Assignment 1

Nachiket Erlekar,41434

August 14, 2020

1 Title

Parallel Computing Using CUDA

2 Objective

- Learn parallel decomposition of problem.
- Learn parallel computing using CUDA.

3 Problem Statement

a) Implement Parallel Reduction using Min, Max, Sum and 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 arithmetic mean of the vector
- The standard deviation of the values in the vector

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

4 Outcome

We will have designed a parallel algorithm using CUDA to calculate Sum, Minimum, Maximum, Mean ans Standard Deviation of a vector.

5 Software and Hardware requirements

1. Operating System : 64-bit Linux or its derivative
2. Programming Language: C/C++
3. Nvidia GeForce 920MX ,2GB
4. CUDA API >= 9.0

6 Date of Completion

August 14, 2020

7 Assessment grade/marks and assessor's sign

8 Theory- Concept in brief

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 algorithms. 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 the entire problem. Tasks can be of arbitrary 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.

8.1 Sum of elements of vector

We can find sum of elements of vector iteratively by stepping through the array and adding each element into global variable. In the end, the global variable will have the sum of vector

To do this operation parallely, we will divide entire vector into blocks, ans assign each block to a core of GPU. Each core will calculate the sum of that block and return it into shared memory. We will then recursively call add on this shared memory dividing the shared memory array into block and combining their results. In the end the shared memory will have sum as it's only element.

8.2 Minimum and Maximum

We can compare each element with current minimum/maximum . As we step through the vector, the current minimum/maximum will be updated accordingly until we reach the end. Then the current minimum/maximum will be the global minimum/maximum

Doing this parallaly, we will calculate the minimum/maximum if each block of vector. We will then recursively combine this block and find minimum and maximum out of these block.

8.3 Mean

: After we find the sum the mean is easily calculated as

$$Mean = \frac{sum}{N} \tag{1}$$

8.4 Variance and Standard Deviation

The formula for variance is:

$$VAR = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2 \tag{2}$$

The formula for standard deviation is basically just square root of variance:

$$STD\_DEV = \sqrt{VAR} \tag{3}$$

9 Algorithm design

This particular code will find the sum by reducing the shared memory. We can use the same approach to calculate minimum,maximum,mean and standard deviation.

```
__global__ void vector_add(LLI *dev_out, LLI *dev_a, int n)
    extern __shared__ LLI sdata[];
unsigned int i = threadIdx.x + blockIdx.x * blockDim.x;
unsigned int tid = threadIdx.x;
sdata[tid] = 0;
while (i < n) {
sdata[tid] + = dev_a[i];
i + = blockDim.x ∗ gridDim.x;
}

    __syncthreads();

    for (unsigned int s = blockDim.x / 2; s > 0; s / = 2)
{
if (tid < s)
sdata[tid] = sdata[tid] + sdata[tid + s];

__syncthreads();
}
if (tid == 0) {
dev_out[blockIdx.x] = sdata[0];}}
```

10 Test cases

83 86 77 15 93 35 86 92
No of threads = 4No of threads = 4No of threads = 4No of threads = 4
The minimum element is 15
The maximum element is 93
The sum is 567
The mean is 70.875
The variance is 748.359
The standard deviation is 27.3562

11 Conclusion

We have successfully conducted a job of finding sum,minimum,maximum, mean and standard deviation of a given vector using parallel computing through CUDA.