Name: Nikita Shivchandra Khot

PRN No: 2020BTECS00041

High Performance Computing Lab

Practical No. 8

Title of practical: Implementation of Vector-Vector addition & N-Body Simulator using CUDA C

Problem Statement 1:
Implement Vector-Vector addition using CUDA C. State and justify the speedup using different size of threads and blocks.

Screenshots:

```
#include <ctime>
#include <cstdlib>
#include <bits/stdc++.h>
#define SIZE 100000000
void vectorAdd(int* a, int* b, int* c, int size) {
    for (int i = 0; i < size; ++i) {
        c[i] = a[i] + b[i];
    }
}

int main() {
    clock_t start_time = clock();
    int* a = new int[SIZE];
    int* b = new int[SIZE];
    int* c = new int[SIZE];
    for (int i = 0; i < SIZE; ++i) {
        a[i] = rand();
        b[i] = rand();
    }
    vectorAdd(a, b, c, SIZE);
    clock_t end_time = clock();
    double execution_time = static_cast<double>(end_time - start_time)
/ CLOCKS_PER_SEC;
    std::cout << "\nExecution time for size " << SIZE << ": " <<
execution_time << " seconds\n" << std::endl;
    delete[] a;
    delete[] b;
    delete[] c;
    return 0;
}
```

Serial:

```
PS D:\FinalYear\HPCL\Assignment 8\output> & .\'vector_addition.exe'

Execution time for size 100000: 0.006 seconds

PS D:\FinalYear\HPCL\Assignment 8\output> cd 'd:\FinalYear\HPCL\Assignment 8\output'
PS D:\FinalYear\HPCL\Assignment 8\output> & .\'vector_addition.exe'

Execution time for size 1000000: 0.029 seconds

PS D:\FinalYear\HPCL\Assignment 8\output> cd 'd:\FinalYear\HPCL\Assignment 8\output'
PS D:\FinalYear\HPCL\Assignment 8\output> & .\'vector_addition.exe'

Execution time for size 10000000: 0.281 seconds

PS D:\FinalYear\HPCL\Assignment 8\output> cd 'd:\FinalYear\HPCL\Assignment 8\output'
PS D:\FinalYear\HPCL\Assignment 8\output> & .\'vector_addition.exe'

Execution time for size 100000000: 2.76 seconds
```

Parallel:

```
!nvcc -o vector-add-manual-alloc 07-manual-malloc/solutions/01-manual-malloc-solution.cu -run
Execution time for size 100000 and No. of threads 1024: 0.127251 seconds

!nvcc -o vector-add-manual-alloc 07-manual-malloc/solutions/01-manual-malloc-solution.cu -run
Execution time for size 100000 and No. of threads 2048: 0.13035 seconds

!nvcc -o vector-add-manual-alloc 07-manual-malloc/solutions/01-manual-malloc-solution.cu -run
Execution time for size 100000 and No. of threads 4096: 0.115844 seconds

!nvcc -o vector-add-manual-alloc 07-manual-malloc/solutions/01-manual-malloc-solution.cu -run
Execution time for size 100000 and No. of threads 8192: 0.124343 seconds

!nvcc -o vector-add-manual-alloc 07-manual-malloc/solutions/01-manual-malloc-solution.cu -run
Execution time for size 100000 and No. of threads 16384: 0.128392 seconds

!nvcc -o vector-add-manual-alloc 07-manual-malloc/solutions/01-manual-malloc-solution.cu -run
Execution time for size 100000 and No. of threads 32768: 0.127004 seconds

!nvcc -o vector-add-manual-alloc 07-manual-malloc/solutions/01-manual-malloc-solution.cu -run
Execution time for size 100000 and No. of threads 65536: 0.127113 seconds
```

```
!nvcc -o vector-add-manual-alloc 07-manual-malloc/solutions/01-manual-malloc-solution.cu -run
```

```
Execution time for size 1000000 and No. of threads 1024: 0.183386 seconds
```

```
!nvcc -o vector-add-manual-alloc 07-manual-malloc/solutions/01-manual-malloc-solution.cu -run
```

```
Execution time for size 1000000 and No. of threads 2048: 0.173838 seconds
```

```
!nvcc -o vector-add-manual-alloc 07-manual-malloc/solutions/01-manual-malloc-solution.cu -run
```

```
Execution time for size 1000000 and No. of threads 4096: 0.170051 seconds
```

```
!nvcc -o vector-add-manual-alloc 07-manual-malloc/solutions/01-manual-malloc-solution.cu -run
```

```
Execution time for size 1000000 and No. of threads 8192: 0.170245 seconds
```

```
!nvcc -o vector-add-manual-alloc 07-manual-malloc/solutions/01-manual-malloc-solution.cu -run
```

```
Execution time for size 1000000 and No. of threads 16384: 0.164162 seconds
```

```
!nvcc -o vector-add-manual-alloc 07-manual-malloc/solutions/01-manual-malloc-solution.cu -run
```

```
Execution time for size 1000000 and No. of threads 32768: 0.190183 seconds
```

```
!nvcc -o vector-add-manual-alloc 07-manual-malloc/solutions/01-manual-malloc-solution.cu -run
```

```
Execution time for size 1000000 and No. of threads 65536: 0.170634 seconds
```

```
!nvcc -o vector-add-manual-alloc 07-manual-malloc/solutions/01-manual-malloc-solution.cu -run
```

```
Execution time for size 10000000 and No. of threads 1024: 0.737486 seconds
```

```
!nvcc -o vector-add-manual-alloc 07-manual-malloc/solutions/01-manual-malloc-solution.cu -run
```

```
Execution time for size 10000000 and No. of threads 2048: 0.712315 seconds
```

```
!nvcc -o vector-add-manual-alloc 07-manual-malloc/solutions/01-manual-malloc-solution.cu -run
```

```
Execution time for size 10000000 and No. of threads 4096: 0.72509 seconds
```

```
!nvcc -o vector-add-manual-alloc 07-manual-malloc/solutions/01-manual-malloc-solution.cu -run
```

```
Execution time for size 10000000 and No. of threads 8192: 0.739249 seconds
```

```
!nvcc -o vector-add-manual-alloc 07-manual-malloc/solutions/01-manual-malloc-solution.cu -run
```

```
Execution time for size 10000000 and No. of threads 16384: 0.734344 seconds
```

```
!nvcc -o vector-add-manual-alloc 07-manual-malloc/solutions/01-manual-malloc-solution.cu -run
```

```
Execution time for size 10000000 and No. of threads 32768: 0.714431 seconds
```

```
!nvcc -o vector-add-manual-alloc 07-manual-malloc/solutions/01-manual-malloc-solution.cu -run
```

```
Execution time for size 10000000 and No. of threads 65536: 0.744849 seconds
```

```
!nvcc -o vector-add-manual-alloc 07-manual-malloc/solutions/01-manual-malloc-solution.cu -run
Execution time for size 100000000 and No. of threads 1024: 6.13931 seconds

!nvcc -o vector-add-manual-alloc 07-manual-malloc/solutions/01-manual-malloc-solution.cu -run
Execution time for size 100000000 and No. of threads 2048: 6.15163 seconds

!nvcc -o vector-add-manual-alloc 07-manual-malloc/solutions/01-manual-malloc-solution.cu -run
Execution time for size 100000000 and No. of threads 4096: 6.09415 seconds

!nvcc -o vector-add-manual-alloc 07-manual-malloc/solutions/01-manual-malloc-solution.cu -run
Execution time for size 100000000 and No. of threads 8192: 6.16268 seconds

!nvcc -o vector-add-manual-alloc 07-manual-malloc/solutions/01-manual-malloc-solution.cu -run
Execution time for size 100000000 and No. of threads 16384: 6.13403 seconds

!nvcc -o vector-add-manual-alloc 07-manual-malloc/solutions/01-manual-malloc-solution.cu -run
Execution time for size 100000000 and No. of threads 32768: 6.1795 seconds

!nvcc -o vector-add-manual-alloc 07-manual-malloc/solutions/01-manual-malloc-solution.cu -run
Execution time for size 100000000 and No. of threads 65536: 6.10776 seconds
```

Output:

Sequential Execution Time

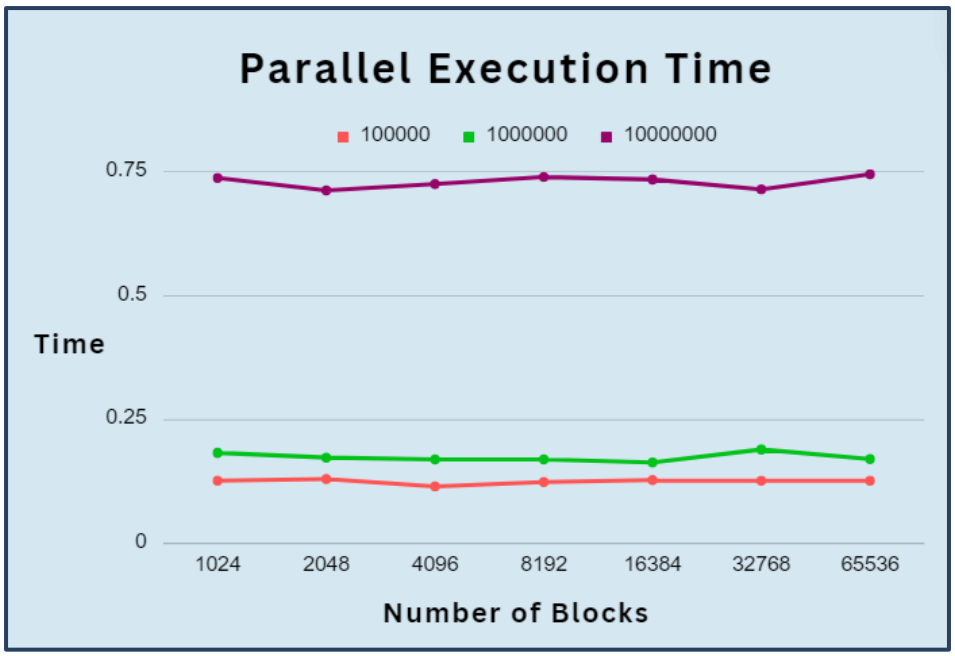
Input Size	Execution Time
100000	0.006 sec
1000000	0.029 sec
10000000	0.281 sec
100000000	2.76 sec

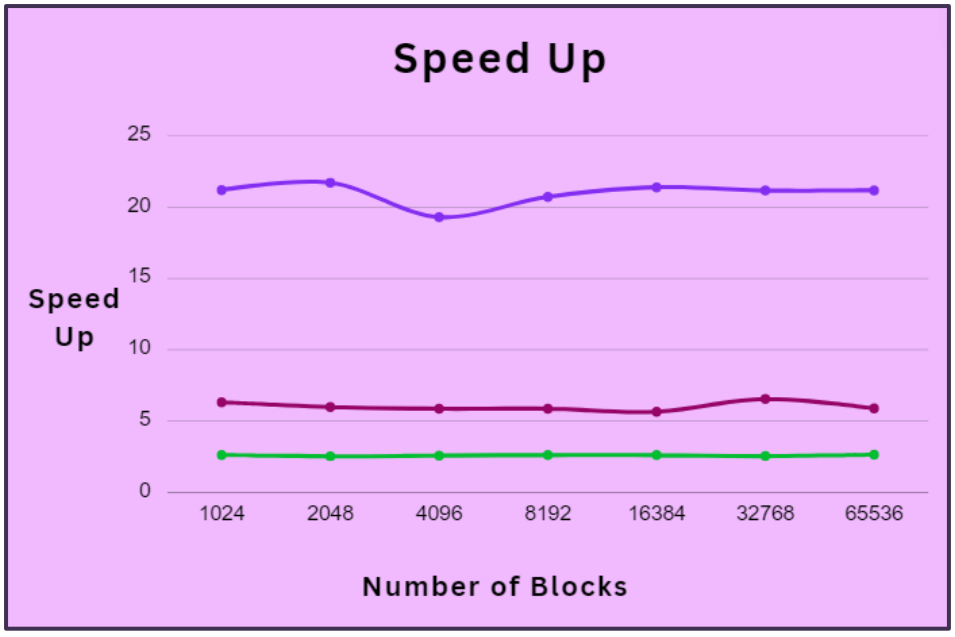
Parallel Execution Time

Input Size\ Threads	1024	2048	4096	8192	16384	32768	65536
100000	0.1272	0.1304	0.1158	0.1243	0.1284	0.1270	0.1271
1000000	0.1834	0.1738	0.1701	0.1702	0.1642	0.1902	0.1706
10000000	0.7375	0.7123	0.7250	0.7392	0.7343	0.7144	0.7448
100000000	6.1393	6.1516	6.0942	6.1627	6.1340	6.1795	6.1078

Speed Up

Input Size\ Threads	1024	2048	4096	8192	16384	32768	65536
100000	21.2085	21.7250	19.3073	20.7238	21.3987	21.1673	21.1855
1000000	6.3237	5.9944	5.8638	5.8705	5.6608	6.558	5.8839
10000000	2.6245	2.5349	2.5804	2.6133	2.6133	2.5425	2.6507
100000000	2.2244	2.2289	2.208	2.2329	2.2225	2.2389	2.213





Problem Statement 2:

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

Screenshots of Output:

Serial:

```
Running nbody simulator with 4096 bodies
-----------------------------------------

Application should run faster than 0.9s
Your application ran in: 4.1776s
Your application is not yet fast enough
```

Parallel:

```
run_assessment()
```

```
Running nbody simulator with 4096 bodies
----------------------------------------

Application should run faster than 0.9s
Your application ran in: 0.1979s
Your application reports  17.045 Billion Interactions / second
Number of Blocks: 8threadsPerBlock : 512
Your results are correct
```

```
run_assessment()
```

```
Running nbody simulator with 4096 bodies
----------------------------------------

Application should run faster than 0.9s
Your application ran in: 0.2123s
Your application reports  20.314 Billion Interactions / second
Number of Blocks: 16threadsPerBlock: 256
Your results are correct
```

```
run_assessment()
```

```
Running nbody simulator with 4096 bodies
----------------------------------------

Application should run faster than 0.9s
Your application ran in: 0.2149s
Your application reports  21.375 Billion Interactions / second
Number of Blocks: 32threadsPerBlock : 128
Your results are correct
```

```
run_assessment()
```

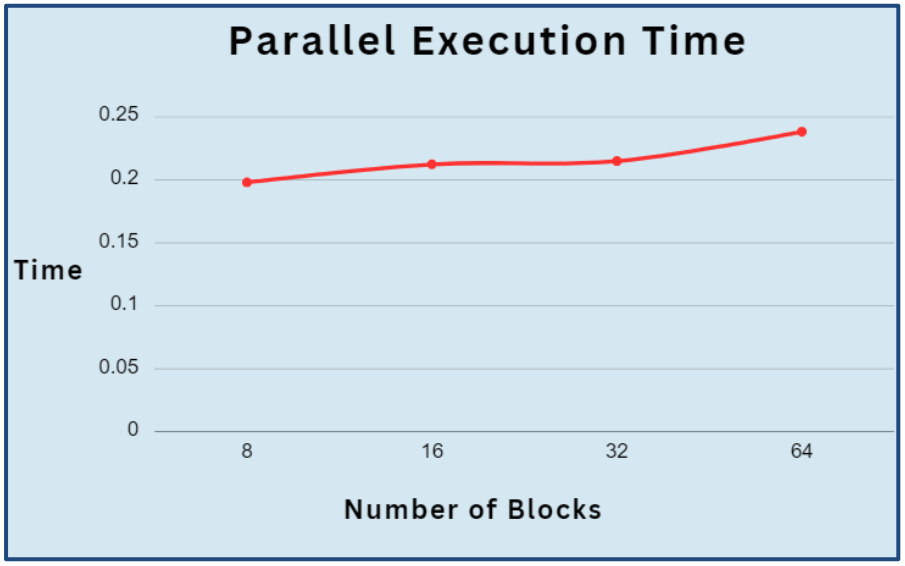
```
Running nbody simulator with 4096 bodies
----------------------------------------

Application should run faster than 0.9s
Your application ran in: 0.2140s
Your application reports  20.922 Billion Interactions / second
Number of Blocks: 64threadsPerBlock : 64
Your results are correct
```

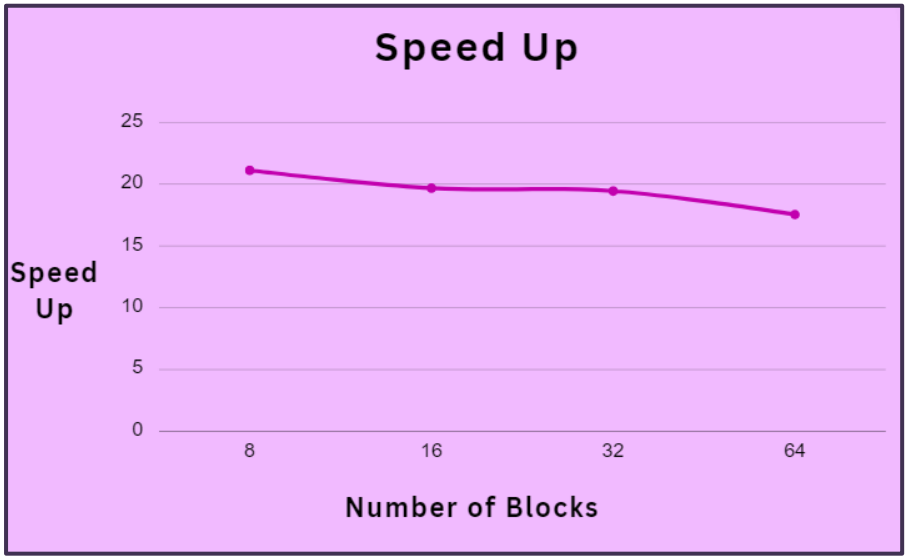
Speedup analysis:

1. **Tabular**

Number of Threads per block	Number of Blocks	Time	Speed Up
512	8	0.1979 sec	21.1096
256	16	0.2123 sec	19.6778
128	32	0.2149 sec	19.4397
64	64	0.2382 sec	17.5382

2. Graphical


Parallel Execution Time
0.25
0.2
0.15
0.1
0.05
0
Time
8
16
32
64
Number of Blocks


Speed Up
25
20
15
10
5
0
Speed Up
8
16
32
64
Number of Blocks