

Lab 5

Keith Evan Schubert

October 17, 2018

1 Objective

This lab is divided into two parts. The objective of the first part is to get you familiar with the parallel reduction algorithm. The objective of the second part is to get you familiar with the parallel prefix scan algorithm. Please read Mark Harris's report "Parallel Prefix Sum (Scan) with CUDA" to learn the algorithmic background for the second part.

2 Activity

2.1 Reduction

1. Login to kodiak. `cd` to your `mplabs` directory and type `git pull`.
2. Examine the file `<lab-directory>/5.-reduction/main.cu1` to see new method `cudaMemset`.
3. Edit the file `<lab-directory>/5.-reduction/kernel.cu1` 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.
4. Compile and test your code.

```
cd <lab-directory>
make
nano reduction.sh # add reduction commands per below
~/<lab-directory>/5.1-reduction/reduction # Uses the default input size
~/<lab-directory>/5.1-reduction/reduction <m># Uses an input with size m
qsub -q tardis reduction.sh
```

5. Answer these questions for reduction:
 - (a) How many times does a single thread block synchronize to reduce its portion of the array to a single value?

- (b) 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.

2.2 Prefix Scan

1. Edit the file `<lab-directory>/5.2-prefix-scan/kernel.cu` to implement host and device kernel code for the parallel prefix scan algorithm. The algorithm should be work-efficient. You are expected to support an input array of any size that fits in one kernel and are not allowed to add up partial sums on the CPU. In other words, you must use the hierarchical scan approach.
2. Compile and test your code.

```
cd ../5.2-prefix-scan
make
nano reduction.sh # add prefix scan commands per below
~/<lab-directory>/5.2-prefix-scan/prefix-scan # Uses the default input size
~/<lab-directory>/5.2-prefix-scan/prefix-scan <m># Uses an input with size m
qsub -q tardis prefix-scan.sh
```

3. Answer the following question for prefix scan:
 - (a) Describe how you handled arrays not a power of two in size and all performance-enhancing optimizations you added.

3 Turn in

Upload to the course Canvas site:

1. a report that includes :
 - (a) the output
 - (b) answer section where you answer the three questions above (two in reduction, one in prefix scan).
2. 5.1-reduction/kernel.cu
3. 5.2-prefix-scan/kernel.cu

The cuda code will be graded for completeness, correctness, handling of boundary, and style (5pts). The report will be graded on readability, clarity, analysis, and solution to the questions (5pts).