute it with:

```
./maxseq x
```

where x is a positive number.

The program that will allocate an array of long, fill it with random numbers from 0 to x and then call a function getmax that returns the maximum number of the array.

Coding:

- A. Keep a copy of the sequential version maxseq.c
- B. Make another copy of the above file and rename it maxgpu.cu
- C. Modify that file maxseq.cu such that instead of calling getmax you call a kernel getmaxcu() that outsources the job of finding the maximum number to the GPU. Let the main function allocate and populate the array in the host. Then you need to allocate in the device memory for that array and transfer the value from the host to the device, calculate the maximum in the device, then transfer that number back to the host.
- D. Let the printf() in the host print the maximum number as in the sequential version.
- E. Measure the time for both the sequential and the CUDA versions.

Reporting Results (the report):

Put one line at the beginning stating which machine have you used to do the experiments.

1. Explain how you picked the block and grid sizes/dimensions, and justify your choices.
2. Write the command line you used to compile the CUDA version.
3. Draw a bar graph as follows:
 - a. The Y-axis is the time
 - b. X-axis is the length of the array (i.e. the number of elements), try with 1000, 10000, 100000, 1000000 (one million), 10000000 (ten million), and 100000000 (hundred million)
 - c. For each value of X show two bars: the sequential version and the CUDA version
4. Explain the behavior in the graph.

What to submit (in single zip file called **yourlastname.firstname.zip)?**

- the source code maxgpu.cu
- the report in pdf
- email it to the grader with **subject line: lab 3 submission**

Grading policy:

- maxgpu.cu compiles and run correctly **10 points**. Otherwise, we will look at the code and give you a partial credit that does not exceed 10/20
- #1 in the reporting results above: **2 points** (Let's see how you design your project)
- #2 **1 point** (Let's see which parameters you picked, if any!)
- #3 **2 points**
- #4 **5 points**

Have Fun!