# ASSIGNMENT- A1

- Aim - Parallel computing using CUDA
- Problem Statement -
  a. Implement parallel reduction using Min, Max, Sum & average operations.
  b. Write a CUDA program that, given an N-element vector. Find -
    - The maximum element in the vector
    - The minimum element in the vector
    - The arithmatic mean of the vector
    - The standard deviation of the values in the vector. Test for N and generate a randomized vector V of length N. The program should generate output as the two computed maximum values as well as the time taken to find each value.
- Learning Objectives -
  - Learn parallel decomposition of problem
  - Learn parallel computing using CUDA
- Learning outcomes -
  We will be able to decompose problem into sub problem to learn how to use GPUs, to learn to solve sub problem using threads on GPU cores
- Requirements -
  64 bit OS linux
  NVidia GPU
  CUDA API

Page No.
Date: / /

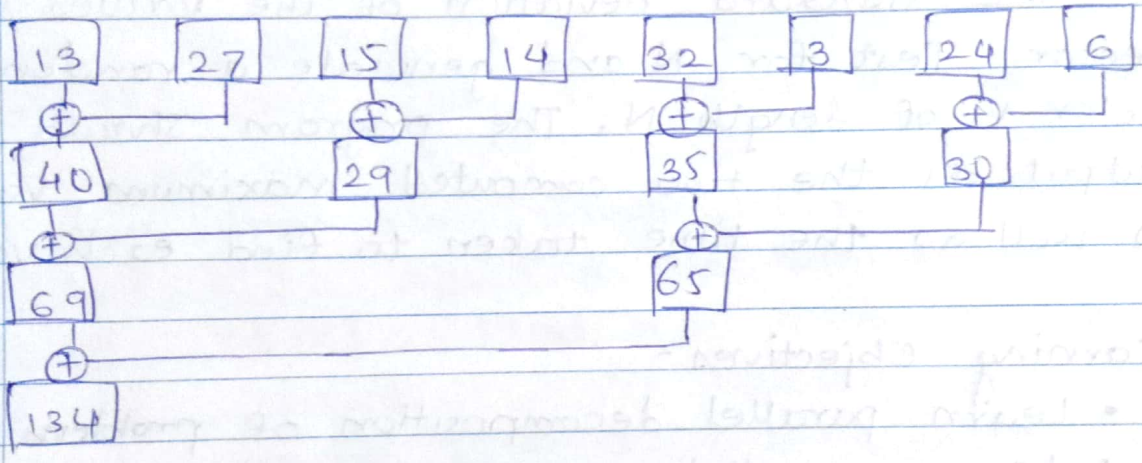
- Theory -

Parallel Reduction -

Implementation of parallel Reduction in CUDA, Reduction operations are those that reduce a collection of values to a single value. Operations which are associative & commentative can be reduction operations.

Finding maximum minimum amongst a given set of number sequential computations complexity can be O(logn).

- Operation for sum of the elements in vector -




CUDA -

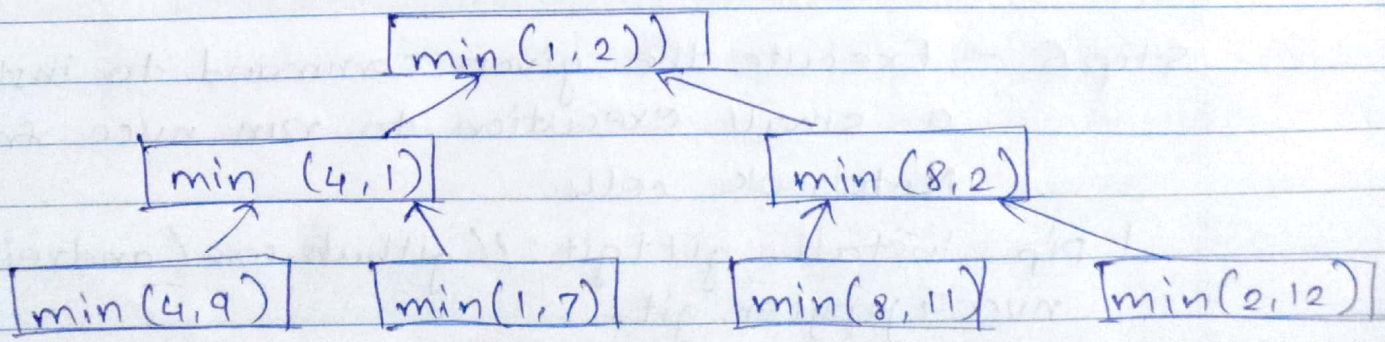
CUDA ( Compute Unified Device Architecture ) is a parallel computing platform & application programming interface model created by NVIDIA

The CUDA platform is also a software layer that gives ~~in~~direct access to the GPU's virtual instruction set & parallel computational elements for the execution of compute kernels.

The CUDA platform is designed to work with programming languages such as C, C++ & fortan.

Dividing a computation into smaller computation & assigning them to different processors for parallel execution are the two keysteps in the design of parallel algorithms. The process of dividing the computation into smaller parts, some or all of which may potentially may be executed in parallel is called decomposition

consider an array {4, 9, 1, 7, 8, 11, 2, 12}. Divide this array into subgroups such as shown in fig. so we {4, 9}, {1, 7}, {8, 11}, {2, 12} Find min. from each group so we get - {4, 1, 8, 2}. Again divide into subgroup - {4, 1}, {8, 2}. Find min from each group. {1, 2}. divide - {1}, {2}
So min among 1 & 2 is 1
Hence, 1 is minimum among all the elements of array.




Similarly, we can find minimum from elements in array, also the sum with same procedure.

For average, calculate sum by recursion & then divide it by number of elements.
Standard deviation is -

$$\sigma = \sqrt{\frac{\Sigma (x - \bar{x})^2}{n}}$$ where $\bar{x}$ is median.

- How to run CUDA program -
  using google colab -

Step 1 - Go to https://colab.research.google.com & click on New Python3 Notebook.

Step 2 - Click to Runtime > change > Hardware Accelerator GPU

Step 3 - Refresh the cloud instance of CUDA on server.

Step 4 - Install CUDA version 9.9

Step 5 - Check the version of CUDA by running the command below -
!nvcc --version.

Step 6 - Execute the given command to install a small execution to run nvcc from Notebook cells.
! pip install git+git://github.com/andreinechaev/nvcc4jupyter.git.

Step 7 - %load_ext nvcc_plugin.
load the ~~execut~~ extension using abv command.

Test cases -

Consider an array of size NCN should be larger value)

11920 6253 11528 4666 8552 1190 1395

31 19949 ------ 19311 625 5903

| | Description | Expected | Actual | Result |
|---|---|---|---|---|
| 1. | Min element | 31 | 31 | Pass |
| 2. | Max element | 19949 | 19949 | Pass |
| 3. | Sum of elements | 4976172 | 4976172 | Pass |
| 4. | Average of elements | 4 | 4 | Pass |
| 5. | Standard deviation | 11286.6 | 11286.6 | Pass |

- Conclusion -

We implemented the given problem statement using parallel Reduction techniques & used Google colab to execute CUDA programs.