CSE 520 Computer Architecture -- Spring 2020

Programming Assignment 3

In this assignment, you will execute matrix multiplication on different devices (CPU or GPU) using an OpenCL program. The goal of this assignment is to compare the performance of single thread execution with multi-thread execution as well as to experience OpenCL programming framework.

Requirement

You have to measure the execution time on following configurations:

- Matrix size: 512x512, 1024x1024, 2048x2048
- Reference C/C++ matrix multiplication.
- OpenCL naive matrix multiplication on CPU
- OpenCL naive matrix multiplication on GPU
- OpenCL tiled matrix multiplication on CPU (tile size = 8 and 16)
- OpenCL tiled matrix multiplication on GPU(tile size = 8 and 16)

All these configurations must be put together and executed under a **single** main.cpp. To do so, you need to duplicate some variables and specific OpenCL structure for different devices (say CPU and GPU). For example, two OpenCL command queues and contexts must be created for CPU and GPU separately.

Sample programs, including an OpenCL host program and a kernel file, are provided as **main.cpp** and **matrix_mul.cl** respectively. The main.cpp has a reference implementation of matrix multiplication and an OpenCL framework to execute kernel function on the GPU target. The default matrix size is 1024x1024.

In the kernel file, there are two kernel functions of matrix multiplication, i.e., a naive version “matrix_mul”, and a tiled version “matrix_mul_tile”. Please refer to the course material for the details of these implementations. You don’t have to modify kernel file in this assignment. Only main.cpp is required to change to perform the following steps:

1. initialize the matrices with random numbers between 20 to -20.
2. adjust the matrix size by modifying the SIZE macro definition.
3. run the sequential version of matrix multiplication and measure the execution time
4. run the “matrix_mul” kernel on CPU and GPU and measure the execution times
5. run the “matrix_mul_tile” kernel on CPU and GPU and measure the execution times

Environment

All measured execution time must be reported from the Linux systems in BYENG 217 lab. The systems are equipped with Nvidia’s Quadro RTX 5000 GPU and with OpenCL driver installed. To log in the machines remotely, you need to connect to ASU’s SSL VPN first using CISCO AnyConnect SSLVPN software which can downloaded from myapps.asu.edu. Then, you can ssh to connect to the Linux machines in BYENG 217 lab and log in with your ASURITE id. Remote desktop connection is not enabled on these machines. However, you can use ssh with X tunneling (ssh -X or ssh -Y) from a Linux machine to send X windows back to the client.

You can choose a SSH client (e.g., PuTTY for Linux/Windows and MobaXterm for Windows) to connect to the lab machines. Note that there are terminal-based text editors, vi, vim, and emacs, installed in the lab machines. If you want to use a GUI-based text editor, you can do the editing with any text editor in your local machine and scp or sftp files to the lab machines.

The current list of the host machines are:

en6169301-217.etslabs.dhcp.asu.edu	en6169314-217.etslabs.dhcp.asu.edu*
en6169302-217.etslabs.dhcp.asu.edu	en6169315-217.etslabs.dhcp.asu.edu*
en6169303-217.etslabs.dhcp.asu.edu	en6169316-217.etslabs.dhcp.asu.edu*
en6169304-217.etslabs.dhcp.asu.edu	en6169317-217.etslabs.dhcp.asu.edu*
en6169305-217.etslabs.dhcp.asu.edu	en6169318-217.etslabs.dhcp.asu.edu*
en6169306-217.etslabs.dhcp.asu.edu*	en6169319-217.etslabs.dhcp.asu.edu*
en6169307-217.etslabs.dhcp.asu.edu	en6169320-217.etslabs.dhcp.asu.edu *
en6169308-217.etslabs.dhcp.asu.edu	en6169321-217.etslabs.dhcp.asu.edu*
en6169310-217.etslabs.dhcp.asu.edu	en6169322-217.etslabs.dhcp.asu.edu*
en6169311-217.etslabs.dhcp.asu.edu	en6169323-217.etslabs.dhcp.asu.edu
en6169312-217.etslabs.dhcp.asu.edu	en6169324-217.etslabs.dhcp.asu.edu*

Note that some machines in the lab may be off-line temporarily. If you cannot get connected with a specific machine, please try another one. Also, once you log in, you should run the command “nvidia-smi” to check Nvidia driver and GPU status. At this moment, in the above list, the machines marked with an asterisk work properly. Our IT staffs will try to fix all other machines in the next few days.

Reporting:

The following table is provided for you to record the execution time. In the report, you only have to include this table without answering any questions. In addition, your report should include:

1. The models of CPU and GPU of the machine on which you collect the measurement data. Note the machines in BYENG 217 lab have two computing platforms, i.e., Intel and Nvidia, where the Intel platform only has a CPU device and the Nvidia platform only has a GPU device.
2. A console (terminal) screenshot of running main.cpp and its output message.

Matrix size	Reference C implementation (single thread)	OpenCL on CPU			OpenCL on GPU		
		Normal kernel	Tiled kernel (tile_size=8)	Tiled kernel (tile_size=16)	Normal kernel	Tiled kernel (tile_size=8)	Tiled kernel (tile_size=16)
512x512							
1024x1024							
2048x2048							

Due Date

The assignment is due by April 29 at 11:59pm.

What to Turn in for Grading

1. Create a working directory, named “cse520-assgn03-LastName_FirstInitial”, for the assignment to include
   - A pdf report to include a list of CPU/GPU models, a screenshot, and the table of measured execution times. Don’t forget to add your name and ASU id in the report.
   - The modified main.cpp and matrix_mul.cl.
   - A Makefile to compile your program
   - A README file to explain how to compile and run your program.
2. Compress the directory into a zip archive file named cse520-assgn03-LastName_FirstInitial.zip. Note that any object code or temporary files should not be included in the submission.
3. Submit the zip archive to the course Canvas by the due date and time. No email submission will be accepted.
4. There will be 20 points penalty per day if the submission is late. Note that submissions are time stamped by Canvas. If you have multiple submissions, only the newest one will be graded. If needed, you can send an email to the instructor and TA to drop an early submission.
5. The assignment must be done individually. No collaboration is allowed, except the open discussion in the forum on Canvas. The instructor reserves the right to ask any student to explain the work and adjust the grade accordingly.
6. ASU Academic Integrity Policy (http://provost.asu.edu/academicintegrity), and FSE Honor Code (http://engineering.asu.edu/integrity) are strictly enforced and followed.

Important Notes:

- Using fake data in the report will be treated as unethical practice and a violation of academic integrity will be reported.
- Only the submissions in Canvas will be graded. Any modifications to your submission or replacement of files will be handled as a new submission and may be subject to late submission penalty.