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

**Year:** 2024-25 **Semester:** 1

**Course:** High Performance Computing Lab

**Practical No. 2**

**Exam Seat No: 21510055**

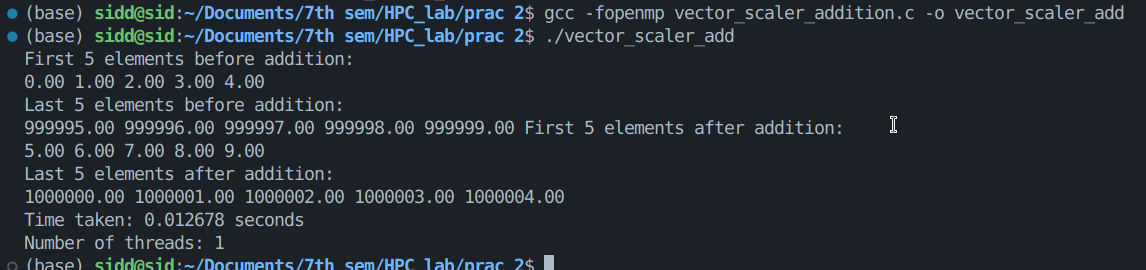
**Title of practical: Study and implementation of basic OpenMP clauses**

Implement following Programs using OpenMP with C:

1. Vector Scalar Addition
2. Calculation of value of Pi

Analyse the performance of your programs for different number of threads and Data size.

**Problem Statement 1:** Implementation of vector scalar addition using OpenMP with C.

**Screenshots: **

**Information:**

* + Vector scalar addition involves adding a scalar value to each element of a vector.
  + OpenMP is a parallel programming API that can be used to parallelize the addition operation.
  + The problem requires implementing this operation efficiently using C and OpenMP.

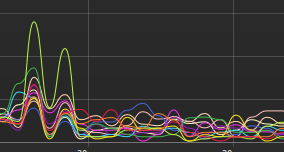
**Analysis:**

Main function: *vectorScalarAdd().*

Utilizes dynamic memory allocation for the vector.

Measures execution time using omp\_get\_wtime().

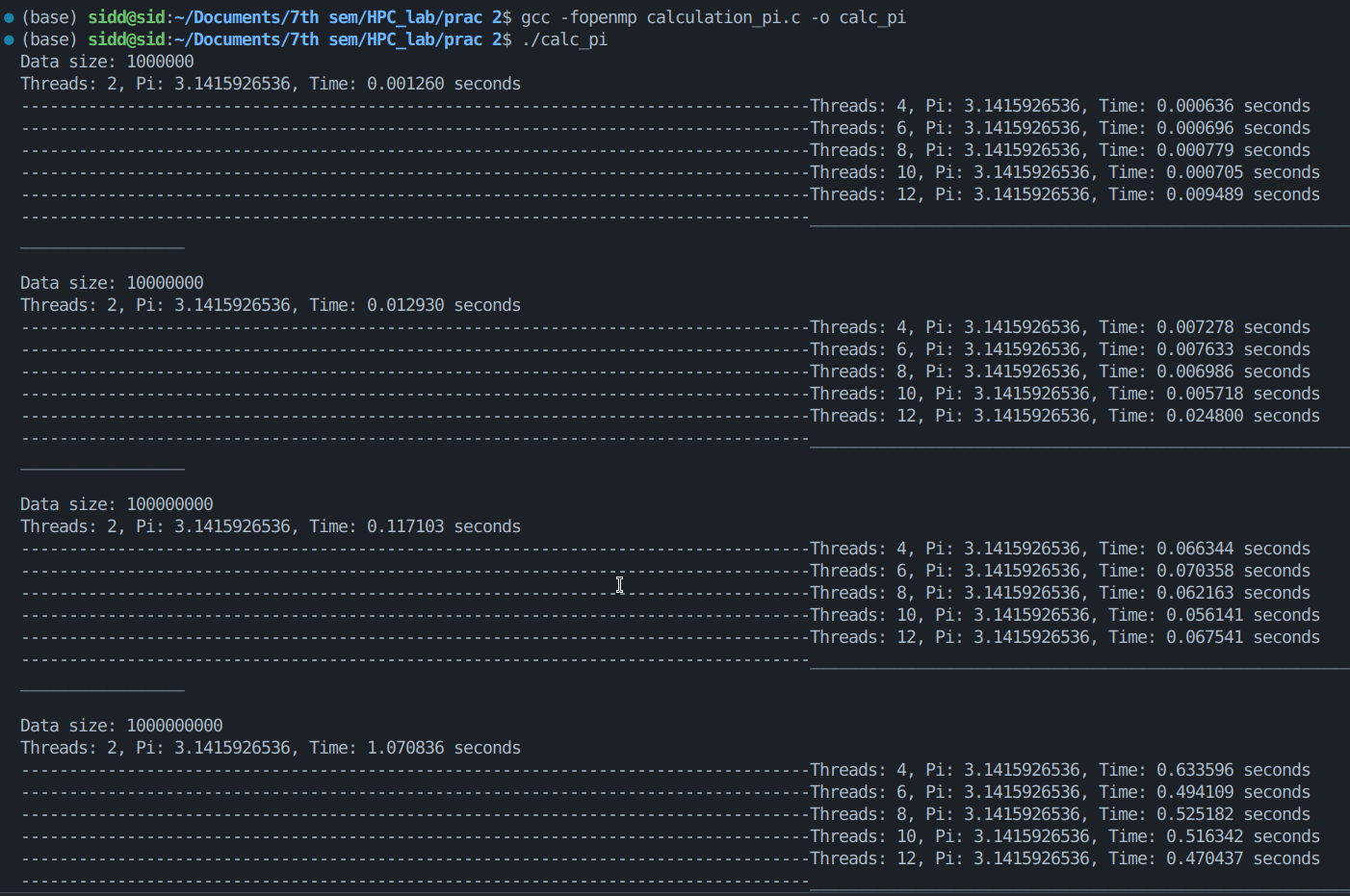
Efficiency depends on vector size and available CPU cores.



From the above figure we can clearly see that when executing the program all cores of cpus were utilized.

**Problem Statement 2:** Calculate the value of Pi using OpenMP in C, analyzing performance for different numbers of threads and data sizes.

**Screenshots:**

****

**Information:**

* Calculate Pi using numerical integration : <https://www.mathworks.com/help/parallel-computing/numerical-estimation-of-pi-using-message-passing.html>
* Use OpenMP for parallelization
* Analyze performance with thread counts: 2, 4, 6, 8, 10, 12
* Vary data size (number of intervals for integration)

**Analysis:**

1. Execution time decreases as thread count increases, up to a point.
2. For smaller data sizes, increasing thread count can increase execution time because the overhead of creating, managing, and synchronizing threads outweighs the benefits of parallel processing.
3. Larger data sizes benefit more from increased thread counts.

**Github Link:** [***https://github.com/Sid-1164/HPC\_lab/tree/main/prac%202***](https://github.com/Sid-1164/HPC_lab/tree/main/prac%202)