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

Class: Final Year (Computer Science and Engineering)

Year: 2021-22 **Semester:** 1

Course: High Performance Computing lab

ESE Exam

22/11/2021 01.00 PM – 04.00 PM

Exam Seat No: 2018BTECS00044

Name: Aniruddha Sanjay Palekar
Exam Seat Number: 2018BTECS00044

Problem Statement 1

Statement: Find sum of 1000 numbers using OpenMP.

Screenshot 1:

```
TERMINAL    PROBLEMS    OUTPUT    DEBUG CONSOLE

anya@LAPTOP-F25DMQQC:/mnt/d/HPC/ESE_lab$ gcc -o myexe -fopenmp sum.c
anya@LAPTOP-F25DMQQC:/mnt/d/HPC/ESE_lab$ ./myexe
Current thread is 6 and SUM 65050
Current thread is 3 and SUM 35050
Current thread is 8 and SUM 85050
Current thread is 5 and SUM 55050
Current thread is 7 and SUM 75050
Current thread is 4 and SUM 45050
Current thread is 2 and SUM 25050
Current thread is 1 and SUM 15050
Current thread is 0 and SUM 5050
Current thread is 9 and SUM 95050
The sum of first 1000 numbers is: 500500
anya@LAPTOP-F25DMQQC:/mnt/d/HPC/ESE_lab$
```

Information 1: here sum of 1000 numbers is calculated using reduce function.

Problem Statement 2

Statement: Implement MPI program to send integer array from one process to other.

Screenshot 1:

```
TERMINAL    PROBLEMS    OUTPUT    DEBUG CONSOLE

anya@LAPTOP-F25DMQQC:/mnt/d/HPC/ESE_lab$ mpicc -o myexe send_array.cpp
anya@LAPTOP-F25DMQQC:/mnt/d/HPC/ESE_lab$ mpirun -np 2 ./myexe
Int Array Received from Process 0 to 1
array: 1        2       3       4       5       6       7       8       9       10
```

Information 1: here array is sent from 1 process to another using MPI_Send and MPI_Recv.

Problem Statement 3

Statement: Write a CUDA C program to demonstrate the use of different GPU memories.

- **Use of private memory.**
- **Use of shared memory.**
- **Use of global memory**

Screenshot 1:

```
Overwriting cuda1.cu

[28] !nvcc -o cuda1 cuda1.cu
0s

[29] !./cuda1
0s

 first matrix of size 2*3
1       2       3       4       5       6
 second matrix of size 3*2
1       2       3       4       5       6
Product of two matrices:
22      28
49      64
```

Information 1: here we have done program for matrix multiplication using cuda where we have used differrent memory types like global,shared and local.

Screenshot 2:

```
!nvprof ./cuda1

*This is an example of shared memory in cuda**
 first matrix of size 2*3
1       2       3       4       5       6
 second matrix of size 3*2
==17396== NVPROF is profiling process 17396, command: ./cuda1
==17396== Warning: Profiling results might be incorrect with current version of nvcc compiler used to compile cuda app. Compile with nvcc compiler 9.0 or later version
1       2       3       4       5       6
 Product of two matrices:
 22      28
49      64
==17396== Profiling application: ./cuda1
==17396== Profiling result:
            Type  Time(%)      Time     Calls       Avg       Min       Max  Name
 GPU activities:   41.87%  4.4480us         1  4.4480us  4.4480us  4.4480us  matrixproduct(int*, int*, int*)
                   35.84%  3.8080us         2  1.9040us  1.5360us  2.2720us  [CUDA memcpy HtoD]
                   22.29%  2.3680us         1  2.3680us  2.3680us  2.3680us  [CUDA memcpy DtoH]
      API calls:   99.54%  257.65ms         3  85.884ms  2.3340us  257.65ms  cudaMalloc
                    0.20%  511.82us         1  511.82us  511.82us  511.82us  cuDeviceTotalMem
                    0.09%  220.77us         1  220.77us  220.77us  220.77us  cudaLaunchKernel
                    0.08%  213.04us        96  2.2190us     140ns  76.380us  cuDeviceGetAttribute
                    0.06%  164.40us         3  54.799us  5.6000us  136.74us  cudaFree
                    0.02%  56.706us         3  18.902us  11.201us  27.616us  cudaMemcpy
                    0.01%  25.728us         1  25.728us  25.728us  25.728us  cuDeviceGetName
                    0.00%  7.4360us         1  7.4360us  7.4360us  7.4360us  cuDeviceGetPCIBusId
                    0.00%  1.7820us         3     594ns     188ns     977ns  cuDeviceGetCount
                    0.00%  1.4180us         2     709ns     337ns  1.0810us  cuDeviceGet
```

Information 2: here is the profiling og cuda code.

Technologies Used: Openmp, MPI,WSL,CUDA,Google Colab,VScode.
GitHub Link:https://github.com/aniruddhapalekar/HPC/tree/main/HPCESE