CS4345 General Purpose Computation on GPU (2016/2017 Semester 1)

Programming Assignment 4

Release Date: 3 November 2016, Thursday
Submission Deadline: 20 November 2016, Sunday, 11:59 PM

LEARNING OBJECTIVES

Writing CUDA programs using CUDA C and the Thrust library to perform general-purpose computation on the GPU. After completing the programming assignment, you should have

- learned to use some of the routines in the Thurst library to perform data-parallel primitive operations on the GPU,
- learned to perform stream compaction on the GPU using the scan-and-scatter approach.

TASKS

In this programming assignment, you are required to complete the implementation of a function and two CUDA kernels to compute a sorted version of an input integer array with all the duplicate elements removed. For example, given the input array [5, 3, 7, 5, 8, 3, 1, 3, 1, 8], the required result array is [1, 3, 5, 7, 8].

Please download the ZIP file **cs4345_assign4_2016_todo.zip** from the **Assignment** folder in the IVLE Workbin. It contains Visual Studio 2010 Solution and Project files (can be loaded and converted by VS2012 and later). To build the project, you need to have **CUDA Toolkit 6.5** and the corresponding **CUDA Samples 6.5** already installed on your computer, and the environment variables **`CUDA_PATH`** and **`NVCUDASAMPLES_ROOT`** set to the correct directories. Please read **Readme.txt** to find out the additional information about building the executable file.

The **Thrust** library is already included in the **CUDA Toolkit 6.5**. Thrust is a library of data-parallel algorithm primitives such as parallel prefix-sum ("scan"), parallel sort and parallel reduction. Primitives such as these are important building blocks for a wide variety of data-parallel algorithms. Thrust has a programming interface resembling the C++ Standard Template Library (STL). Please go to the GitHub Thrust project page (http://thrust.github.com) for the usage details.

Your task is to complete the source file **unique.cu** by implementing the two CUDA kernels **`Kernel_MarkUnique()`** and **`Kernel_Scatter()`** and the function **`GPU_Unique()`** according to the following requirements and the **requirements specified in the source file**.

Given an input array, the steps to compute the required result on the GPU are already outlined in the source file **unique.cu**. Therefore, you are advised to read the source file very carefully. Basically, that involves a **stream compaction** operation. Even though the Thrust library already has stream compaction routines, you are not allowed to use them and must write your own using the **scan-and-scatter** approach. You can make use of the parallel sort and parallel scan routines in the Thrust library, and they have been wrapped up nicely in the functions **`GPU_SortIntegerArray()`** and **`GPU_AllPrefixSums()`** for you to use.

Any optimization that you do must be targeted for CUDA hardware with **Compute Capability 1.0 and 1.1**. You can assume that the thread block size to be used is always a multiple of 32.

An executable of the sample completed program, **unique_done.exe**, is provided in the ZIP file. You are welcome to run it to get a sense of the relative performance difference between the CPU and GPU.

GRADING

The maximum marks for this programming assignment is **100**, and it constitutes **7%** of your total marks for CS4345.

Program **correctness** constitutes **90 marks** while **design/style** constitutes **10 marks**. Note that if your program cannot be compiled and linked, you get 0 (zero) mark for program correctness.

Good coding style. Comment your code adequately, use meaningful names for functions and variables, and indent your code properly. You must fill in your **name**, **matriculation number**, and **NUS email address** in the **header comment**.

Correctness. Marks are allocated as follows to the implementation of the following kernels and function:

- **`Kernel_MarkUnique()`** — **30 marks**
- **`Kerbel_Scatter()`** — **20 marks**
- **`GPU_Unique ()`** — **40 marks**

SUBMISSION

For this assignment, you need to **submit only** the completed source file **unique.cu**.

You must put it in a ZIP file and name your ZIP file **<*matric_no.*>.zip**. For example, **A0123456X.zip**. All letters in your matric. number must be capitalized.

Submit your ZIP file to the **Assignment 4 Submission** folder in the IVLE Workbin. Before the submission deadline, you may upload your ZIP file as many times as you want to the correct folder. **We will take only your latest submission.** Once you have uploaded a new version to the folder, you **must delete the old versions**. Note that when your file is uploaded to the Workbin folder, the filename may be automatically appended with a number. This is fine, and there is no need to worry about it.

DEADLINE

Late submissions will NOT be accepted. The submission folder in the IVLE Workbin will automatically close at the deadline.

——— End of Document ———