Lab Assignment 2
Due Oct 14, 2023 at 11:59pm

1 Objective

The purpose of this lab is to implement a tiled dense matrix multiplication routine using **shared memory**.

2 Instructions

The code template in `template.cu` provides a starting point and handles the import and export as well as the checking of the solution. Students are expected to insert their code is demarcated with `//@@`. Students are expected to leave the other code unchanged.
Edit the skeleton code to perform the following:

- Allocate device memory
- Copy host memory to device
- Initialize thread block and grid dimensions
- Invoke CUDA kernel
- Copy results from device to host
- Free device memory
- Write the CUDA kernel

Compile the template with the provided `Makefile`. The executable generated as a result of compilation can be run using the following code:

```
./TiledGEMM_Template -e <expected.raw> -i <input1.raw>,<input2.raw>
-o <output.raw> -t matrix
```

where `<expected.raw>` is the expected output, `<input0.raw>,<input1.raw>` is the input dataset, and `<output.raw>` is an optional path to store the results.
`README.md` has details on how to build `libgputk`, `template.cpp` and the dataset generator.

3 What to Turn in

Submit a report that includes the following:

1. How many floating operations are being performed by your kernel?
2. How many global memory reads are being performed by your kernel?
3. How many global memory writes are being performed by your kernel?
4. Describe what further optimizations can be implemented to your kernel to achieve a performance speedup.
5. Your version of `template.cu`.

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`).

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`).