**University of Southern California**

**Department of Electrical Engineering**

**EE557 Fall 2022**

**Instructor: John Paul Walters**

**Project #4, Due: 11:59 PM., Tuesday, November 22th**

**TOTAL SCORE: / 5**

**I. Overview**

The objective of this assignment is for students to gain an understanding of how to bench the performance of GPU programs on real GPUs.

**II. Benchmark Programs: Matrix Multiplication on GPU**

**CUTLASS** is a collection of CUDA C++ template abstractions for implementing high-performance matrix-multiplication (GEMM) and related computations at all levels and scales within CUDA. In this project, we use three different GEMM test bench: **DWMMA**, **SGEMM**, and **DGEMM**. To explore the performance of GPU performing GEMM, we vary the tile size, column/row-major, and matrix size. In this project, we use 24 groups of parameters for **DWMMA**, 3 groups of parameters for **SGEMM**, and 3 groups of parameters for **DGEMM**.

The benchmark programs are given in the material we provided. The script we use to run the test is located at **cutlass-gpgpu-sim/script**. Read the source code in **cutlass-gpgpu-sim/gemm-test/** and answer the following question:

* what is the tile size, matrix size, and column/row-major of the input matrices for each of the test?

You might also find it useful to check out the cutlass-tutorial (**cutlass\_tutorial.pdf** uploaded to DEN) and the source code of cutlass <https://github.com/NVIDIA/cutlass/tree/v1.1.0> .

(Note that the script we provide is different with the script on the github.)

**III. Experimental Setup**

We run the testbenches on real GPU. We use USC CARC's Discovery Cluster to run the experiments. CARC provides a variety type of GPUs. In this project, we use the **NVIDIA Tesla A40** GPU (a40) and **NVIDIA Tesla A100** GPU (a100) on CARC.

* Please refer to

<https://carc.usc.edu/user-information/user-guides/software-and-programming/using-gpus>

for instruction on using GPUs on CARC.

* To submit interactive job on CARC, you can refer to

<https://carc.usc.edu/user-information/user-guides/hpc-basics/running-jobs>

You might want to run interactive section for your convenience. An example command of interactively running the CUTLASS on a100 GPU on CARC is:

* module purge
* module load gcc/8.3.0 cuda/10.1.243
* salloc --partition=gpu --time=4:00:00 --cpus-per-task=8 --gres=gpu:a100:1 --mem=32GB
* cd cutlass-gpgpu-sim/
* ./script

Check the output of the CUTLASS, if the CUTLASS testbench program is successfully executed, then the output file should be written with:

|  |
| --- |
| Successfully Launched  Result Verified |

For running on an a40 GPU, simply use “--gres=gpu:a40:1” in the salloc command. The GPU resources are limited and may need waiting for resource allocation.

Make a screenshot of your successful execution of the CUTLASS and a screenshot of one log file (e.g. log\_dwmma1).

**IV. Performance Evaluation**

We are only interested in the **GFLOPS** (**GFLOPs per second)** metric on the real GPU. We first modify the code to compute the real GFLOPS of the kernel section (excluding the time to allocate the arrays and the overheads). Specifically, you need to modify the **gemm.h** file and output **three** **results**:

* Record and print the execution time *t* of the calculation (in second).
* Calculate and print the number of FLOPs.
* Calculate and print the GFLOPS using FLOPs and *t*.

HINT: there are pre-defined functions to compute the FLOPSs and GFLOPS; see the function **GFLOPs\_per\_sec in cutlass-gpgpu-sim/gemm-test/gemm\_testbed.h**. You may directly use the functions.

Run the same script with the experimental setup to output the results of the same 24+3+3 programs.

Make a screenshot of one log file that shows the above three results (e.g. log\_dwmma1). Compare the performance of the two GPUs a40 and a100, and comment on the influence of the configurations of GEMM to the com

-putation performance.

**V. Submission**

You only need to submit a report of this project. Your pdf report should include the following:

A table in **MS Excel** or Word, that shows:

* The key parameters (tile size, matrix size, A major, B major) of the calculation
* Results for 2 real GPUs in Section IV. (1.5 pts)

A report in **PDF** that includes:

* The screenshots of your terminal in your milestones and your output files (1 pts)
* Your analysis to the results, including the comparison of the two GPU performance and the influence of GEMM configurations to the computation performance. You may use tables and charts to assist your analysis. (1.5 pts)

The **source code** with your modification:

* gemm.h. (for varification)

**Like other assignments, this project must be done INDIVIDUALLY!**

**Similar designs will be securitized.**