**CSE467 - Parallel and Distributed Computing**

**Assignment 2**

**Hiba Mallick - 24015**

**Objective: Write a program for parallel vector addition by using multithreading.**

The complexity of vector addition is O(N) and the execution time depends on the vector length. In this assignment you will learn how to pass and return values from a thread and how logically work can be divided into multiple threads to be performed in parallel and reduce total execution time. You need do device some logic for load balancing.

**Submit a word file containing all code, output snapshots and the comparison table and graph.**

**Input:**

1. Vector Length: **N** (to be used to create and initialize 3 vectors)
2. No of Threads: **T**

**Processing:** Determine the execution time of vector addition with various values of T and N given in the table.

T0 = get clock cycle/system time

Perform V1 = V2 + V3 with load balancing among T threads.

T1 = get clock cycle/system time

Time elapsed = T1 – T0

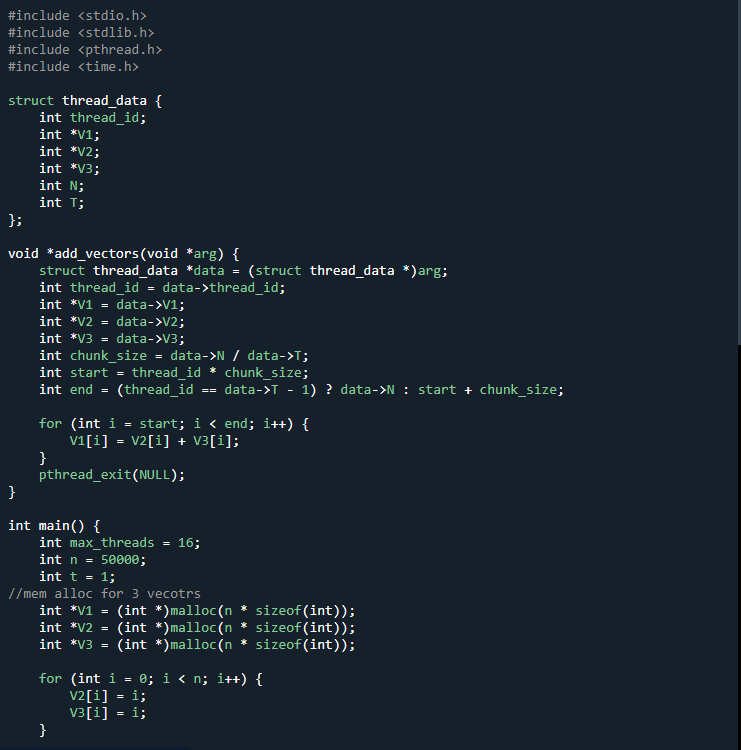
**Output:** Time elapsed - Tables show the comparison

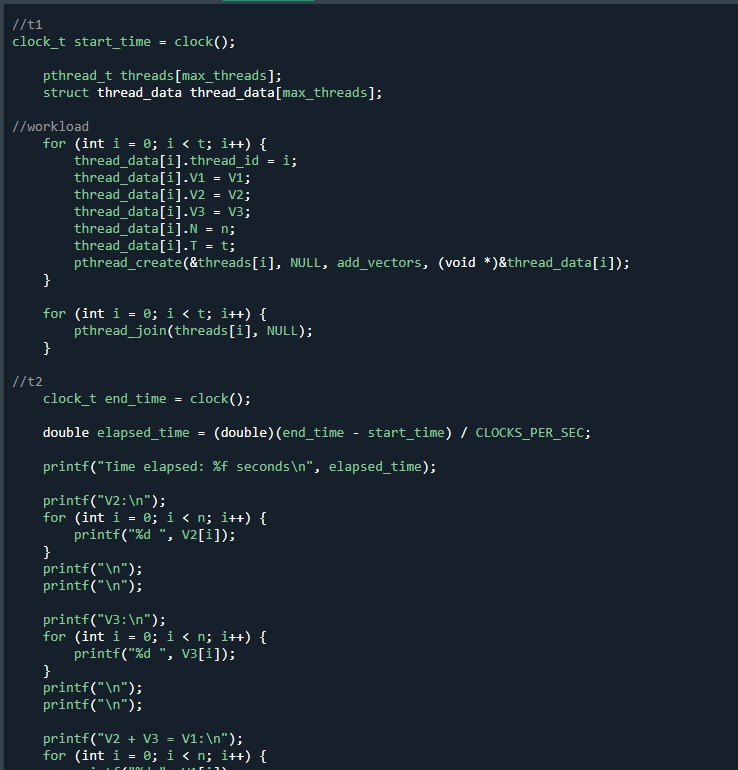
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | N (vector size in thousands) | | | | | | | |
|  |  | 50 | 100 | 150 | 200 | 250 | 300 | 350 | 400 |
| T (no of threads) | 1 | 0.000766 | 0.001011 | 0.000517 | 0.000836 | 0.000727 | 0.000908 | 0.001185 | 0.001659 |
| 5 | 0.000489 | 0.001155 | 0.000535 | 0.000811 | 0.000761 | 0.001286 | 0.002053 | 0.001482 |
| 10 | 0.000997 | 0.001233 | 0.000643 | 0.001186 | 0.000897 | 0.002052 | 0.001168 | 0.001677 |
| 15 | 0.001430 | 0.001217 | 0.000849 | 0.001000 | 0.001338 | 0.001278 | 0.001537 | 0.001427 |
| 20 | 0.001091 | 0.001174 | 0.001101 | 0.000982 | 0.000941 | 0.001458 | 0.002274 | 0.001908 |
| 25 | 0.001103 | 0.001297 | 0.001212 | 0.000932 | 0.000869 | 0.001397 | 0.002147 | 0.001445 |
| 30 | 0.001100 | 0.001284 | 0.001420 | 0.001022 | 0.000876 | 0.001038 | 0.001238 | 0.001443 |

**Draw the Graph for above table.**

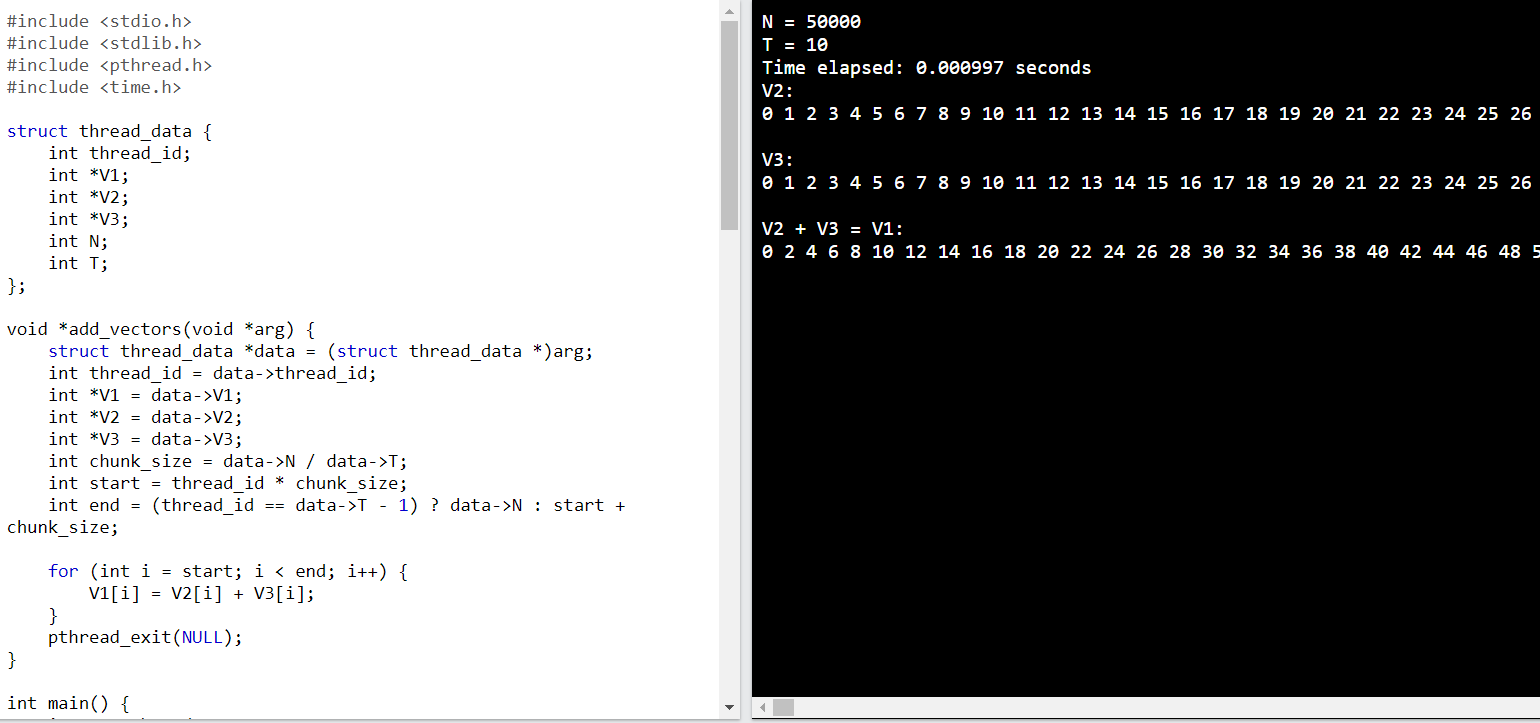
**A graph of different colored lines

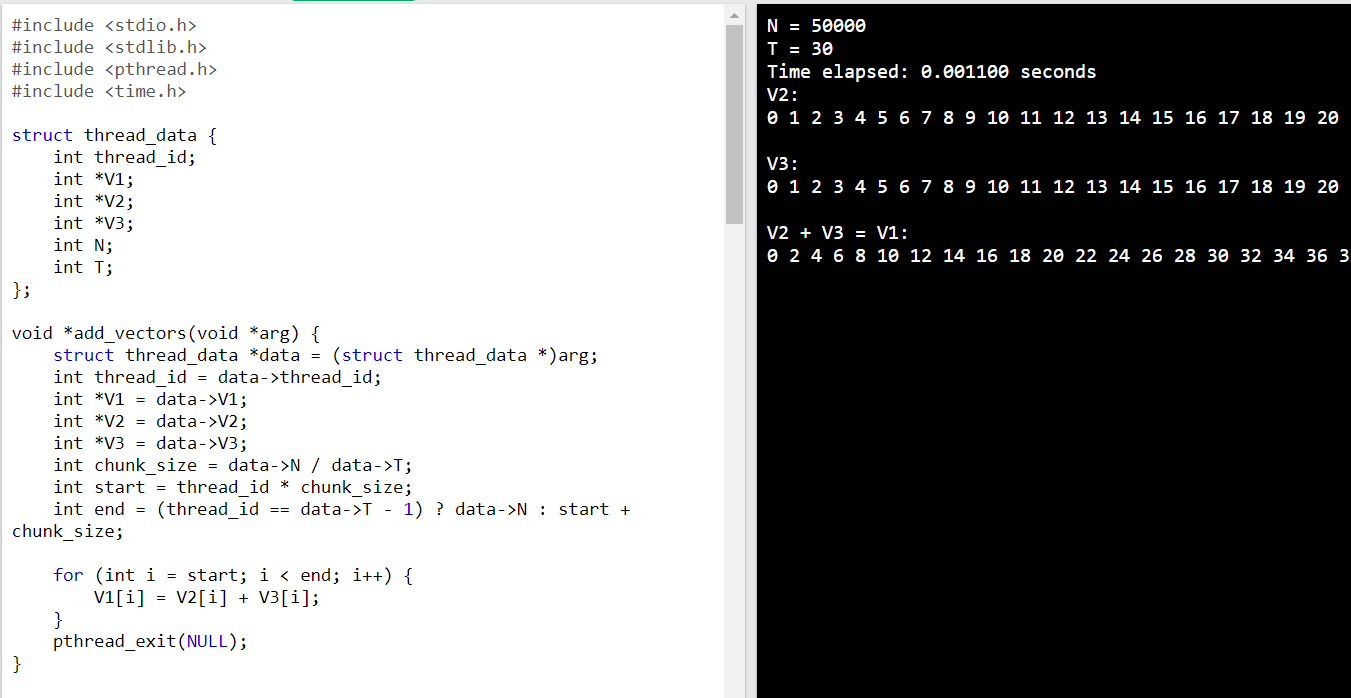
Description automatically generated**

****

**A blue square with white text

Description automatically generated with medium confidence**

****

****