Walchand College of Engineering, Sangli
Department of Computer Science and Engineering

Class: Final Year (Computer Science and Engineering)

Year: 2022-23 **Semester:** 7

Course: High Performance Computing Lab

Practical No. 01

Exam Seat No:

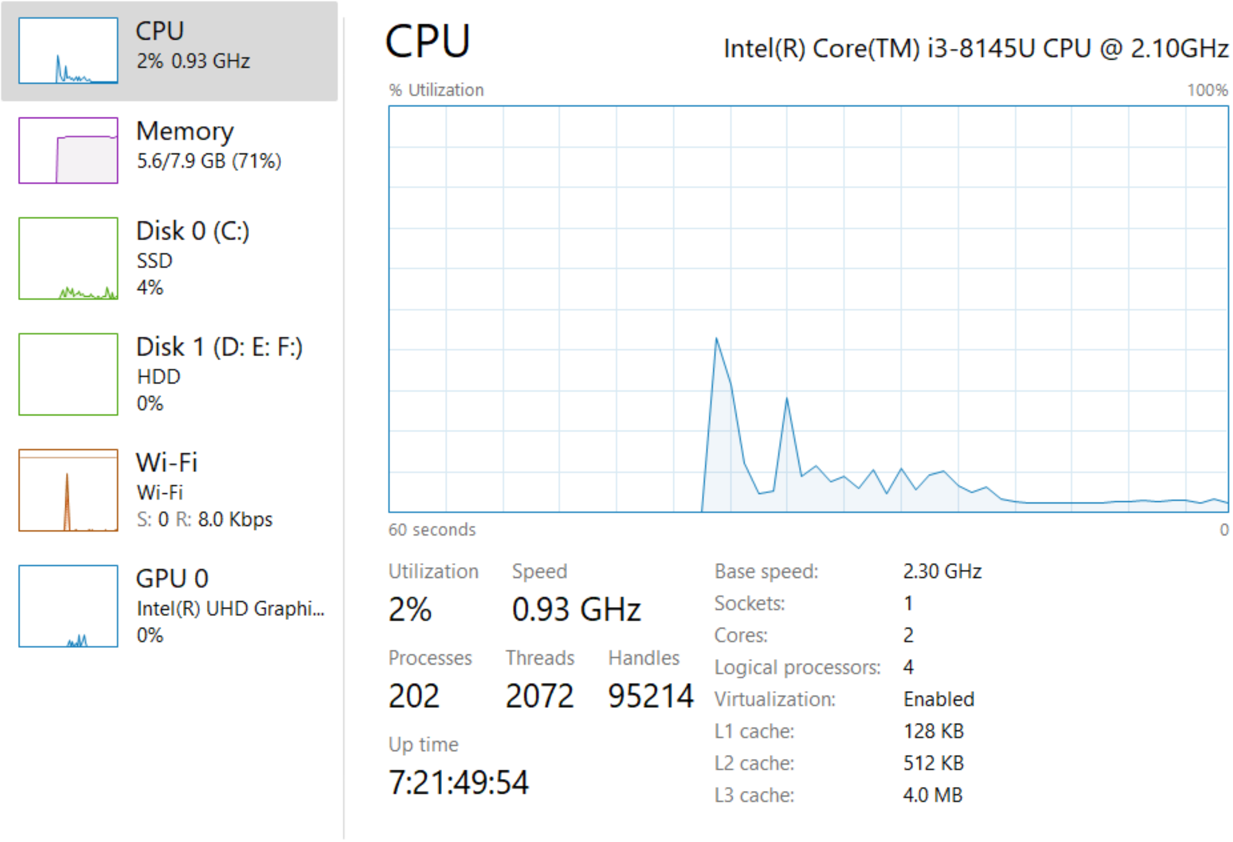
1. 2019BTECS00033 --- Teknath K Jha

Title of practical:

openMp program for :

1) **Hello World**
2) **Squares from 1 to 100**

MY SYSTEM CONFIGURATION :



CPU
2% 0.93 GHz

Memory
5.6/7.9 GB (71%)

Disk 0 (C:)
SSD
4%

Disk 1 (D: E: F:)
HDD
0%

Wi-Fi
Wi-Fi
S: 0 R: 8.0 Kbps

GPU 0
Intel(R) UHD Graphi...
0%

CPU

Intel(R) Core(TM) i3-8145U CPU @ 2.10GHz

Utilization
2%

Speed
0.93 GHz

Processes
202

Threads
2072

Handles
95214

Up time
7:21:49:54

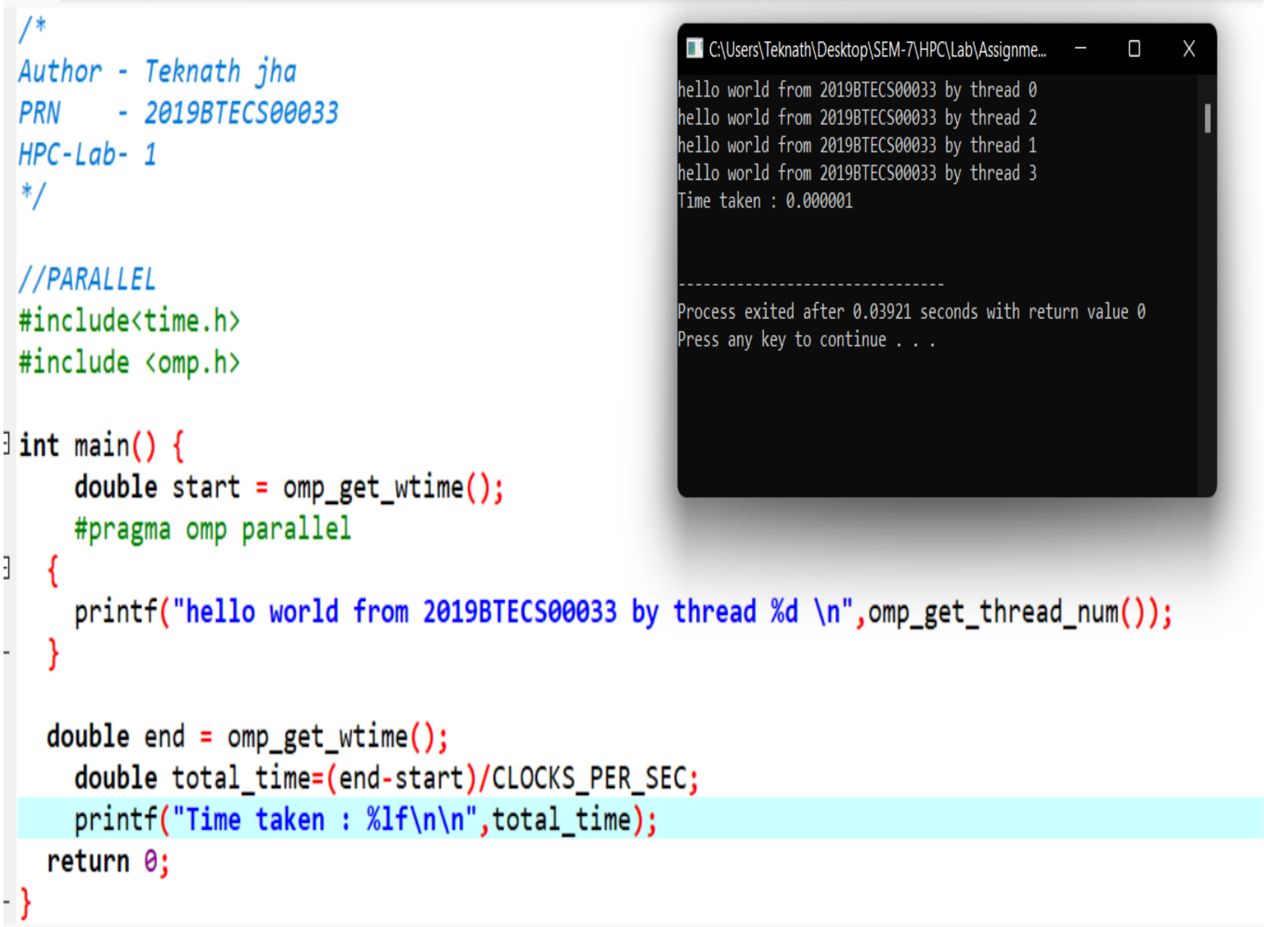
Base speed:	2.30 GHz
Sockets:	1
Cores:	2
Logical processors:	4
Virtualization:	Enabled
L1 cache:	128 KB
L2 cache:	512 KB
L3 cache:	4.0 MB

Physical Vs Software threads :

- Software threads are threads of execution managed by the operating system.
- A "hardware thread" is a physical CPU or core. So, a 4 core CPU can *genuinely* support 4 hardware threads at once - the CPU really is doing 4 things at the same time.

- One hardware thread can run many software threads.
- In modern operating systems, this is often done by time-slicing - each thread gets a few milliseconds to execute before the OS schedules another thread to run on that CPU

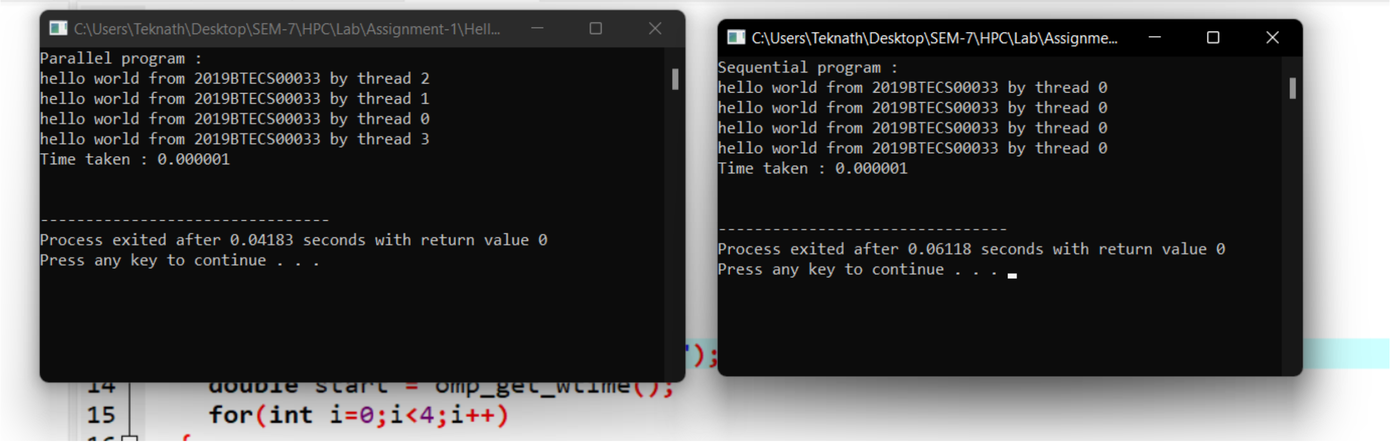
A) Problem Statement 1: Hello World :



```
/*
Author - Teknath jha
PRN    - 2019BTECS00033
HPC-Lab- 1
*/

//PARALLEL
#include<time.h>
#include <omp.h>

int main() {
    double start = omp_get_wtime();
    #pragma omp parallel
  {
    printf("hello world from 2019BTECS00033 by thread %d \n",omp_get_thread_num());
  }

  double end = omp_get_wtime();
    double total_time=(end-start)/CLOCKS_PER_SEC;
    printf("Time taken : %lf\n\n",total_time);
  return 0;
}
```

```
hello world from 2019BTECS00033 by thread 0
hello world from 2019BTECS00033 by thread 2
hello world from 2019BTECS00033 by thread 1
hello world from 2019BTECS00033 by thread 3
Time taken : 0.000001


--------------------------------
Process exited after 0.03921 seconds with return value 0
Press any key to continue . . .
```

Comparison with sequential :



C:\Users\Teknath\Desktop\SEM-7\HPC\Lab\Assignment-1\Hell...

```
Parallel program :
hello world from 2019BTECS00033 by thread 2
hello world from 2019BTECS00033 by thread 1
hello world from 2019BTECS00033 by thread 0
hello world from 2019BTECS00033 by thread 3
Time taken : 0.000001


--------------------------------
Process exited after 0.04183 seconds with return value 0
Press any key to continue . . .
```

C:\Users\Teknath\Desktop\SEM-7\HPC\Lab\Assignme...

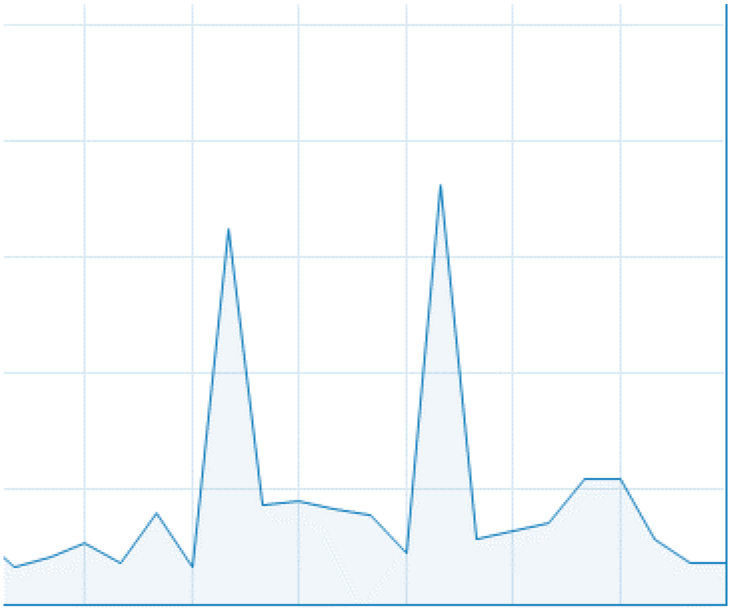
```
Sequential program :
hello world from 2019BTECS00033 by thread 0
hello world from 2019BTECS00033 by thread 0
hello world from 2019BTECS00033 by thread 0
hello world from 2019BTECS00033 by thread 0
Time taken : 0.000001


--------------------------------
Process exited after 0.06118 seconds with return value 0
Press any key to continue . . .
```

```
    double start = omp_get_wtime();
    for(int i=0;i<4;i++)
```

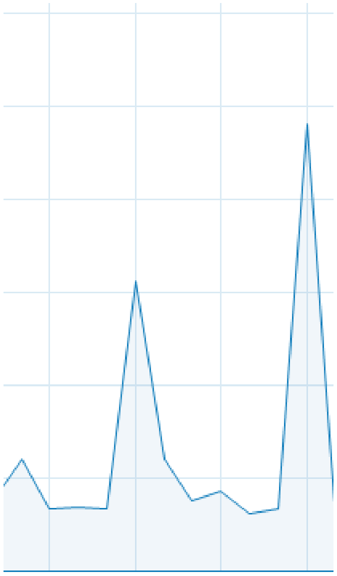
In below images 1st peak is of sequential and later is of parallel program :

Images from CPU Utilization Task Manager :

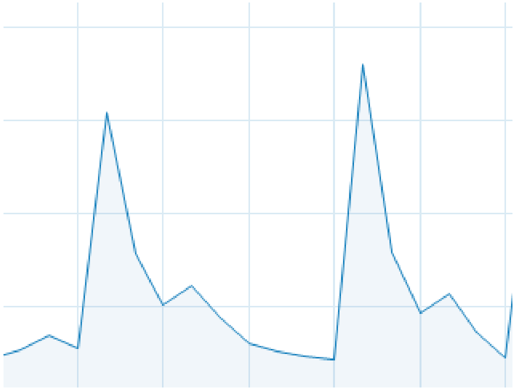
1)



2)



3)



<u>Conclusion</u> **: my sequential program uses less CPU and parallel program uses more UPU for same program and same number of instruction Sets.**

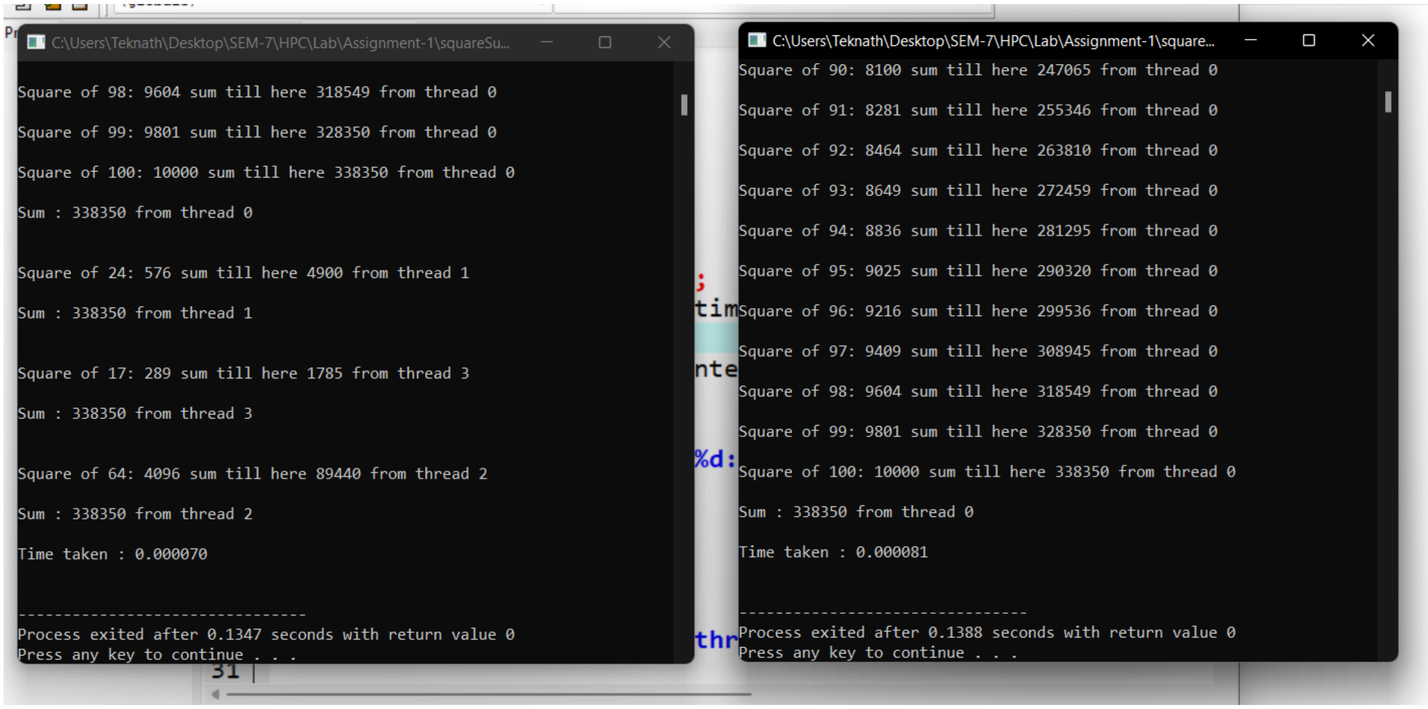
Although time calculation is negligible as it is small program .

Problem Statement 2: Squares from 1 to 100

Screenshot #:
Information #:

Squares from 1 to 100

Output Screenshots :


C:\Users\Teknath\Desktop\SEM-7\HPC\Lab\Assignment-1\squareSu...
C:\Users\Teknath\Desktop\SEM-7\HPC\Lab\Assignment-1\square...

```
Square of 98: 9604 sum till here 318549 from thread 0

Square of 99: 9801 sum till here 328350 from thread 0

Square of 100: 10000 sum till here 338350 from thread 0

Sum : 338350 from thread 0


Square of 24: 576 sum till here 4900 from thread 1

Sum : 338350 from thread 1


Square of 17: 289 sum till here 1785 from thread 3

Sum : 338350 from thread 3


Square of 64: 4096 sum till here 89440 from thread 2

Sum : 338350 from thread 2

Time taken : 0.000070


--------------------------------
Process exited after 0.1347 seconds with return value 0
Press any key to continue . . .
```

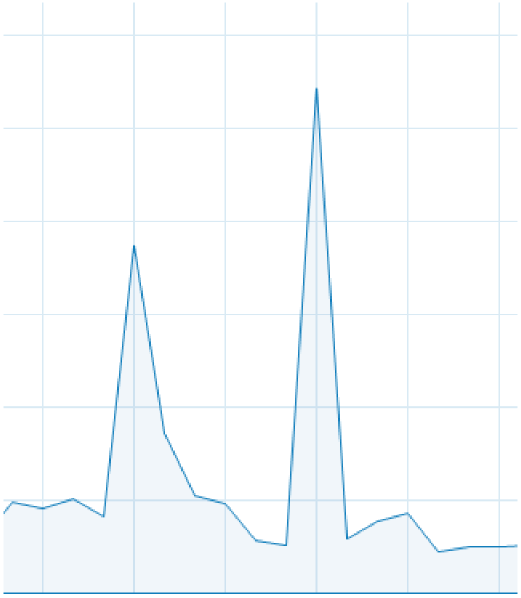
```
Square of 90: 8100 sum till here 247065 from thread 0

Square of 91: 8281 sum till here 255346 from thread 0

Square of 92: 8464 sum till here 263810 from thread 0

Square of 93: 8649 sum till here 272459 from thread 0

Square of 94: 8836 sum till here 281295 from thread 0

Square of 95: 9025 sum till here 290320 from thread 0

Square of 96: 9216 sum till here 299536 from thread 0

Square of 97: 9409 sum till here 308945 from thread 0

Square of 98: 9604 sum till here 318549 from thread 0

Square of 99: 9801 sum till here 328350 from thread 0

Square of 100: 10000 sum till here 338350 from thread 0

Sum : 338350 from thread 0

Time taken : 0.000081


--------------------------------
Process exited after 0.1388 seconds with return value 0
Press any key to continue . . .
```

Parallel :70 **Sequential: 81**

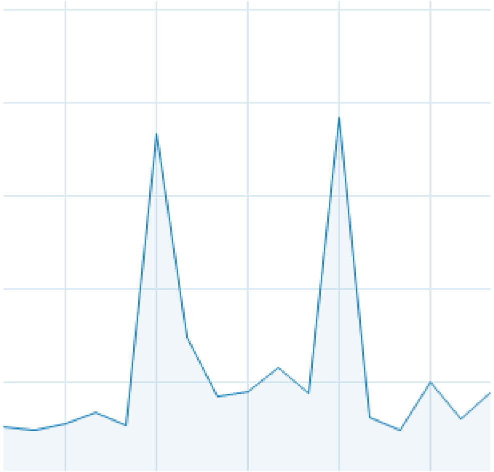
So here parallel is faster than sequential.

In below images 1st peak is of sequential and later is of parallel program :

CPU Graphs :

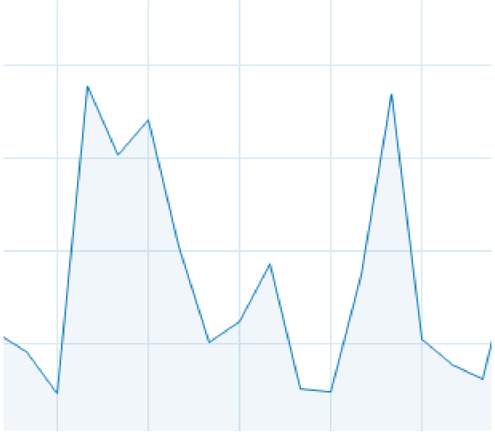


1)



2)

3)



Here most of time width of sequential is more than parallel which shows time difference.

Conclusion :

In execution : sequential taken 0.000081 while parallel taken 0.000070 which is considerable difference , further observation of CPU cycles also proves this that parallel is faster than sequential .

Github Link: https://github.com/Teknath-jha/HPC-LAB-2019BTECS00033/tree/main/Assignment-1