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 program that given N-number vector find:
- maximum element in vector
- minimum element in vector
- arithmetic mean of vector
- standard deviation of values in the vector

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

Objectives

To learn parallel computing concepts

To learn parallel computing using CUDA

Outcomes

To be able to learn, understand and implement parallel computing concepts using CUDA

Software and hardware requirements

OS: Fedora 20 / Windows 10 (64-bit)

CUDA API with C/C++

NVIDIA GPU / Google Colab

Theory related concepts

CUDA

It is a parallel computing platform and API model created by NVIDIA. It enables programming to use CUDA enabled GPU for general purpose processing. The CUDA platform is a software layer that gives direct access to the GPU's virtual instruction set and parallel computational elements, for the execution of compute kernels.

CUDA 8.0 comes with the following libraries

CUDART : CUDA runtime library

CUBLAS : CUDA basic linear algebra

CUDPIT : CUDA Fast Fourier Transform library

CUDA programming

NVCC compiler is used for compilation. It separates both host code and device code (GPU) in compilation phase. Source code file for CUDA has .cu extension

CUDA Program structure

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

Running CUDA programs in remote machines

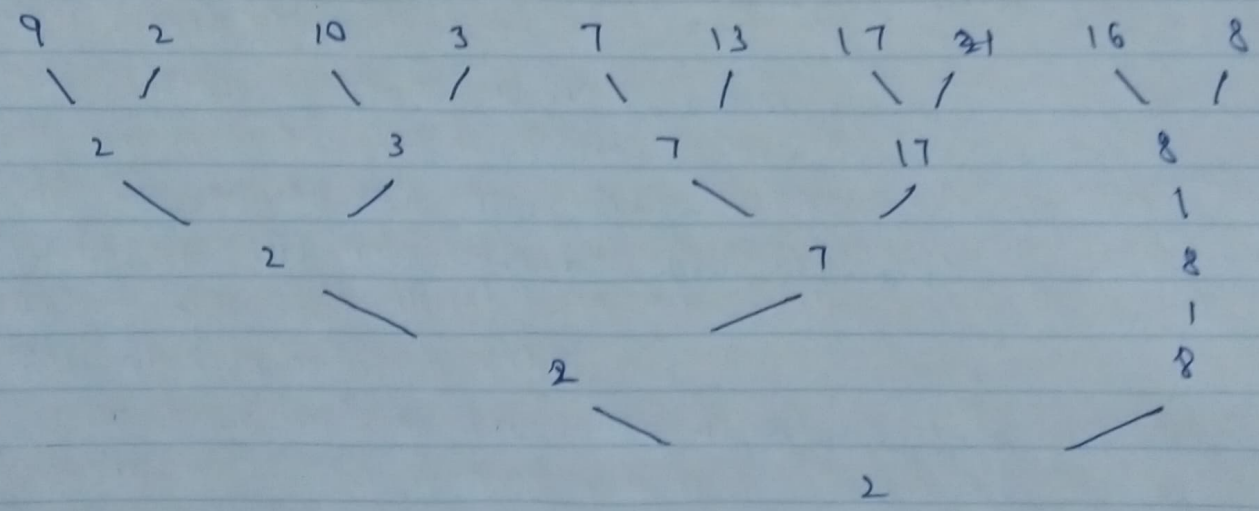
1. Open terminal
2. Get the login to remote system which has CUDA and GPU
   eg: Student@10.10.15.21

3. Create a CUDA file with NVCC compiler
4. It will create an executable file a.out → run it

Parallel Reduction

Suppose we have an array with 10 elements.

- Decompose this array into subgroups of 2 elements
- Find min from each subgroup parallely.
- Repeat this process



Test cases

Function	Input size	Sequential time	Parallel time	Efficiency
max	n = 1024	0.01232 ms	0.1654 ms	0.7447
min	n = 65536	0.138464 ms	0.27584 ms	
Standard deviation variation	n = 16384	0.260064	0.33856	7.681475
Avg / mean	n = 1024	0.011488	0.080720	0.373958

Conclusion

Thus, we successfully implemented CUDA program to find max, min, mean and standard deviation of a vector of N elements. Analysed the speedup for parallel vs serial programs. The speedup increases as the array size increases.