# CLE - Assignment 3

Eduardo Santos - 93107
Pedro Bastos - 93150

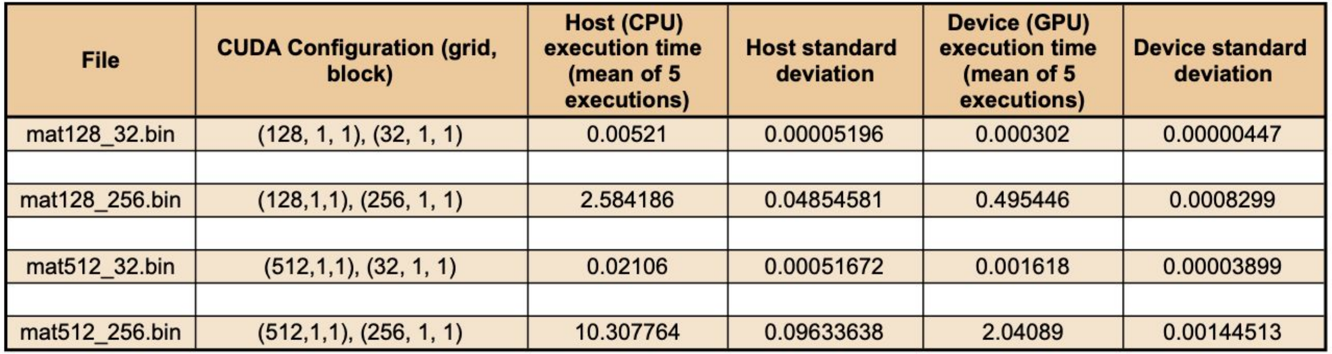
# Program 1 and 2

Both problem’s solution were made using the CUDA API. The implementation for each problem can be described as:

The **main** method contains the main logic of the solution’s implementation, it will process the command line input, setting up the GPU device. It will then, for each file:

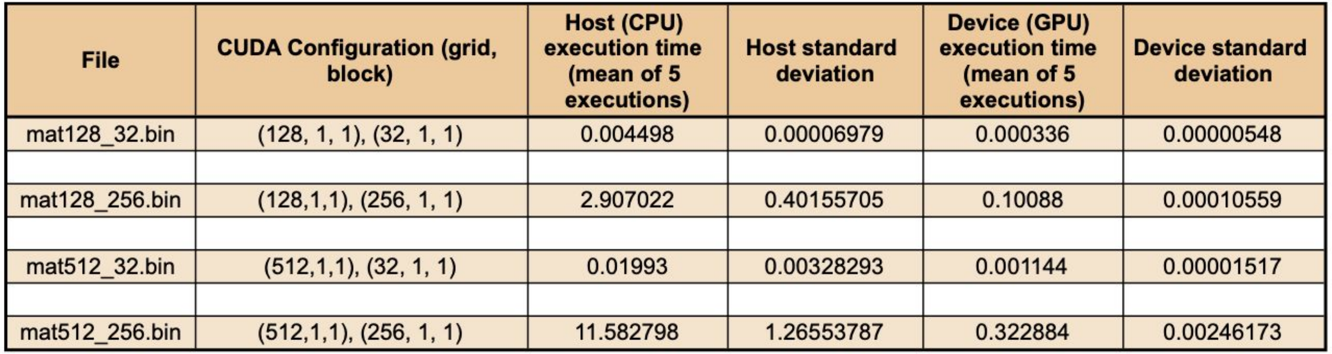
- Allocate memory on the GPU device;
- Copy the file data from CPU memory to GPU memory;
- Call the **computeDeterminantGPU** that will invoke CUDA kernel to perform the computation of the determinant of each matrix from the given matrix array, row by row or column by column, depending on the problem, storing the computed results;
- Copy data back from GPU memory to CPU memory;
- Free the memory allocated on the GPU device;
- Call the **computeDeterminantHost** that will, for each matrix, call the **computeDeterminant** method, which will compute the determinant of a given matrix, row by row or column by column, depending on the problem, also storing the final results;
- Print the results from both the host and the GPU device, for comparison, as well as the time taken by each operation (host and device).

# Execution Times - Program 1



| File | CUDA Configuration (grid, block) | Host (CPU) execution time (mean of 5 executions) | Host standard deviation | Device (GPU) execution time (mean of 5 executions) | Device standard deviation |
|---|---|---|---|---|---|
| mat128_32.bin | (128, 1, 1), (32, 1, 1) | 0.00521 | 0.00005196 | 0.000302 | 0.00000447 |
| | | | | | |
| mat128_256.bin | (128,1,1), (256, 1, 1) | 2.584186 | 0.04854581 | 0.495446 | 0.0008299 |
| | | | | | |
| mat512_32.bin | (512,1,1), (32, 1, 1) | 0.02106 | 0.00051672 | 0.001618 | 0.00003899 |
| | | | | | |
| mat512_256.bin | (512,1,1), (256, 1, 1) | 10.307764 | 0.09633638 | 2.04089 | 0.00144513 |

# Execution Times - Program 2



| File | CUDA Configuration (grid, block) | Host (CPU) execution time (mean of 5 executions) | Host standard deviation | Device (GPU) execution time (mean of 5 executions) | Device standard deviation |
|---|---|---|---|---|---|
| mat128_32.bin | (128, 1, 1), (32, 1, 1) | 0.004498 | 0.00006979 | 0.000336 | 0.00000548 |
| | | | | | |
| mat128_256.bin | (128,1,1), (256, 1, 1) | 2.907022 | 0.40155705 | 0.10088 | 0.00010559 |
| | | | | | |
| mat512_32.bin | (512,1,1), (32, 1, 1) | 0.01993 | 0.00328293 | 0.001144 | 0.00001517 |
| | | | | | |
| mat512_256.bin | (512,1,1), (256, 1, 1) | 11.582798 | 1.26553787 | 0.322884 | 0.00246173 |

# Conclusion

- Looking at the results, we can see that there is a significant difference between the time computing the matrices determinants on the host and on the device (GPU). The time taken by the GPU is much smaller.

- We believe this happens because a CPU core has to handle every single operation a computer does, so it has a huge and complex instruction set. To implement instruction set we need more logic which leads to higher cost compared to a GPU.

- GPU cores have less cache memory and simpler instructions. However, they are optimized to do more calculations as a group. Since the instructions are simpler, less memory is used, making them more efficient.

- To conclude, we believe that it is worthwhile to use a GPU to run this kind of problem.