* Assignment HPC1 *

Roll no. 41310

Page No.:
Date:
YOUVA

* Title:- Parallel Computing using CUDA.

* Problem statement:-

(1) Implement parallel reduction using Min, Max, sum and average operations.
(2) Write Cuda program that given an N element vector, find
  (1) Maximum element in the vector
  (2) Minimum element in the vector
  (3) Arithmetic mean of the vector
  (4) Standard deviation of values in vector

* Objective:-
  (1) Learn parallel decomposition of problem
(2) Learn parallel Computing Using CUDA.

* Outcomes:-
  Students will be able to decompose problem into sub problems, to learn how to use GPU's to learn to solve subproblem using threads on GPU Cores.

* Software and Hardware requirements:-

Operating system:- 64bit open source linux or its derivative.

Programming language:- C/C++ Nvidea GPU/ Google Colab.

Page No.:
Date:

Theory:- Dividing a Computation into smaller Computations and assigning them to different processors for parallel execution are the two key steps in the design of parallel algorithm.

The process of dividing a Computation into smaller parts, some or all of 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 to reducing the time required to solve 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.

* Important functions & Concepts.

① CudaMalloc:-
MalloC is used to dynamically allocate memory on CPU but to operate on gpu whole data must be global memory.

② CudaMemcpy:-
It Copies source to distination that host to device for operation.

③ Cudafree (arr):
Free the resources allocated.

④ minimum <<< 1, element#/2 >>> (arr)
first argument is ~~distance~~ blocks of thread and second is no. of threads in block.

⑤ --global--
Indicates that function run on device called from host code

⑥ tid = threadIdx.x
It consist of thread index within a block.

⑦ no. of thread = blockdim.x
It contains dimension of block.

Algorithm steps:-

① Define the no. of elements i.e. n.
② Define size for array and allocate memory.
③ Insert random elements in an array.
④ Copy source to destination that host to device for operation.
⑤ Find minimum element in array by comparing 2 elements in one thread
⑥ Again Copy result from device to host for printing result.
⑦ same find maximum by comparing two in decreasing order and returning 1st max element.
⑧ Find sum of array element store it in

first element of array.
(9) Find avg of array by sum in
(10) Convert int array to float and Calculate mean for variance and print it.
(11) Square root of variance is then Calculated and printed as standard deviation.
(12) free the allocated resource.

Test Case:-

Sr. No.	operation	Expected output	Actual output	Result
1.	Min of array	0	0	Pass.
2.	Max element in array	9	9	pass.
3.	Sum of element	4643	4643	Pass.
4.	Avg. of element	4.53	4.53	Pass.
5.	standard deviation	2.89	2.89	Pass.
6.	Variance	8.48	8.48	Pass.
	Considered n = 1024 and array as n%10			

Page No.:
Date:

Conclusion:-

Decomposed problem to subproblems to handle large data using SPU and performed min, max, sum, avg, standard deviation and variance on array elements.