Assignment : HPC 1

Title :- Parallel Reduction using CUDA.

Problem Statement :-

a) Implement parallel reduction using min, max, Sum and average operations.

b) Write a cuda perform that given N numbers vector find:

- Maximum element in vector
- Minimum element in vector
- Arithmetic mean of vector.
- Standard deviation.

Test for input N and generate a randomized vector of length N. The program should operate output as the two computed maximum values as well as the time taken to find each value.

Objectives:

- To learn parallel programming concepts.
- To learn parallel computing using CUDA.

Requirements :-

OS: Fedora / windows (64 bit)
Nvidia GPU or google colab.
CUDA API

Teacher Signature

Mathematical Model

Let S be the System Set.

$S = \{ s, E, x, y, F_{me}, DD, NDD, F_c, S_c \}$

Where,

s = start state

E = End state

x = set of inputs

y = output set { min, max, avg. std, dev }

DD = Deterministic Data

F_c = Failure case

f_{me} = set of functions = $\{ f_1, f_2, f_3, f_4 \}$

Theory:

CUDA :-

- It is parallel computing platform and API model. model created by NVIDIA
- It enabler programmers to use CUDA embbed GPU for general purpose processing
- The CUDA platform is software layer that gives direct access to GPU, virtual ~~mi~~ instruction set an parallel computational elements for the execution of computer kernels.
- CUDA 8.0 comes with following libraries.
  - CUDART : CUDA Runtime library
  - CUBLAS : CUDA Basic linear algebra
  - CUFFT : CUDA fast Fourier Transform library

Teacher Signature _____

CUDA Programming:-
- NVCC compiler is wed for compilation. It separats both host code and device code (GPU) in compilation phase
- Source code file for CUDA har .cu extention.

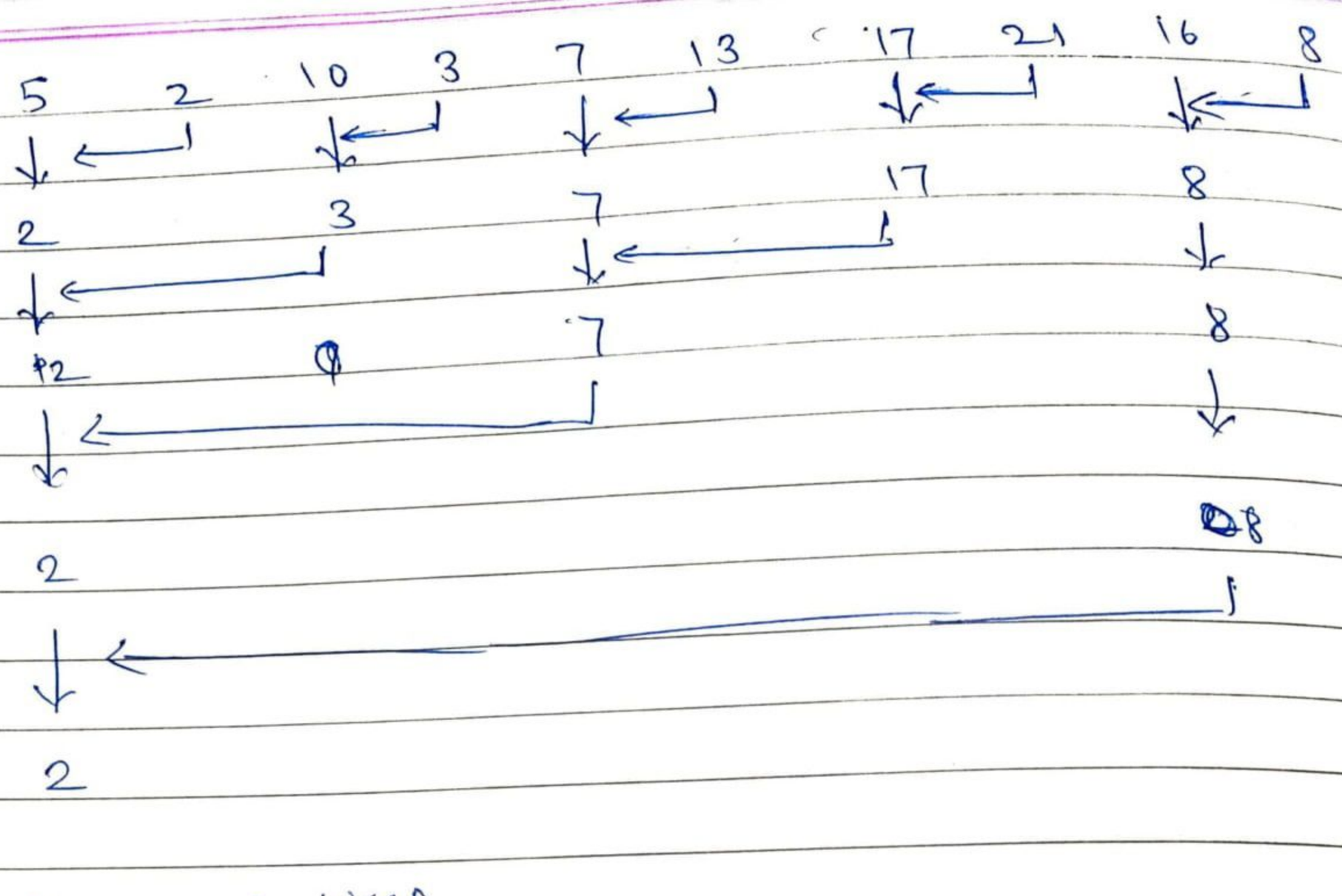
CUDA program structure.
1) Allocate GPU memories.
2) Copy data from GPU to GPU memory
3) Inviee the CUDA kernel
4) Copy data back from GPU to GPU memory
5) free the ~~gpu~~ GPU memories.

Running cuda program on remote machine:
i) open kernal
2) get the login to remote system which har CUDA GPU.
   eg. student @ 10.10.15.21
3) Create a CUDA file with nvcc compiler.
4) It will create an execute file: a.out =)

* parallel reduction.

Suppose we have array with 10 elements
- Decompose this array into subgroups of 2elements
- find min. from each subgroup parallely.
  - Repeate the process.

Teacher Signature



Efficiency = WCSA / WCPA.

Here, we observe that as size of input increase parallel algorithm give better performance than serial algorithms.

Input :- No of elements = 180
output :- Maximum : 19936
Minimum : 180
Sum : 633114
Average : 6331

Conclusion :-

Thus, we successfully executed the parallel reduction using CUDA.

Teacher Signature