Assignment - HPC 2

* TITLE: Parallel computing using CUDA.

* PROBLEM STATEMENT: Vector & Matrix operations; Design parallel algorithm to:
1. add two large vectors.
2. multiply vector & matrix
3. multiply two NXN arrays using n^2 process.

* LEARNING OBJECTIVES: learn parallel computing using CUDA, & parallel decompostion of a problem.

* OUTCOME: Decomposed problem into sub-problems, learned how to use GPUs, solved sub problems using threads on GPU cores.

* SOFTWARE & HARDWARE REQUIREMENTS: 64 bit CPU, 4GB RAM, CUDA toolkit, Nvidia GPU, NVCC compiler, Google collab.

* THEORY:

Dividing a computation into smaller computations & assigning them to different processors for parallel execution are 2 key steps in the design of parallel algorithms.

The process of dividing a computation in smaller parts, some or all which may potentially be executed in parallel is called decomposition. Tasks are programmer defined units of computation into which the main computation is subdivided by means of decomposition. Simultaneous execution of multiple tasks is the key reducing time required to solve the entire problem.

1. In addition of 2 vectors, we have to add i^{th} element from first array with i^{th} element of second array to get i^{th} element of resultant array.

Using CUDA, vectors can be added using:
i> n blocks, 1 thread/block ii> 1 block, n threads iii> m blocks, n threads / blocks

2. Similarly, the product of a vector (1xm) & matrix (mxn) will result in a 1xn vector containing the result of multiplication.

3. The product of 2 matrices $(x_1 x n_1)$, $(n_1 x x_2)$ will result in a matrix of dimension of $x_1 x \cdot x_2$.

CUDA Kernel and Threads:

The fundamental part of a CUDA code is the Kernel program. Kernel is the function that can be executed in parallel in the GPU device. A CUDA Kernel is executed by an array of CUDA threads. All threads run the same code. Each thread has an id that it uses to compute memory addresses & make control decisions. CUDA organizes thousands of threads into a hierarchy of a grid of thread blocks.

A grid is a set of thread blocks that can be processed on a device in parallel. A thread block is a set of concurrent threads that can cooperate among themselves through synchronization barriers & access to a shared memory space private to block.

Each thread is given an unique ID within block, each block has a unique ID within grid.

* CONCLUSION:

Successfully implemented and executed vector and matrix operator operations parallely using CUDA.