Lab #4: Basic Programming Exercises of CUDA

Lab 4 is due Wednesday, Nov. 29. Your lab report and source code must be submitted by **1:50PM** before class. The late policy applies to this lab project.

For problems working on matrix multiplication, an implementation template (mm.c) is provided. The template includes a basic implementation of matrix multiplication on CPU.

IMPORTANT NOTE: You should verify the correctness of ***every*** GPU version of mm that you implement. The way of verification is to transfer the GPU result back to CPU and compare it with the outcome of the CPU matrix multiplication version as implemented in mm.c. The metric of comparison is Root Mean Square Error (RMSE), as discussed in class.

When you measure the performance of the programs you write for this lab, make sure the measurement is stable and consistent. A simple way to do that is to repeat the execution multiple times and use the average.

Problem 1: NVIDIA CUDA

The machine used is "cpeg655.ece.udel.edu". The machine is equipped with four NVidia graphic cards. You should try logging on to cpeg655.ece.udel.edu and inform the instructor as soon as possible if you have any problem accessing the machine. Log in with your ECE acad account.

SLURM (Simple Linux Utility for Resource Management) is used to queue and run jobs on the CUDA hardware. Jobs can be queued to run using the srun command:

- srun -N1 --gres=gpu:G <command>
  - -N1 indicates the number of nodes (computers). Currently, only one CUDA machine exists, so don't change this option.
  - G is the number of GPU cards the job requires. Currently 4 cards are available for jobs on cuda.acad.

Additional SLURM commands are:

- squeue - Shows the current state of the SLURM queue for cuda.acad.
- sinfo - Shows info about the current SLURM configuration.
- scancel - Used to cancel a currently running job.

More information and documentation about SLURM is available at https://computing.llnl.gov/linux/slurm/documentation.html

CUDA is installed to /usr/local/cuda. You may need to add /usr/local/cuda/bin to your path, and /usr/local/cuda/lib64 to your LD_LIBRARY_PATH. Please avoid setting your LD_LIBRARY_PATH if things are working without setting it.

SLURM is installed in /usr/bin. You usually don’t need to add /usr/bin to your path.

task a:

Write a multithreaded program using the CUDA programming model to compute matrix multiplication, that is, *C=AxB*, where *A, B, C* are three matrices with size *N x N*. One thread computes one element of the product matrix *C*. Use only one thread block to compute the whole matrix. Measure and report the performance of two sizes, *N=16* and *N=32*.

task b:

Write a multithreaded program using the CUDA programming model to compute matrix multiplication using the tiling algorithm. One thread still computes only one element of the product matrix, but a thread block computes only a tile of the matrix *C*, and uses multiple thread blocks to compute the whole matrix. Measure and report the performance of the matrix size *N=2048*, and for two tile sizes, *8* and *16*.

task c:

Extend the version of 1.b by making one thread compute a NB*NB tile of matrix C. Also try to change the number of threads NT*NT in a thread block. You can use multiple thread blocks. If the number of thread blocks is NK*NK, the relationship between NB, NT, NK and the matrix size N is $N^2 = NB^2 * NT^2 * NK^2$. Find the best NB and NT for N=2048. NB and NT should be power-of-2 values.

What to submit:

The package you submit for this lab should include your source code, performance results and analysis, and brief description that you think might help the instructor understand your code, e.g., about any design choices you make in your program. The instructor will compile, run and measure the performance of your code. Therefore, you should also describe how to compile and run your code in your submitted documentation.

How to Submit:

Copy your lab report, which is a .pdf, a .doc, or a .html file, and all your source code into an empty directory. Assuming the directory is "submission", make a tar ball of the directory using the following command:

```
tar czvf [your_first_name]_[your_last_name]_lab4.tar.gz submission.
```

Replace [your_first_name] and [your_last_name] with your first name and your last name.

Submit the tar ball. The submission time will be used as the time-stamp of your submission.
