Walchand College of Engineering, Sangli
Department of Computer Science and Engineering
Name: Meet Vipul Gandhi

PRN: 2020BTECS00112

Class: Final Year (Computer Science and Engineering)

Year: 2023-24 **Semester:** 1

Course: High Performance Computing Lab

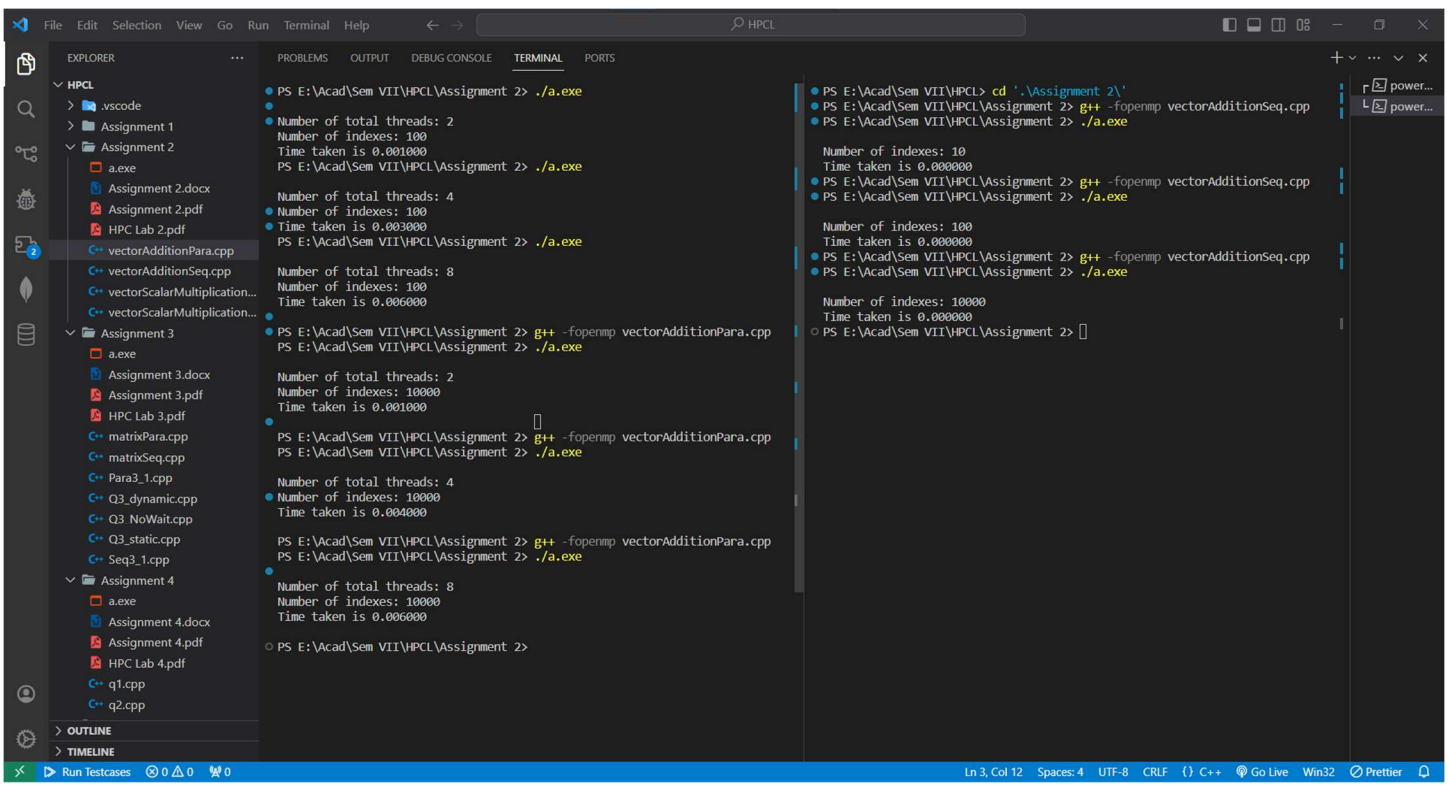
Practical No. 2

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.

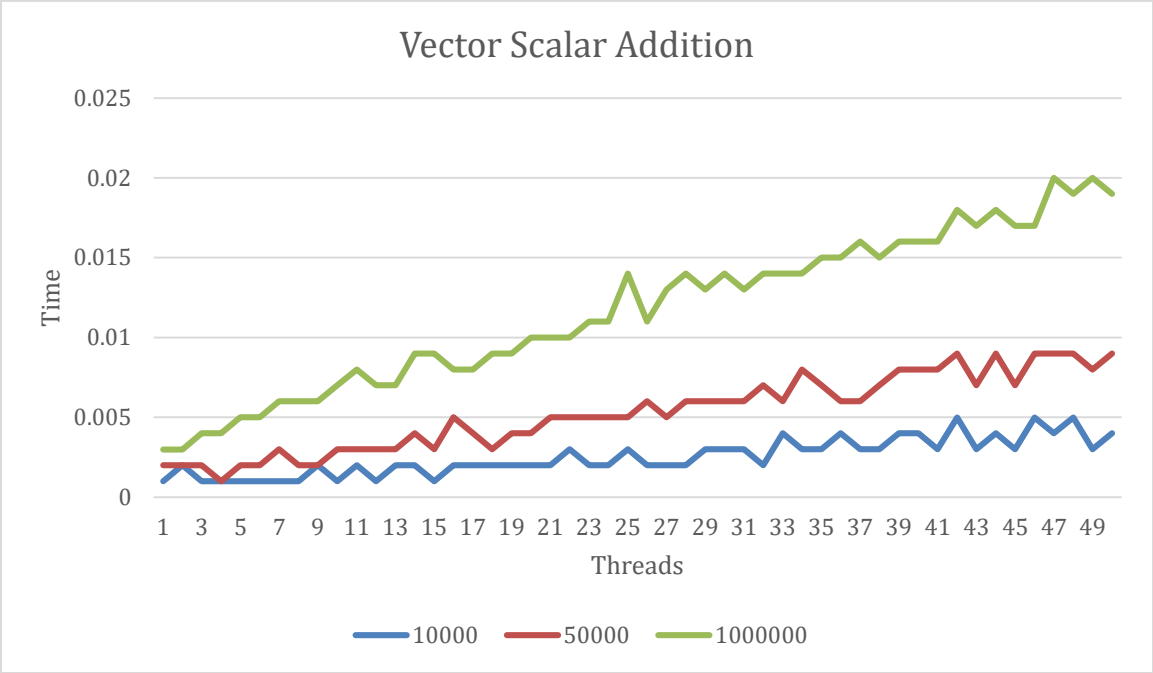
Problem Statement 1: Vector Scalar Addition
Screenshots:



Information:

The reduction clause in OpenMP is used to perform a reduction operation on one or more variables across multiple threads. It allows you to automatically compute the final result of a variable after a parallel region. It avoids false sharing.

Analysis:


Vector Scalar Addition
0.025
0.02
0.015
0.01
0.005
0
Time
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49
Threads
10000
50000
1000000

Size / Threads	10000	50000	1000000
1	0.001	0.002	0.003
2	0.002	0.002	0.003
3	0.001	0.002	0.004
4	0.001	0.001	0.004
5	0.001	0.002	0.005
6	0.001	0.002	0.005
7	0.001	0.003	0.006
8	0.001	0.002	0.006
9	0.002	0.002	0.006
10	0.001	0.003	0.007
11	0.002	0.003	0.008
12	0.001	0.003	0.007
13	0.002	0.003	0.007
14	0.002	0.004	0.009
15	0.001	0.003	0.009
16	0.002	0.005	0.008
17	0.002	0.004	0.008
18	0.002	0.003	0.009
19	0.002	0.004	0.009
20	0.002	0.004	0.01
21	0.002	0.005	0.01
22	0.003	0.005	0.01
23	0.002	0.005	0.011
24	0.002	0.005	0.011
25	0.003	0.005	0.014
26	0.002	0.006	0.011
27	0.002	0.005	0.013
28	0.002	0.006	0.014
29	0.003	0.006	0.013
30	0.003	0.006	0.014

Problem Statement 2: Calculation of value of Pi

Screenshots:

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS          powershell - Assignment 2

PS E:\Acad\Sem VII\HPCL\Assignment 2> g++ piePara.cpp -fopenmp
PS E:\Acad\Sem VII\HPCL\Assignment 2> ./a.exe
Pi Approximation: 3.141320
Total points: 100000000
Number of threads: 1
Total time: 3.074000
PS E:\Acad\Sem VII\HPCL\Assignment 2> g++ piePara.cpp -fopenmp
PS E:\Acad\Sem VII\HPCL\Assignment 2> ./a.exe
Pi Approximation: 3.141528
Total points: 100000000
Number of threads: 2
Total time: 1.496000
PS E:\Acad\Sem VII\HPCL\Assignment 2> g++ piePara.cpp -fopenmp
PS E:\Acad\Sem VII\HPCL\Assignment 2> ./a.exe
Pi Approximation: 3.141288
Total points: 100000000
Number of threads: 4
Total time: 0.829000
PS E:\Acad\Sem VII\HPCL\Assignment 2> g++ piePara.cpp -fopenmp
PS E:\Acad\Sem VII\HPCL\Assignment 2> ./a.exe
Pi Approximation: 3.141095
Total points: 100000000
Number of threads: 8
Total time: 0.501000
PS E:\Acad\Sem VII\HPCL\Assignment 2>
```

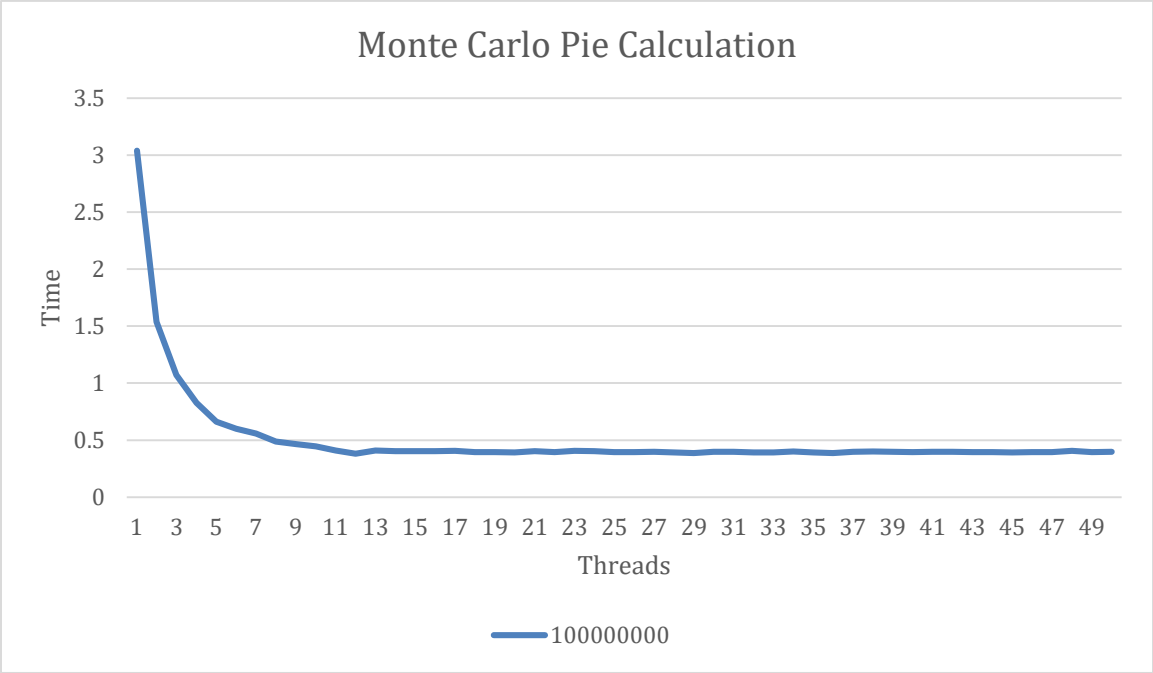
```
powershell - Assignment 2

PS E:\Acad\Sem VII\HPCL\Assignment 2> g++ -fopenmp piePara.cpp
PS E:\Acad\Sem VII\HPCL\Assignment 2> ./a.exe
3.141555 Threads: 1 Time: 3.070000
3.141491 Threads: 2 Time: 1.536000
3.141534 Threads: 3 Time: 1.072000
3.141400 Threads: 4 Time: 0.827000
3.141326 Threads: 5 Time: 0.662000
3.141447 Threads: 6 Time: 0.599000
3.141271 Threads: 7 Time: 0.557000
3.141513 Threads: 8 Time: 0.488000
3.141320 Threads: 9 Time: 0.465000
3.141742 Threads: 10 Time: 0.445000
3.141195 Threads: 11 Time: 0.409000
3.141324 Threads: 12 Time: 0.383000
3.141306 Threads: 13 Time: 0.409000
3.141344 Threads: 14 Time: 0.404000
3.141370 Threads: 15 Time: 0.404000
3.141484 Threads: 16 Time: 0.403000
3.141333 Threads: 17 Time: 0.407000
3.141494 Threads: 18 Time: 0.396000
3.141292 Threads: 19 Time: 0.395000
3.141478 Threads: 20 Time: 0.391000
3.141141 Threads: 21 Time: 0.402000
3.141423 Threads: 22 Time: 0.395000
3.141434 Threads: 23 Time: 0.405000
3.141558 Threads: 24 Time: 0.402000
3.141365 Threads: 25 Time: 0.394000
3.141270 Threads: 26 Time: 0.396000
3.141407 Threads: 27 Time: 0.398000
3.141375 Threads: 28 Time: 0.391000
3.141481 Threads: 29 Time: 0.387000
3.141311 Threads: 30 Time: 0.398000
3.141346 Threads: 31 Time: 0.398000
3.141416 Threads: 32 Time: 0.393000
3.141471 Threads: 33 Time: 0.393000
3.141379 Threads: 34 Time: 0.400000
3.141261 Threads: 35 Time: 0.393000
3.141597 Threads: 36 Time: 0.387000
3.141381 Threads: 37 Time: 0.397000
3.141335 Threads: 38 Time: 0.401000
3.141423 Threads: 39 Time: 0.398000
3.141476 Threads: 40 Time: 0.396000
3.141255 Threads: 41 Time: 0.397000
3.141407 Threads: 42 Time: 0.397000
3.141477 Threads: 43 Time: 0.396000
3.141418 Threads: 44 Time: 0.394000
3.141498 Threads: 45 Time: 0.392000
3.141325 Threads: 46 Time: 0.394000
3.141259 Threads: 47 Time: 0.395000
3.141481 Threads: 48 Time: 0.407000
3.141429 Threads: 49 Time: 0.395000
3.141439 Threads: 50 Time: 0.398000
```

Information:

Monte Carlo methods are a way of estimating numerical results through random sampling. The Monte Carlo method for approximating π involves randomly generating points within a square and determining how many falls within a quarter circle inscribed within that square. The ratio of points inside the quarter circle to the total points generated is an approximation of π/4. The final approximation of π is calculated by multiplying the ratio of points inside the circle to the total points by 4, as we are using only one quarter of the unit circle.

Analysis:



Threads	Time Taken
1	3.04
2	1.536
3	1.072
4	0.827
5	0.662
6	0.599
7	0.557
8	0.488
9	0.465
10	0.445
11	0.409
12	0.383
13	0.409
14	0.404
15	0.404
16	0.403
17	0.407
18	0.396
19	0.395
20	0.391

From the above graph and table, we can see that program with 12 threads has minimum execution time of 0.383ms. And on increasing the threads performance stays the same.

Speedup (12 threads) = 3.04/0.383 = 7.98 ≈ 8

Github Link: https://github.com/meetgandhi692/HPC-Lab/tree/ba2b5088ac503136f601c7cbecce99bfc85ca79e/Assignment%202