Lab Assignment 7
Due Dec 3, 2023 at 11:59pm

1 Objective

The purpose of this lab is to implement a kernel that performs **sparse matrix-vector multiplication on a transposed JDS (Jagged Diagonal Sparse) format**. Please refer to the lectures notes (slide 40 to 45) on how JDS format with transposition works. You should change the data layout of a sparse matrix in the dense matrix format first to transposed JDS format first on the host, perform matrix-vector multiplication on GPU, then convert the result back to a dense matrix to verify the result on the host.

2 Instructions

Edit the skeleton code to perform the following:

- Allocate device memory
- Copy host memory to device
- Initialize thread block and kernel grid dimensions
- Invoke CUDA kernel
- Copy results from device to host
- Deallocate 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:

```
./JDS_T_Template -e <expected.raw> -i <input0.raw>,<input1.raw> -o <output.raw>
-t vector
```

where `<expected.raw>` is the expected output, `<input0.raw>` and `<input1.raw>` is the input dataset (`input0.raw` is the sparse matrix, and `input1.raw` is the input vector), 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. Two versions of `template.cpp` with and without shared memory staging (bringing data into shared memory for computation and writing back at the end)
2. 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`).