Roll no :- 41310

Page No.:

Date:

* Assignment HPC-2 *

* Title:- Parallel Computing using CUDA.

* Problem Statement:-

Vector and Matrix operations.
Design parallel algorithm to
1. Add two large vectors.
2. Multiply vector and matrix.
3. Multiply two NxN arrays using n² processors.

* Objective:-

1. Learn parallel decomposition of problem.
2. Learn parallel computing using CUDA.

* S/W packages and Hardware Apparatus used:

1. operating system
   64 bit open source Linux or windows.
2. programming language C/C++.
3. NVidia GPU.
4. CUDA API

* OutComes:-

students will be able to decompose problem into subproblems to learn how to use GPU's to learn to solve subproblems using threads on GPU core.

* Theory:-

Dividing computation into smaller computation and assigning them to different processors for parallel execution are the two key steps in the design of parallel algorithm.

The process of dividing computation into smaller parts, some or all of which may potentially be executed in parallel is called decomposition. Tasks are programm defined units of computation into which the main computation is subdivided by means of decomposition.

simultaneous execution of multiple tasks is the key to reducing the time required to solve the entire problem. Tasks can be of arbitary size, but once defined, they are regarded as indivisible units of computation. The tasks into which a problem is decomposed may not all be of the same size

In addition of two vectors, we have to add the i^{th} element from first array with i^{th} element of second array to get i^{th} element of resultant array. we can allocate this each addition to distinct thread. same thing can be done for the product of two vectors.

There can be 3 cases for addition of two vectors using CUDA.

1. n blocks and one thread per block.

2. 1 block and n threads in that block.
3. m blocks and n threads per block.

Mathematical Model

Let S be the system set

$S = \{s; e; x; y; Fme; DD; NDD; FC; SC\}$

s = start state

e = end state

x = set of inputs

$x = \{x_1\}$ where x_1 is element of vector or matrix where

x_1 = Elements of vector

Y = output set (sum of product of element of vector/matrix)

Fme is set of main functions

$Fme = \{f_1, f_2, f_3\}$

where

f_1 = decomposition function.

f_2 = function to find sum/product

f_3 = function to merge results.

DD = Deterministic Data.

Vector/Matrix of Elements.

NDD = Non Deterministic Data

FC = Failure Case.

No failure Case identified for this application.

Test Cases:-

① vector addition
for n=8

6 4 2 9 1 3 2 1
7 3 7 1 6 7 5 4

Result:- 18 7 9 10 7 10 7 5

② vector Matrix multiplication.
Vector
4 3 2 4
Matrix

2 4 3 1 2
2 3 4 3 4
4 3 1 3 1
1 4 1 2 2

Result:-
26 47 30 35 30

③ Matrix to Matrix Multiplication

Matrix 1
4 7
8 6

Matrix 2
4 6
7 3

Result.
65 45
74 66

Conclusion:-

In this way, vector addition, vector matrix multiplication & matrix-matrix multiplication is performed with less time complexity using GPU & programming