Paleti Krishnasai
CED18I039

HPC LAB - Vector Addition CUDA

Name: Paleti Krishnasai
Roll No: CED18I039
Programming Environment: CUDA (collab)
Problem: Vector Addition

Hardware Configuration:
PU NAME: Intel(R) Core(TM) i5-8300H CPU @ 2.30GHz
Number of Sockets: 1
Cores per Socket: 4
Threads per core: 2
L1d cache: 128 KiB
L1i cache: 128 KiB
L2 cache: 1 MiB
L3 cache: 8 MiB

```
paleti@paleti-Lenovo-ideapad-330-15ICH:~$ likwid-topology
--------------------------------------------------------------------------------
CPU name:       Intel(R) Core(TM) i5-8300H CPU @ 2.30GHz
CPU type:       Intel Coffeelake processor
CPU stepping:   10
********************************************************************************
Hardware Thread Topology
********************************************************************************
Sockets:                1
Cores per socket:       4
Threads per core:       2
--------------------------------------------------------------------------------
HWThread        Thread          Core            Socket          Available
0               0               0               0               *
1               0               1               0               *
2               0               2               0               *
3               0               3               0               *
4               1               0               0               *
5               1               1               0               *
6               1               2               0               *
7               1               3               0               *
--------------------------------------------------------------------------------
Socket 0:               ( 0 4 1 5 2 6 3 7 )
--------------------------------------------------------------------------------
********************************************************************************
Cache Topology
********************************************************************************
Level:                  1
Size:                   32 kB
Cache groups:           ( 0 4 ) ( 1 5 ) ( 2 6 ) ( 3 7 )
--------------------------------------------------------------------------------
Level:                  2
Size:                   256 kB
Cache groups:           ( 0 4 ) ( 1 5 ) ( 2 6 ) ( 3 7 )
--------------------------------------------------------------------------------
Level:                  3
Size:                   8 MB
Cache groups:           ( 0 4 1 5 2 6 3 7 )
--------------------------------------------------------------------------------
********************************************************************************
NUMA Topology
********************************************************************************
NUMA domains:           1
--------------------------------------------------------------------------------
Domain:                 0
Processors:             ( 0 1 2 3 4 5 6 7 )
Distances:              10
Free memory:            3546.2 MB
Total memory:           7831.84 MB
--------------------------------------------------------------------------------
```

```
********************************************************************************
Graphical Topology
********************************************************************************
Socket 0:
+---------------------------------------------+
| +--------+ +--------+ +--------+ +--------+ |
| |  0 4   | |  1 5   | |  2 6   | |  3 7   | |
| +--------+ +--------+ +--------+ +--------+ |
| +--------+ +--------+ +--------+ +--------+ |
| |  32 kB | |  32 kB | |  32 kB | |  32 kB | |
| +--------+ +--------+ +--------+ +--------+ |
| +--------+ +--------+ +--------+ +--------+ |
| | 256 kB | | 256 kB | | 256 kB | | 256 kB | |
| +--------+ +--------+ +--------+ +--------+ |
| +-----------------------------------------+ |
| |                  8 MB                   | |
| +-----------------------------------------+ |
+---------------------------------------------+
```

Paleti Krishnasai
CED18I039

CUDA code:

```
%%cu
#include <stdio.h>
#include <stdlib.h>
#include<stdlib.h>
#include<time.h>
#include<math.h>
#define n_size 100000
__global__ void add(float *a, float *b, float *c) {
int index=threadIdx.x+blockIdx.x*blockDim.x;
c[index]=a[index]+b[index];
}
void random_init(float a[],int ch)
{
  srand(time(NULL));
  if(ch==0)
  {
      for(int i=0;i<n_size;i++)
      {
          a[i]=((float)rand()/(float)(RAND_MAX)) * 5.0;
      }
  }
  else
  {
      for(int i=0;i<n_size;i++)
      {
            a[i]=(i+1);
      }
  }
 }
int main() {
 float a[n_size], b[n_size],c[n_size];
cudaEvent_t start, end;
// host copies of variables a, b & c
float *d_a, *d_b, *d_c;
// device copies of variables a, b & c
int size = n_size*sizeof(float);
// Allocate space for device copies of a, b, c
cudaMalloc((void **)&d_a, size);
```

Paleti Krishnasai
CED18I039

```
cudaMalloc((void **)&d_b, size);
cudaMalloc((void **)&d_c, size);
// Create Event for time
cudaEventCreate(&start);
cudaEventCreate(&end);
// Setup input values
random_init(a,0);
random_init(b,0);
// Copy inputs to device
cudaMemcpy(d_a, &a, size, cudaMemcpyHostToDevice);
cudaMemcpy(d_b, &b, size, cudaMemcpyHostToDevice);
int Thread[]={1,2,4,6,8,10,12,16,20,32,64,128,150};
int thread_arr_size=13;
for(int i=0;i<thread_arr_size;i++)
{
	int Threads=Thread[i];
	cudaEventRecord(start);
	// Launch add() kernel on GPU
	add<<<n_size/Threads,Threads>>>(d_a, d_b, d_c);
	cudaEventRecord(end);
	cudaEventSynchronize(end);
	float time = 0;
	cudaEventElapsedTime(&time, start, end);
	// Copy result back to host
	cudaError err = cudaMemcpy(&c, d_c, size, cudaMemcpyDeviceToHost);
	if(err!=cudaSuccess) {
	    printf("CUDA error copying to Host: %s\n",
cudaGetErrorString(err));
	}
	    int flag=0;
	for(int i=0;i<n_size;i++)
	{
	    if(c[i]!=(a[i]+b[i]))
	    {
	        flag=1;
	    break;
	    }
	}
	if(flag==0)
	{
```

```
            printf("Program Executed as Expected\n");
            printf("Time Taken by the program for %d
Threads=%f\n",Threads,time);
        }
        else
        {
            printf("Vector Addition hasnt been done properly,Mismatch in
Values!!!\n");
        }

}
// Cleanup
cudaFree(d_a);
cudaFree(d_b);
cudaFree(d_c);
return 0;
}
```

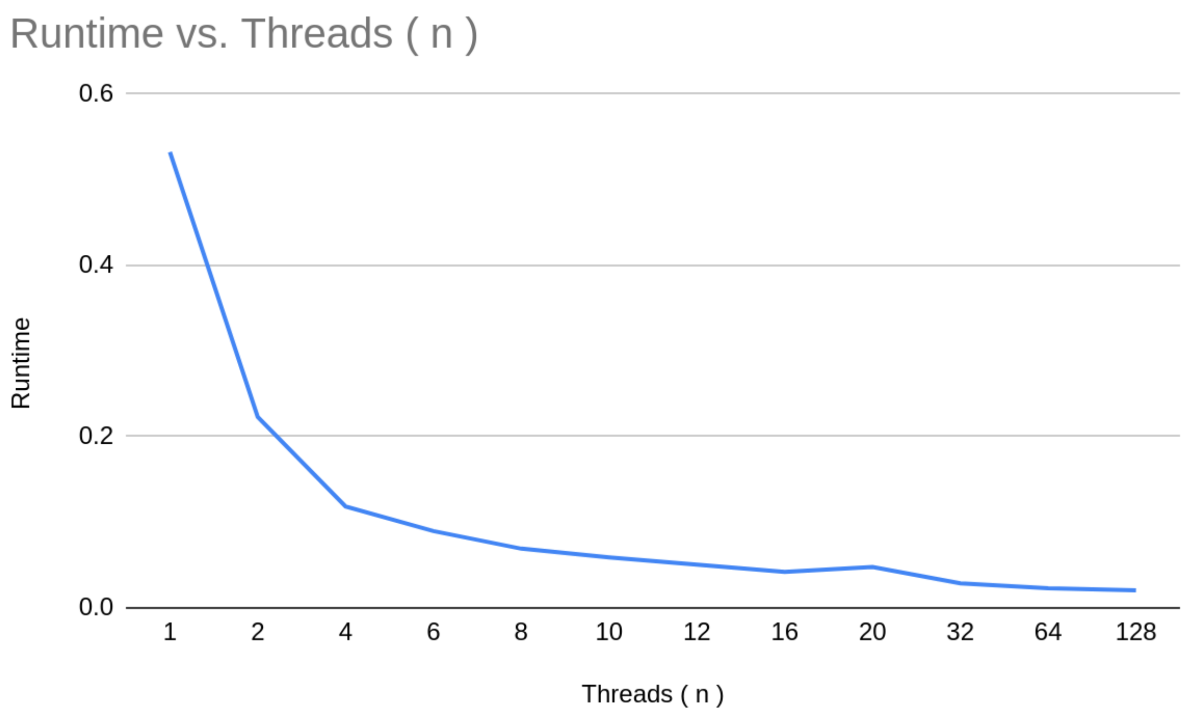
Paleti Krishnasai
CED18I039

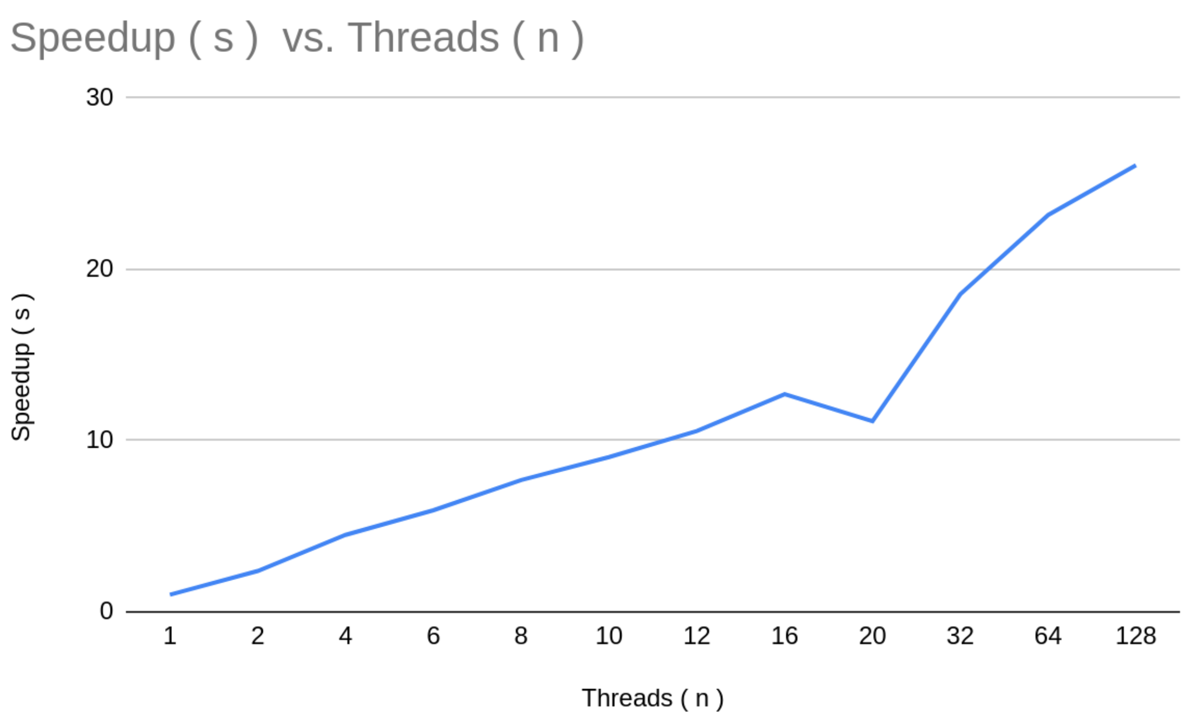
Observations:
The function add() is executed for an array of size 'N' on GPU for "N/T" times for 'T' Threads.

Blocks	Threads (n)	Runtime	Speedup (s)	Parallelization Fraction
N	1	0.532512	1	
N/2	2	0.22304	2.387517934	1.162309957
N/4	4	0.118464	4.495137763	1.036716543
N/6	6	0.089728	5.934736091	0.9978006129
N/8	8	0.069152	7.700601573	0.9944457321
N/10	10	0.058912	9.03910918	0.9881884769
N/12	12	0.050432	10.55901015	0.9875936214
N/16	16	0.041888	12.71275783	0.9827614526
N/20	20	0.04784	11.13110368	0.9580648936
N/32	32	0.028704	18.55183946	0.9766162471
N/64	64	0.022976	23.17688022	0.9720417061
N/128	128	0.020416	26.0830721	0.9692330914

Paleti Krishnasai
CED18I039

Graphical Inference:





Paleti Krishnasai
CED18I039

Inference:

From the Graph it is observed that the speedup starts to increase steadily after 20 threads constantly and the highest Speedup is for 128 threads.
It also increases steadily from 1 to 16 threads, but goes down at 20 and rises steadily again.

Runtime Decreases drastically from 1 to 4 threads and steadily decreases from then on till 128, with a minor rise at threads = 20.