

# HPC CUDA C TUTORIAL 10 REPORT

## CS22B2015 – HARSHITH B

### 1. Introduction

This report analyzes the performance of serial and parallel implementations for calculating the vector dot product of 1.5 million floating-point double numbers. A serial code written in C and a parallel code written in CUDA (.cu) were developed to perform this computation, and their execution times are compared to evaluate efficiency and potential speedup.

### 2. Serial Code Snippet

```
clock_t start = clock();
for (int i = 0; i < count; i++) {
    dot_product += h_arr1[i] * h_arr2[i];
}
clock_t end = clock();
```

### 3. Parallel Code Snippet

```
_global_ void dot_product(double *d_arr1, double *d_arr2, double *d_dot, int n) {
    int tid = threadIdx.x + blockIdx.x * blockDim.x;
    if (tid < n) {
        atomicAdd(d_dot, d_arr1[tid] * d_arr2[tid]);
    }
}
```

#### 4. Terminal Output Screenshot

```
(venv) harshith@harshithb:~/Projects /SEM 6/HPC/tutorial-10$ ./dot_p
Dot Product: 3758429994325991936.00
Kernel Execution Time: 0.038129 seconds
(venv) harshith@harshithb:~/Projects /SEM 6/HPC/tutorial-10$ ./dot_s
Dot Product: 3758429994325987840.00
Time taken: 0.055251
(venv) harshith@harshithb:~/Projects /SEM 6/HPC/tutorial-10$
```

#### 5. Serial and Parallel Code Execution Time and Speedup Calculation

- Serial Code Execution Time: 0.055251 seconds
- Parallel Code Execution Time: 0.038129 seconds

Speedup Calculation:

**Speedup = Serial Execution Time / Parallel Execution Time**

**Speedup Estimation = 1.449**