**Course: High Performance Computing Lab**

**Practical No 3**

**PRN : 23520006**

**Name: Vivek katkar**

**Batch : B8**

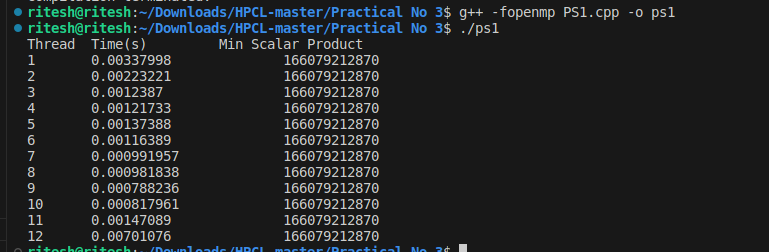
**Title -** Study and Implementation of schedule, nowait, reduction, ordered and collapse clauses

**Problem Statement 1:**

Analyse and implement a Parallel code for below program using OpenMP.

Program 1 :- C Program to find the minimum scalar product of two vectors (dot product)

****

****

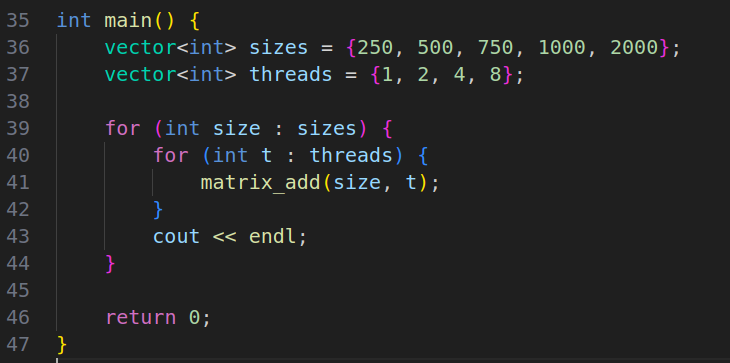
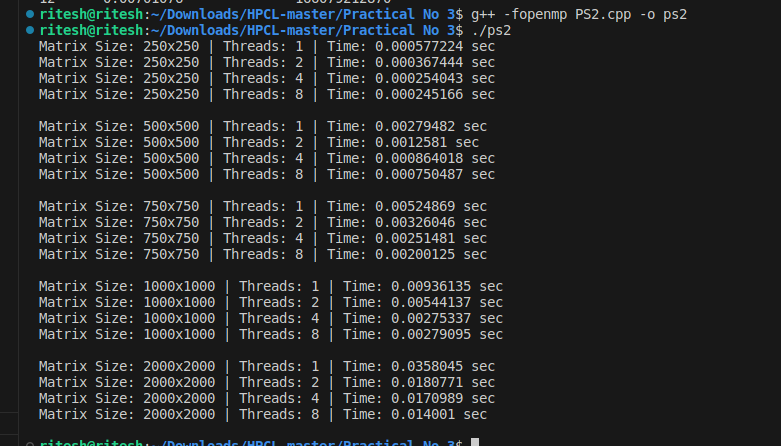
**Information and analysis:**

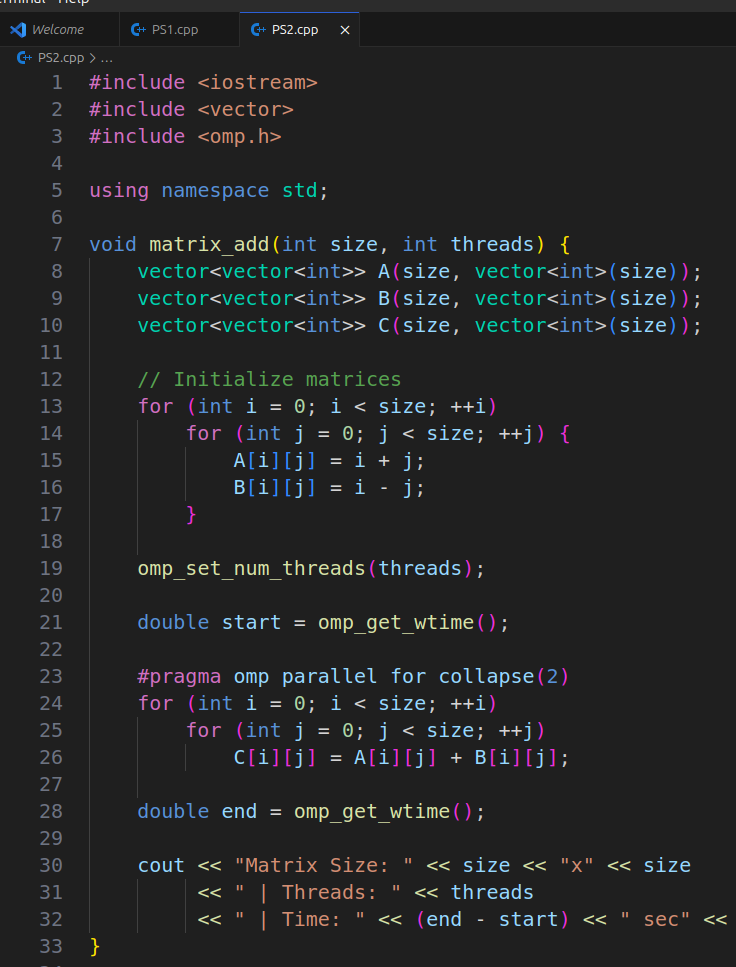
As we have 6 physical core so we can see the decrease in time but when we are going form the 7 to 12 there is slight increase in time the reason is the thread scheduling overhead.

**Problem Statement 2:**

Write OpenMP code for two 2D Matrix addition, vary the size of your matrices from 250, 500, 750, 1000, and 2000 and measure the runtime with one thread (Use functions in C in calculate the execution time or use GPROF)

1. For each matrix size, change the number of threads from 2,4,8., and plot the speedup versus the number of threads.
2. Explain whether or not the scaling behaviour is as expected.

****

****

**Information and analysis:**

**Speedup =**

|  |  |  |  |
| --- | --- | --- | --- |
| Matrix Size | Threads | Time (s) | Speedup |
| 250×250 | 1 | 0.0021848 | 1.00 |
|  | 2 | 0.00135728 | 1.61 |
|  | 4 | 0.000790233 | 2.77 |
|  | 8 | 0.000758023 | 2.88 |
| 500×500 | 1 | 0.0119584 | 1.00 |
|  | 2 | 0.0049791 | 2.40 |
|  | 4 | 0.00231982 | 5.15 |
|  | 8 | 0.00161937 | 7.38 |
| 750×750 | 1 | 0.0187392 | 1.00 |
|  | 2 | 0.00687037 | 2.73 |
|  | 4 | 0.00393679 | 4.76 |
|  | 8 | 0.0028859 | 6.49 |
| 1000×1000 | 1 | 0.0177866 | 1.00 |
|  | 2 | 0.0102103 | 1.74 |
|  | 4 | 0.00609756 | 2.91 |
|  | 8 | 0.0046221 | 3.85 |
| 2000×2000 | 1 | 0.050548 | 1.00 |
|  | 2 | 0.0275277 | 1.84 |
|  | 4 | 0.0143359 | 3.52 |
|  | 8 | 0.0135943 | 3.72 |

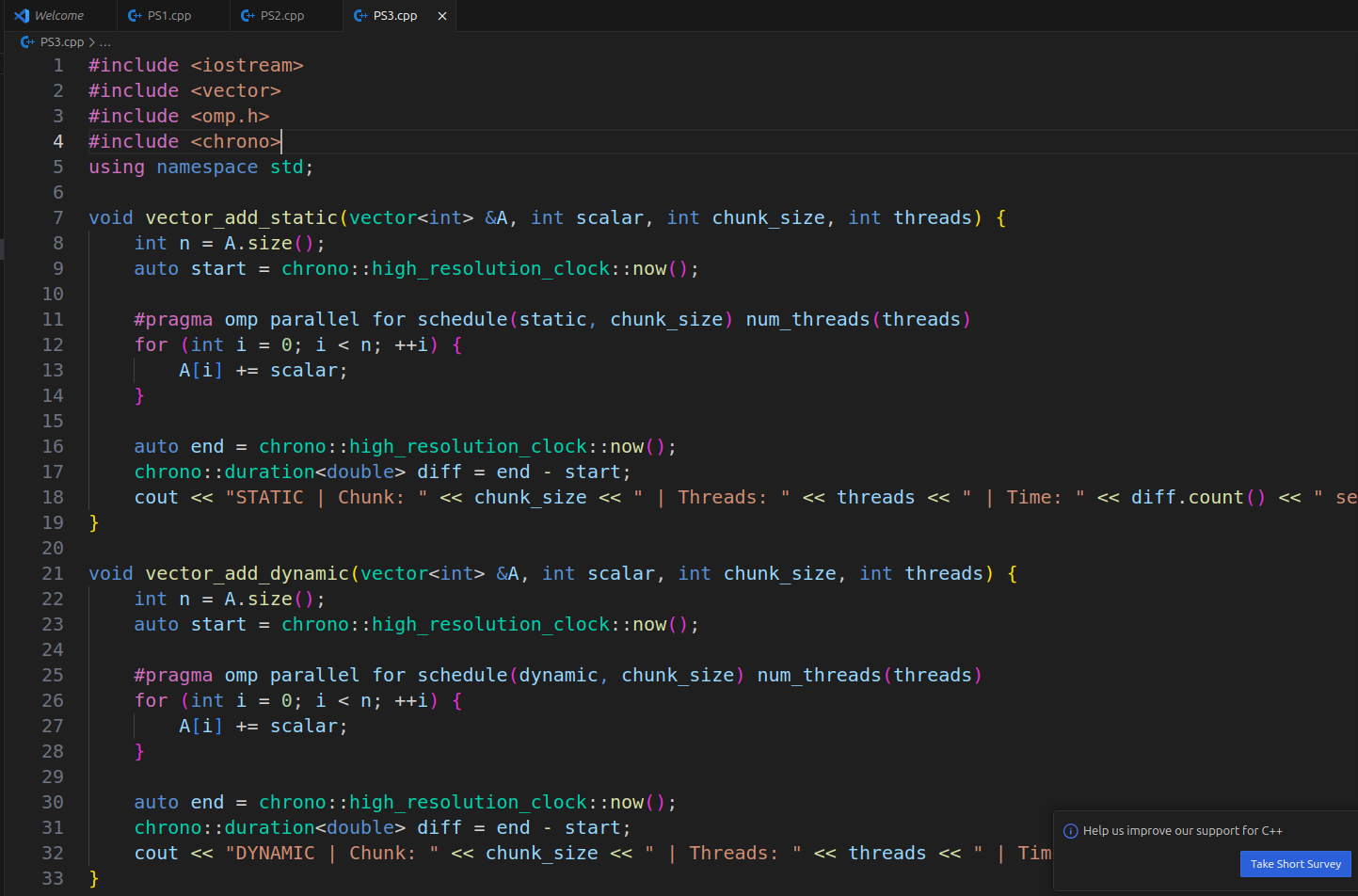
**Information & Analysis** :

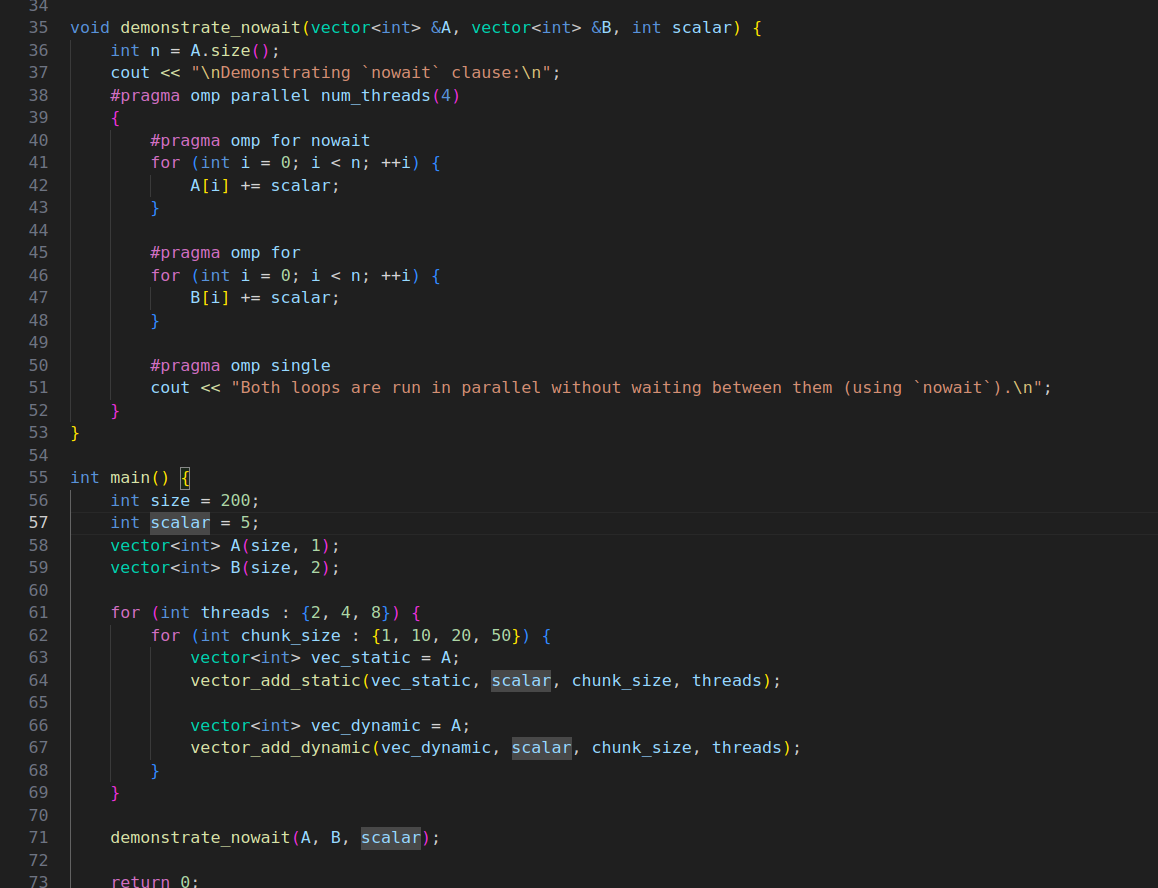
As we have 6 physical core so we can see the decrease in time from thread 2 to 4 but when we are going form the 8 thread there is slight increase in time the reason is the thread scheduling overhead.

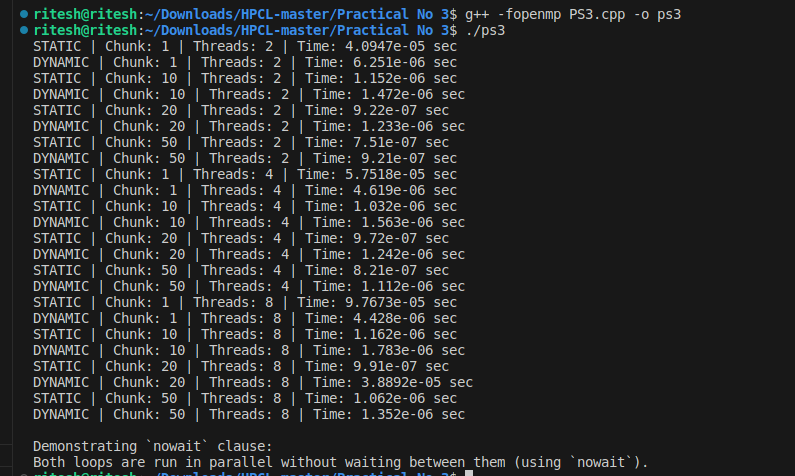
**Problem Statement 3:**

For 1D Vector (size=200) and scalar addition, Write a OpenMP code with the following: i. Use STATIC schedule and set the loop iteration chunk size to various sizes when changing the size of your matrix. Analyze the speedup. ii. Use DYNAMIC schedule and set the loop iteration chunk size to various sizes when changing the size of your matrix. Analyze the speedup. iii. Demonstrate the use of no wait clause.

**Screenshots:**

****

****

****

**GitHub Link:** <https://github.com/vivekkatkar/hpcl>

**Conclusion**

This practical explored advanced OpenMP clauses like schedule, nowait, reduction, ordered, and collapse through vector and matrix operations. The experiments highlighted how performance scales with thread count up to the number of physical cores, while excess threads introduce scheduling overhead, and how scheduling strategies impact load balancing and execution time.