GPU programming using CUDA

In this assignment, you will get familiar with CUDA programming through some incomplete CUDA programs that you need to complete and analyze.

This assignment is worth 40 points and the submission is due on 11:59 pm ET on Mar 17, 2025.

Your tasks

1) Connect to a GPU server (on CS portal)
    a. ssh <computing-id>@gpusrv<#>.cs.virginia.edu
    b. Check all the GPU servers here
2) Load CUDA toolkit
    a. module load cuda/12.4.0
3) Copy the tar file provided (cuda_assignment.tar.gz), it has:
    a. matadd.cu
    b. matmul.cu
    c. parallelSum.cu
4) You need to add steps in the above programs where there is a "**// TODO : "**
5) Compile the program
    a. Example command: nvcc matmul.cu -o matmul
6) Run the program
    a. ./matmul
    b. Or nvprof ./matmul (This will give you breakdown of time taken by the kernel launched on GPU as well data movement time and time taken by APIs.)

7) For each of the cuda code in the provided tar file (matadd, matmul, parallelSum), do the following:
    1. **[6 points]** Complete the CUDA program
    2. **[3 points]** Add code to the .cu program that runs the kernel on CPU
    3. **[3 points]** Verify the output of CPU and GPU kernel
    4. **[9 points]** Compare the CPU and GPU runtime for the kernel (please mention in ms)
        a. Hint: use cudaEventCreate, cudaEventRecord for GPU
        b. Run the kernels for enough iterations to get steady state time.
        c. (for GPU) Memory access for data transfer is considered as a part of execution. Start the GPU timer before sending the data from CPU to GPU and stop when GPU returns the data to CPU
        d. (for GPU) Use nvprof to report the breakdown for time taken by
            i. Data movement from host to device
            ii. Kernel time

iii. Data movement from device to host
iv. APIs time

5. **[12 points]** Change the size of matrices and vector in the given CUDA program. Report your observation and time (as reported in #4) for at least 4 different sizes (adjust grid size accordingly).
    a. For what input size do you see GPU outperforming CPU?
    b. For what input size do you see Data movement time is less the kernel execution time on GPU?

8) **[1 point]** Explain why the kernel launch is different for matmul.cu/matadd.cu and parallelSum.cu. Why are there three parameters in <<<,,>>> for parallelSum.cu?

9) **[1 point]** Explain the purpose of __syncthreads() call in parallelSum.cu.

10) **[5 points]** Modify matmul.cu (submit as matmul_sharedmem.cu) to utilize shared memory for intermediate results. Update the grid and block dimensions accordingly.

a. Compare the runtime difference of matmul_sharedmem.cu with matmul.cu (without shared memory).
b. Pick any one matrix size and change the shared memory tile size and find an optimum shared memory size.
c. Please also report the GPU time breakdown using nvprof for all cases.

Please ensure your code compiles and add comments in the code wherever necessary. Submit the modified .cu files and a report on canvas.

References:

1. YouTube Playlist: CUDA Programming - Introduction and Tutorials
2. CUDA Tutorial - Tutorial 01
3. NVIDIA Developer Blog - Even Easier Introduction to CUDA
4. NVIDIA CUDA Documentation - Using Separate Compilation in CUDA
5. CUDA Tutorial by LLPanorama
6. CUDA by Example, from NVIDIA
    Textbook
    Code