CoE 163

Computing Architectures and Algorithms

Capstone Exercise

Academic Period: 2nd Semester AY 2020-2021
Workload: 3 hours
Synopsis: (Very) basic GPU programming with CUDA
Submission Platform: UVLe

Description

Parallel programming (or parallel computing) uses multiple computing modules to solve a problem. Such computing is very useful in reducing the time it takes for an algorithm to finish and perform repetitive but independently-running tasks at the same time. 3D rendering, video encoding, and cryptocurrency mining are one of the several fields where parallel programming is extensively used.

One of the popular platforms for parallel computing is Compute Unified Device Architecture (CUDA), which is developed by NVIDIA since 2007. As of this time of writing, NVIDIA has become a leading firm specializing in the development of graphics processing units (GPUs) that can support parallel computing. Recently released GPU models can cost a fortune due to its demand in the cryptocurrency sector. However, CUDA can still be supported by their relatively cheap and old GPUs.

Because of your interest in cryptocurrency mining, you have finally decided to learn CUDA. The API can only be accessed through C or C++ since it is a bare metal API. For your first foray into CUDA programming, you have considered enrolling in a week-long seminar on experimenting with a Python wrapper of this API. After a session, the instructor gave a homework on parallelizing matrix addition by linking to a Jupyter notebook hosted at

Google Colab. More instructions can be found on the notebook itself, but the summary of the task is to formulate three kernels that perform matrix addition, invoke these kernels, and profile their runtimes.

You cannot wait to start on this task, and so you have perused through the notebook and opened the reference readings. With this and future knowledge on the platform, you can now realize little-by-little your dream of creating a cryptocurrency with a very low carbon footprint.

Additional Description/Requirements

The notebook the instructor sent to you also requires that you duplicate the notebook and fill in the missing parts with your code. A short journal to be written at the end of the notebook is also required.

Due to the fast nature of the seminar, you were only given a mere five days to complete the homework.

Grading Rubric

5% GPU listing routine
5% Addition routine
10% Element parallelization
20% Row parallelization
20% Column parallelization
15% Kernel runs
25% Short journal within notebook

Acknowledgement

This CE has been adapted from one of the exercises offered by CS 239 for 2nd Semester AY 2020-2021.