Experiment no. 04

Title :-

CUDA program for Addition of two large Vectors and Matrix multiplication.

Objectives :-

The objectives is to Understand how to write CUDA program for vector addition and matrix multiplication. We will learn the basic concept of parallel programming and how to use CUDA C to write program that can run on GPUs.

Problem Statement :

We will be performing two operation using CUDA C. The first operation is vector addition, where will be adding two large vector. The Second operation is matrix Multiplication, where we will be multiply two large matrices using CUDAC.

S/W Requirement :-

1) CUDA Toolkit
2) C/C++ compiler
3) Any text editor or IDE

H/w Requirement :

1) A computer with a NVIDIA GPU that Support CUDA Programming
2) 64 bit open Source Linux or window or its derivatives
3) Minimum 4/8 gb RAM.

Theory!

1) CUDA Programming Model:

The CUDA programming model is designed to exploit the parallelism in GPU architectures to accelerate the execution of computation. The CUDA platform provides a programming model that allow developers to write C++ program with additional keyword and construct to express parallelism.

CUDA Kernal are the parallel functions that execute on the GPU. Each kernel is executed by many threads in parallel, Where each thread ~~pr~~ perform the Same computation but on different data elements.

2) Vector Addition :-

Vector addition is the process of adding two or more vectors to obtains a new vector. In CUDA program we can use parallel computing to perform vector addition on large vectors. Each thread in a CUDA Kernel is responsible for adding a single element of the vector. The general algorithm for vector addition is as follow.

1) Initialize two vector, A and B, with random value.
2) Allocate memory on the GPU for vector A & B
3) Copy vector A & B from the ~~GPU~~ CPU to GPU.
4) Launch a Kernal function on the GPU to perform the vector addition
5) Copy the result vector from the GPU to the CPU
6) Free the memory allocated on the GPU.

3) Matrix Multiplication:

Matrix multiplication is the process of multiplying two matrices to obtain a new matrix. In CUDA, we can use parallel computing to perform matrix multiplication on large matrices.

The element of the resulting matrix is calculated by multiplying the corresponding elements of the input matrices. Each thread in a CUDA kernel is responsible for computing a single element of the output matrix. The general algorithm for matrix multiplication is as follow.

1) Initialize two matrices, A and B, with random value.
2) Allocate memory on the GPU for matrices A and B.
3) Copy matrices A and B from the CPU to the GPU
4) Launch a kernel function on the GPU to perform the matrix multiplication
5) Copy the result matrix from the GPU to CPU
6) Free the memory allocated on the GPU.

Conclusion:

We learned how to implement vector addition and matrix ~~mult~~ multiplication using ~~CUDIA~~ CUDA C.