Assignment 2. Tiled Matrix Multiplication

Minjae Gwon, Department of Computer Science and Engineering.

Problem 1

> How many floating operations are being performed by your kernel?

- `(2 * numAColumns) * (numCRows * numCColumns)`
  - `(2 * numAColumns)` operations are used to calculate the inner product for each entry.
  - `(numCRows * numCColumns)` represents the number of entries.

Problem 2

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

- `(2 * ceil((float)numAColumns / TILE_WIDTH)) * (numCRows * numCColumns)`
  - Each thread performs `(2 * ceil((float)numAColumns / TILE_WIDTH))` memory reads.
  - `(numCRows * numCColumns)` is the total number of threads.

Problem 3

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

- `numCRows * numCColumns`
  - Only the number of entries in the output matrix.

Problem 4

> Describe what further optimizations can be implemented to your kernel to achieve a performance speedup.

- Adjusting `TILE_WIDTH` to suit matrix size and GPU environment in terms of memory usage and coalescing.

Problem 5

> Your version of template.cu.

Please refer to the attached `template.cu`.

Problem 6

> Execution times of the kernel with the input data generated by the dataset generator (in a table or graph). Please include the system information where you performed your evaluation. 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 `manage.py`.

Problem 7

> Execution times of the kernel for 4096*8000 and 8000*512 input matrices with different tile widths (2, 4, 8, 12, 16, 24, 32). Please include the system information where you performed your evaluation. 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 `manage.py`.