EECS 368/468 – Lab 1
Programming Massively Parallel Processors with CUDA

Matrix Multiplication in CUDA

In this first assignment, you will work in teams of two to write a matrix multiplication code in CUDA. The lab is due on Friday Jan 23 at 11:59pm. The late submission policies are explained in the syllabus. Please follow the instructions below to do the lab.

Install the CUDA SDK at your Wilkinson Lab account

1. Login to a machine at Wilkinson Lab (Tech M338). If you wish to work remotely (e.g., via ssh), the Wilkinson Lab hostnames are:
   alfred.eecs.northwestern.edu
   bane.eecs.northwestern.edu
   batgirl.eecs.northwestern.edu
   batman.eecs.northwestern.edu
   clayface.eecs.northwestern.edu
   cobblepott.eecs.northwestern.edu
   freeze.eecs.northwestern.edu
   gordon.eecs.northwestern.edu
   gotham.eecs.northwestern.edu
   harley.eecs.northwestern.edu
   huntress.eecs.northwestern.edu
   hush.eecs.northwestern.edu
   joker.eecs.northwestern.edu
   killercroc.eecs.northwestern.edu
   madhatter.eecs.northwestern.edu
   nightwing.eecs.northwestern.edu
   poisonivy.eecs.northwestern.edu
   ras.eecs.northwestern.edu
   riddler.eecs.northwestern.edu
   robin.eecs.northwestern.edu
   scarecrow.eecs.northwestern.edu
   selina.eecs.northwestern.edu
   twoface.eecs.northwestern.edu

2. Setup the CUDA environment. If you use csh or tcsh:
   ```
   source /usr/local/cuda-5.0/cuda-env.csh
   ```
   or if you use the bash shell:
   ```
   . /usr/local/cuda-5.0/cuda-env.sh
   ```

3. Install SDK in your home directory
   ```
   cp -r /home/hardav/NVIDIA_CUDA-5.0_Samples ~/;
   cd ~/NVIDIA_CUDA-5.0_Samples;
   make
   ```

4. Test by finding out the characteristics of your GPU device
   ```
   cd ~/NVIDIA_CUDA-5.0_Samples/1_Utilities/deviceQuery;
   ./deviceQuery
   ```

Install the labs scaffold at your Wilkinson Lab account

5. Download the labs-scaffold tarball from blackboard into some directory in your Wilkinson Lab account.

6. Untar the code scaffold tarball:
   ```
   tar xvf EECS468-CUDA-Labs-Scaffold.tgz
   ```

Install lab1 at your Wilkinson Lab account

7. Download the lab1 tarball from blackboard into some directory in your Wilkinson Lab account.

8. Install the lab1 tarball and compile the lab sources:
   ```
   cd EECS468-CUDA-Labs
   tar xvf EECS468-CUDA-Lab1.tgz
   make
   ```

Complete the assignment

9. **Remember**: every time you login to the Wilkinson lab to program in CUDA you need to setup the CUDA environment. If you use csh or tcsh:
   ```
   source /usr/local/cuda-5.0/cuda-env.csh
   ```
   or if you use the bash shell:
   ```
   . /usr/local/cuda-5.0/cuda-env.sh
   ```

10. **Assignment Task:**
    Edit the following two functions
    `MatrixMulOnDevice(...)` in the source file `matrixmul.cu`
    `MatrixMulKernel(...)` in the source file `matrixmul_kernel.cu`
    to implement matrix multiplication on the GPU device. **Do not change any other piece of the source code**. The size of the matrix is defined such that one thread block will be sufficient to compute the entire solution matrix.

    The source code you will be working on is at :
    `EECS468-CUDA-Labs/labs/src/lab1`

    Compiling your code produces the executable:
    `EECS468-CUDA-Labs/labs/bin/linux/release/lab1`

11. There are several modes of operation for the application.

    a. No arguments: the application will create two randomly initialized matrices to multiply. After the device multiplication is invoked, it will compute the correct solution matrix using the CPU, and compare that solution with the device-computed solution. If it matches (within a certain tolerance), it will print out "`Test PASSED`" to the screen before exiting.

    b. One argument: the application will use the random initialization to create the input matrices, and write the device-computed output to the file specified by the argument.

c. Two arguments: the application will initialize the two input matrices with the values found in the files provided as arguments. No output is written to file.

d. Three arguments: the application will read its inputs from the files provided by the first two arguments, and write its output to the file provided in the third.

Note that if you wish to use the output of one run of the application as an input, you must delete the first line in the output file, which displays the accuracy of the values within the file. The value is not relevant for this application.

12. Hand in your solution:

a. Create a tarball of your solution, which includes the contents of the `lab1` source folder provided, with all the changes and additions you made to the source files. Please make sure you do a `make clobber` before submitting; don't include object code or executables, as these are typically large files:

```
cd EECS468-CUDA-Labs/labs/src/lab1;
make clobber
tar cvfz EECS468-lab1-YourNames.tgz *
```

where you should replace `YourNames` with the **concatenated last names of the team members**.

b. Submit your solution tarball via blackboard. Only one student in a team needs to submit. If you submit multiple times, only the latest submission will be graded. Please do all submissions from the same account.

Good luck!