41145

Page No.
Date

LP1 - HPC1

* Title: Parallel Reduction

* Problem statement :
a) Implement parallel reduction using min, max, sum and avg operations.
b) Write a CUDA program, that, given an N-Element vector, find -
Max. element, min element, and arithmetic mean of the vector., SD of the values.
Test for input N and generate a randomized vector V of length N. The program should generate output as the 2 computed max. values as well as the time taken to find each value.

* Objectives:
Understand data parallel model of computation.

* Theory:
CUDA is a parallel computing platform and programming model that enables dramatic increase in the computing performance by harnessing the power of the graphics processing
Since its introduction in 2006, CUDA has been widely deployed through many applications and published research papers and supported by an installed base of many users.

Applications used in physics, chemistry, biology, data mining, astronomy and other computation intense fields are drastically increasing using CUDA. It is delivering ~~bef~~ benefits of GPU acceleration.

CUDA C extends C by allowing programmers to define C functions called C kernels that, when called, are executed N times in parallel by different CUDA threads as opposed to only 1 like the regular C.

A kernel is defined using the _global_ declaration specifier and the numbers of CUDA threads for a given kernel, call is specified using a new <<< ... >>> execution configuration syntax.
Each thread that executes the kernel is given a unique thread ID that is accessible within the kernel.

* Parallel Reduction:
Reduce is a collective communication primitive used in the context of parallel programming model to combine multiple vectors into one, using an associative binary operator.

Every vector is present in a distant processor in the begin. The goal of the primitive is to apply the operator in the order given by the processor.

41145

Page No.
Date

* Algorithm:

① Read size of vector N and read the numbers
② Record the start time.
③ Using kernel <<< >>> function in CUDA, transfer data to a device, parallelize your code for the given size of vector.
Define size of the grid block and thread to find the result.

④ Read the end time.
⑤ Calculate the execution time = (End time - Start time) and display it.
⑥ Apply Repeat this procedure for various sizes of the vector and compare execution with serial program.

* Conclusion:

From this assignment, I was able to understand the basics of parallel reduction and hence implement this assignment.