Florida Institute of Technology
Department of Electrical Engineering & Computer Sciences
ECE5550 – High Performance Computing

Module 7 Lab: Reduction

Due Date: 5-Apr-2025

1. Objective

The objective of the first part is to get you familiar with the parallel reduction algorithm.

Deployment Platform: Google Colab Jupyter Notebook

Environment Setup: See the Mod 7 Lab notebook

These lines are shell commands executed within the Jupyter Notebook using the `!` prefix. They are primarily focused on setting up and verifying the CUDA environment, which is necessary for running code on NVIDIA GPUs.

1. `!nvidia-smi`: This command displays information about the available NVIDIA GPUs on the system, including their model, memory usage, and current processes running on them. This is useful for verifying that your GPU is recognized and accessible.
2. `!nvcc --version`: This command displays the version of the NVIDIA CUDA Compiler (`nvcc`) installed on the system. This is important because CUDA code needs to be compiled with a compatible compiler version for the target GPU architecture.

2. Procedure

Step 1: Copy the lab files to a folder in Google Drive under your student (fit.edu) account.

Step 2: Ensure the file `main.cu` has implemented the following where indicated:

a) Allocate device memory
b) Copy host memory to device
c) Copy results from device to host
d) Free device memory

Step 3: Edit the file `kernel.cu` where indicated to implement the device kernel code for the parallel reduction algorithm, assuming an input array of any size that fits in one kernel. You only have to produce the partial sums of each thread block for this part. We will sum up the partial results on the host.

Step 4: Compile and test your code.

```
!rm -rf *.o

!nvcc -arch=sm_75 -o reduce main.cu


!./reduce               # Uses the default input size
```

```
!./reduce <m>          # Uses an input with size m
```

It is a good idea to test and debug initially with small input dimensions. Your code is expected to work for varying input dimensions – which may or may not be divisible by your block size – so don't forget to pay attention to boundary conditions.

Step 5: Answer the following questions in a new file named `lab6/answers.pdf`:

1. How many times does a single thread block synchronize to reduce its portion of the array to a single value?
2. What is the minimum, maximum, and average number of "real" operations that a thread will perform? "Real" operations are those that directly contribute to the final reduction value.

Submission process
Zip all the files required in the submission into one format (*.zip, *.7z, *.tar) compressed formats.
Name the compressed file using the following naming convention, lastname_assignment_number.zip.
Example "smith_Lab7.zip"
Note: CANVAS will be restricted to accepting these formats **ONLY**.

Files required in submission

File(s)	**Percentage of Total Score**
1. lastname_assignment_number.zip (includes see below) a. main.cu b. kernel.cu c. Mod7-Lab.ipynb (if changed) d. answers.pdf	100%

Your code submission will be graded based on the following criteria.

Score	**70%**	**80%**	**90%**	**>90%**
Description	A "valid attempt" to use and develop concepts that are required for the lab subject.	The submission must complete the minimum required to receive 70%. The submission must be able to be compiled with **NO** errors and produced and executable.	The submission must complete the minimum required to receive 80%. The submission must reproduce the expected output results.	The submission must complete the minimum required to receive 90%. The submission must demonstrate neatness and clarity in the code and proficiency in coding practices.

Florida Institute of Technology
Department of Electrical Engineering & Computer Sciences
ECE5550 – High Performance Computing

***NOTE**: All late submissions will start with a grade equivalent to the lowest grade of on-time submission. Then subject to the same criteria as defined in the grading rubric.

<u>Note:</u> *This is a simple but essential exercise. Please write out the code and do not copy it from other examples or lecture slides. That process is most important.*