**Class:** Final Year (Computer Science and Engineering)

**Year:** 2022-23 **Semester:** 1

**Course:** High Performance Computing Lab

**Practical No. 9**

**Exam Seat No:**

2019BTECS00033 – Teknath Jha

**Title of practical:**

CUDA programming

**Problem Statement 1:**

Implement Vector-Vector addition using CUDA C. State and justify the speedup using different size of threads and blocks.

**Output 1:**

**CUDA Kernel Statistics:**

**Time(%) Total Time (ns) Instances Average Minimum Maximum Name**

**------- --------------- --------- ------- ------- ------- --------------------------------------**

**100.0 92384 1 92384.0 92384 92384 add\_vectors(double\*, double\*, double\*)**

**---------------------------**

**N = 1048576**

**Threads Per Block = 128**

**Blocks In Grid = 8192**

**---------------------------**

**CUDA Kernel Statistics: 256 TPB**

**Time(%) Total Time (ns) Instances Average Minimum Maximum Name**

**------- --------------- --------- ------- ------- ------- --------------------------------------**

**100.0 91806 1 91806.0 91806 91806 add\_vectors(double\*, double\*, double\*)**

**---------------------------**

**N = 1048576**

**Threads Per Block = 256**

**Blocks In Grid = 4096**

**---------------------------**

**CUDA Kernel Statistics:**

**Time(%) Total Time (ns) Instances Average Minimum Maximum Name**

**------- --------------- --------- ------- ------- ------- --------------------------------------**

**100.0 92127 1 92127.0 92127 92127 add\_vectors(double\*, double\*, double\*)**

**---------------------------**

**N = 1048576**

**Threads Per Block = 512**

**Blocks In Grid = 2048**

**---------------------------**

**CUDA Kernel Statistics:**

**Time(%) Total Time (ns) Instances Average Minimum Maximum Name**

**------- --------------- --------- ------- ------- ------- --------------------------------------**

**100.0 95584 1 95584.0 95584 95584 add\_vectors(double\*, double\*, double\*)**

**---------------------------**

**N = 1048576**

**Threads Per Block = 1024**

**Blocks In Grid = 1024**

**---------------------------**

**CUDA Kernel Statistics:**

**Time(%) Total Time (ns) Instances Average Minimum Maximum Name**

**------- --------------- --------- -------- ------- ------- --------------------------------------**

**100.0 382051 1 382051.0 382051 382051 add\_vectors(double\*, double\*, double\*)**

**---------------------------**

**N = 4194304**

**Threads Per Block = 1024**

**Blocks In Grid = 4096**

**For Sequential :**

Operating System Runtime API Statistics:

Time(%) Total Time (ns) Num Calls Average Minimum Maximum Name

------- --------------- --------- ------- ------- ------- ------

57.5 5025 1 5025.0 5025 5025 putc

42.5 3709 1 3709.0 3709 3709 fwrite

**Speedup :**

**Speedup = serial / parallel**

**Getting Average as Ts = (95584+92127+91806)/3 = 93172.333**

**And Ts is 12342**

**Both Ts and Tp computed on same give NVIDIA EC2 instance**

**Thus Speedup here is Ts/Tp = 7.5**

**Problem Statement 2:**

Implement N-Body Simulator using CUDA C. State and justify the speedup using different size of threads and blocks.

**Screenshot 2:**

**bodyForce<<<1024,64>>>(p, dt, nBodies);**

**CUDA Kernel Statistics:**

**Time(%) Total Time (ns) Instances Average Minimum Maximum Name**

**------- --------------- --------- -------- ------- ------- ----------------------------**

**100.0 9773062 10 977306.2 904535 1412947 bodyForce(Body\*, float, int)**

**bodyForce<<<512,64>>>(p, dt, nBodies);**

**CUDA Kernel Statistics:**

**Time(%) Total Time (ns) Instances Average Minimum Maximum Name**

**------- --------------- --------- -------- ------- ------- ----------------------------**

**100.0 9549214 10 954921.4 914198 1197651 bodyForce(Body\*, float, int)**

**bodyForce<<<512,64>>>(p, dt, nBodies);**

**CUDA Kernel Statistics:**

**Time(%) Total Time (ns) Instances Average Minimum Maximum Name**

**------- --------------- --------- -------- ------- ------- ----------------------------**

**100.0 9726141 10 972614.1 889010 1608037 bodyForce(Body\*, float, int)**

**bodyForce<<<1024,64>>>(p, dt, nBodies);**

**CUDA Kernel Statistics:**

**Time(%) Total Time (ns) Instances Average Minimum Maximum Name**

**------- --------------- --------- -------- ------- ------- ----------------------------**

**100.0 9500817 10 950081.7 915125 1186546 bodyForce(Body\*, float, int)**

For sequential :

Time(%) Total Time (ns) Num Calls Average Minimum Maximum Name

------- --------------- --------- ------- ------- ------- ------

69.5 12610 1 12610.0 12610 12610 putc

30.5 5535 1 5535.0 5535 5535 fwrite

**Speedup :**

**Speedup = serial / parallel**

**Getting Average as Ts = (950081+972614+954921)/3 =** 959205.33

**And Ts is 12610**

**Both Ts and Tp computed on same give NVIDIA EC2 instance**

**Thus Speedup here is Ts/Tp = 76**

**Github Link:**

<https://github.com/Teknath-jha/HPC-LAB-2019BTECS00033/tree/main/Assignment-9>