CODESIGN

LAB 2 : CUDA Programming

Report Deadline: May 11th 2025.

Part1: (35%)

1- Give the characteristics of the used NVIDIA dedicated GPU : Architecture, local, cache Memories sizes, Number of Streaming MultiProcessors , max number of work items, work groups, etc….

2- Test the 2 kernel invocations (**MatmulXrow** and **MatmulYrow** for N=8192) implemented in kernel.cu cuda file. (the A, B and C matrices contain **Float** elements).

Kernel	**Total time** (Float)
MatmulXrow	
MatmulYrow	

a- Compare the performances of ***MatmulXrow*** and ***MatmulYrow*** and interpret the results.

b- Based on your NVIDIA device characteristics (Architecture, CUDA Cores structure, local memory), do you think that the implementation of the same kernel with integer Matrices (A, B and C) will give the same results? Explain.

3- Change the type of matrices A, B and C to **Int** (integer) and test the 2 kernel implementations, for N=8192. Fill the table with value of "*Total execution Time*" (Data Transfer + Kernel execution).

Kernel	**FLOAT**	**INT(integer)**
MatmulXrow		
MatmulYrow		

c- Compare the performances of the **Float** and **Integer** implementations and interpret the results. Explain if the results correspond to the results announced in **b**.

4- Implement a ***CUDA kernel*** based on the ***Block Tiling*** principle with Block Size= (16*16 and 32*32). Test the Kernel (only for float) and ***compare*** the results with those obtained by the kernel ***MatmulYrow***.

Is the **gain** as significant as expected? (when considering the arithmetic Intensity)

Part2: (45%)

1- Starting from the performances, obtained by the 'Block Tiling' kernel, as a reference value:

- Propose improvements that may increase the matrix multiplication performance.
- For **each** of the proposed improvement:
  - Explain the used technique, how and why it improves the performances. Implement the corresponding kernel in CUDA and give the performance gain obtained (= Proposed technique performance / 'Block Tiling' performance)

Recommended reading:
https://siboehm.com/articles/22/CUDA-MMM "How to Optimize a CUDA Matmul Kernel for cuBLAS-like Performance: a Worklog" , December 2022.

Part3: (20%)

1- Implement the matrix multiplication using the **cuBLAS** Library (Developped by NVIDIA for accelerating HPC application and optimized for NVIDIA architectures).

- Give the Host program and the Kernel.
- Compare the performances with the best implementation you proposed.

Optional: [Bonus Question]

1- Explain the principle and the benefits of the TensorCore used in NVIDIA Streaming Multiprocessors. Implement the matrix using the tensorCore :

- Explain the implementation (Host program and Kernel).
- Compare the performances with those obtained by cuBLAS Library.