Assignment 6. List Scan

Minjae Gwon, Department of Computer Science and Engineering.

Problem 1

> How many global memory reads are being performed by your kernel?

- `(numElements) + (2 * numElements)`
  - `(numElements)` reads are being performed by `scan`.
  - `(2 * numElements)` reads are being performed by `merge` to read element and associated prior prefix sum.

Problem 2

> How many global memory writes are being performed by your kernel?

- `(numElements) + (numElements)`
  - `(numElements)` reads are being performed by `scan`.
  - `(numElements)` reads are being performed by `merge`.

Problem 3

> How many times does a single thread block synchronize to reduce its portion of the array to a single value?

- `log(numElements)`
  - As stride doubled, the number of synchronize execution could be `log(numElements)`.

Problem 4

> Suppose that you want to scan using a binary operator that is not commutative. Can you use a parallel scan for that?

- No. A scan should be in a serial manner because calculations should have dependencies between them if a operator is not commutative.

Problem 5

> Is it possible to get different results from running the serial version and parallel version of scan? Explain.

- Yes. Results possible to be different if a floating point precision error exists.

Problem 6

> Your version of `template.cu`.

- Please refer to the attached `template.cu`.

Problem 7

> The result as a table/graph of kernel execution times for different input data, with the system information where you performed your evaluation. Run your implementation with the input generated by the provided dataset generator. For time measurement, use `gpuTKTime_start` and `gpuTKTime_stop` functions (You can find details in `libgputk/README.md`).

- Please refer to the attached `evaluation.pdf`. 'Performing CUDA computation' column expresses the execution times of the kernel. Note that it is generated by `scripts/manage.py`.