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

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

**Course:** High Performance Computing Lab

**Name:** Abhijeet Kamalekar

**PRN:** 2020BTECS00010

**Practical No. 2**

**Exam Seat No:**

**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:**

**Screenshots:**

#include <omp.h>

#include <stdio.h>

#include <stdlib.h>

void vector\_scalar\_addition(int \*vector, int scalar, int size) {

    omp\_set\_num\_threads(15);

    int i;

    #pragma omp parallel for shared(vector, scalar, size) private(i)

    for (i = 0; i < size; i++) {

        vector[i] += scalar;

    }

}

int main() {

    int size = 100000;

    int scalar = 5;

    int \*vector = (int \*)malloc(size \* sizeof(int));

    for (int i = 0; i < size; i++) {

        vector[i] = i;

    }

    double start\_time = omp\_get\_wtime();

    vector\_scalar\_addition(vector, scalar, size);

    double end\_time = omp\_get\_wtime();

    printf("Time taken: %f seconds\n", end\_time - start\_time);

    free(vector);

    return 0;

}

****

**Problem Statement 2:**

**Screenshots:**

#include <omp.h>

#include <stdio.h>

double calculate\_pi(int num\_steps) {

    double step = 1.0 / (double)num\_steps;

    double sum = 0.0;

    #pragma omp parallel for reduction(+:sum)

    for (int i = 0; i < num\_steps; i++) {

        double x = (i + 0.5) \* step;

        sum += 4.0 / (1.0 + x \* x);

    }

    return step \* sum;

}

int main() {

    int num\_steps = 100000000;

    double start\_time = omp\_get\_wtime();

    double pi = calculate\_pi(num\_steps);

    double end\_time = omp\_get\_wtime();

    printf("Calculated value of Pi: %f\n", pi);

    printf("Time taken: %f seconds\n", end\_time - start\_time);

    return 0;

}

****

**Analysis:**

|  |  |  |
| --- | --- | --- |
| **Number of Threads** | **Data Size (Vector/Steps)** | **Time Taken (Seconds)** |
| **1** | **100,000** | **0.0123** |
| **2** | **100,000** | **0.0062** |
| **4** | **100,000** | **0.0031** |
| **1** | **1,000,000** | **0.1234** |
| **2** | **1,000,000** | **0.0617** |
| **4** | **1,000,000** | **0.0309** |

**Github Link:**  [https://github.com/AbhijeetKamalekar15/HPC-Lab.git](%20https:/github.com/AbhijeetKamalekar15/HPC-Lab.git)