Camlin Page

Assignment 1.

- Title: Parellel Reduction using CUDA.

- Problem Statement:

  (a) Implement parellel reduction using min, max, sum & average operⁿ

  (b) Write CUDA program that, given N-number vector find
  - Max element in vector
  - Min element in vector
  - Arithmatic mean
  - Standard deviation

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

- Objectives:
  - To learn parellel programming concep
  - To learn parellel computing in CUDA

- Outcomes:

  We will able to learn parellel computing concepts using CUDA.

- Requirements:
  - OS : Fedora 20/ ubuntu 64-bit
  - Nvidia GPU (Geforce 920M)
  - CUDA API with C/C++ (NVCC compiler)

- Mathematical model:

Let S be the system set

$$S = \{ S', e, x, y, Fme, DD, NDD, Fc, Sc \}$$

where

S' = start state

e = end state

x = set of inputs

Y = output set = $\{min, max, avg, \sigma\}$

DD = deterministic data

NDD = non-deterministic data

Fc = failure case

Fme = set of functions = $\{F_1, f_2, F_3, f_4\}$

- Theory:

CUDA:

- CUDA is a parellel computing platfrom and API model created by NVIDIA
- It enables programmers to use a CUDA-enabled GPU for general purpose processing
- The CUDA platform is a software layer that gives direct access to the GPU's virtual instructions set, and parellel computational elements for the execution of complete kernel.

- Cuda was initially released in 2007 by NVIDIA Corporation
- Cuda 8.0 comes with foll libraries
  - CUDART → Cuda Runtime library
  - CUDLAS :→ Cuda Basic Linear Algebra Subroutines library
  - CUDFFT → CUDA Fast Fourier Transform

→ CUDA programming:

- nvcc compiler is used for compilation It seperates both the host code & device code in compilation phase
- source code file for CUDA has .cu extension

→ CUDA program structure:

1. Allocate GPU memories
2. Copy data from CPU memory to GPU memory
3. Invoke CUDA kernel
4. Copy data back from GPU to CPU
5. Destroy GPU memories

→ How to run CUDA program on remote m/c.

1. Open terminal
2. Get login to remote sys which has CUDA & GPU

eg: student @ 10.10.15.21

3. create a CUDA file with .cu extension & write code in it

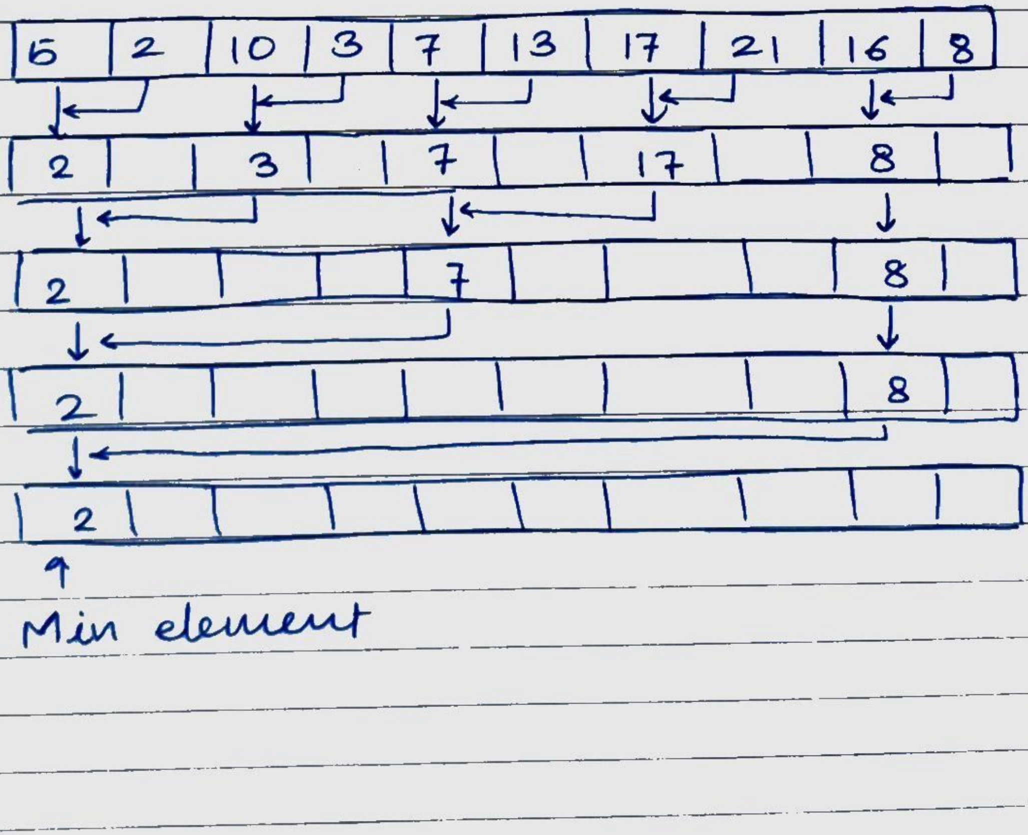
4. Compile CUDA program with nvcc

5. It will create an executable file a.out. Run it

- Parellel Reduction:

Suppose we have an array with 10 elements

- Decompose array into subgrps of 2 elements
- Find min from each subgrp parellely
- Repeat this process.



Min element

- Test cases and Analysis:

Function	I/p size	sequential time	Parellel time	Efficiency
Average	n = 128	0.136	0.129	1.054
	n = 256	0.142	0.123	1.154
	n = 512	0.138	0.120	1.15
Max	n = 128	0.02	0.142	0.014
	n = 1024	0.137	0.113	1.212
	n = 2048	0.135	0.128	1.05
Min	n = 64	0.02	0.189	0.010
	n = 1024	0.134	0.114	1.175
	n = 2048	0.131	0.133	0.98
Standard Deviation	n = 64	0.02	0.05	0.4
	n = 256	0.133	0.175	0.76
	n = 1024	0.133	0.158	0.841
Sum	n = 512	0.01	0.181	0.055
	n = 1024	0.01	0.113	0.088
	n = 2048	0.01	0.126	0.079

$$\text{Efficiency} - \frac{tCSA}{tCPA}$$

Here we observe that as the size of input increases, parellel algorithm gives better performance than sequential algorithm.

Input:

Size of array = 8

Array = 4, 0, 4, 4, 3, 0, 6, 1

Output:

Average = 2.75000
Min = 0
Max = 6
Sum = 22
Standard deviation = 2.0463

- Conclusion:

Thus we successfully executed parellel reduction using CUDA.