**Session 2025-2026**

|  |  |
| --- | --- |
| **Vision:** To help businesses uncover crucial  insights | **Mission:** To be a good data scientist |

**Program Educational Objectives of the program (PEO):** (broad statements that describe the professional and career accomplishments)

|  |  |  |  |
| --- | --- | --- | --- |
| PEO1 | **Preparation** | **P: Preparation** | **Pep-CL abbreviation**  **pronounce as Pep-si-lL easy to recall** |
| PEO2 | **Core Competence** | **E: Environment (Learning Environment)** |
| PEO3 | **Breadth** | **P: Professionalism** |
| PEO4 | **Professionalism** | **C: Core Competence** |
| PEO5 | **Learning Environment** | **L: Breadth (Learning in diverse areas)** |

**Program Outcomes (PO):** 1. Understand and Apply Parallel Programming Concepts

2. Analyse and Improve Program Performance.

3. Demonstrate Practical Skills in HPC Tools and Environments.

**Keywords of POs:**

Engineering knowledge, Problem analysis, Design/development of solutions, Conduct Investigations of Complex Problems, Engineering Tool Usage, The Engineer and The World, Ethics, Individual and Collaborative Team work, Communication, Project Management and Finance, Life-Long Learning

**PSO Keywords:** Cutting edge technologies, Research

“I am an engineer, and I know how to apply engineering knowledge to investigate, analyse and design solutions to complex problems using tools for entire world following all ethics in a collaborative way with proper management skills throughout my life.” *to contribute to the development of cutting-edge technologies and Research*.

**Integrity:** I will adhere to the Laboratory Code of Conduct and ethics in its entirety.

**Name and Signature of Student and Date**

Shreyas Chaurey – 24/10/2025

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Session** | **2025-26 (ODD)** | | **Course Name** | **HPC Lab** | |
| **Semester** | **7** | | **Course Code** | 22ADS706 | |
| **Roll No** | 61 | | **Name of Student** | Shreyas Chaurey | |
|  |  | |  |  |  |
| Practical Number | | 7 | | | |
| Course Outcome | | 1. Understand and Apply Parallel Programming Concepts 2. Analyse and Improve Program Performance | | | |
| Aim | | Hybrid Programming with MPI + OpenMP Practical | | | |
| Problem Definition | | Hybrid Programming with MPI + OpenMP Practical | | | |
| Theory  (100 words) | | **Requirements**   * **Software:**   + Linux OS (Ubuntu/RedHat recommended)   + MPI library (OpenMPI / MPICH)   + GCC compiler with OpenMP support * **Hardware:**   + Multi-core processor   + Optional: Cluster with multiple nodes for full MPI execution  1. **MPI (Message Passing Interface)**    * Used for communication between processes in a distributed memory system.    * Each process has its own address space. 2. **OpenMP (Open Multi-Processing)**    * Used for parallelism within a shared memory node.    * Allows multi-threading using #pragma omp parallel. 3. **Hybrid Programming**    * Combines MPI across nodes and OpenMP within nodes.    * Reduces communication overhead and improves parallel efficiency.   **Algorithm**   1. Initialize MPI and get rank and size. 2. Distribute rows of the matrix A among MPI processes. 3. Each process computes its local result using OpenMP threads. 4. MPI\_Reduce is used to gather results to the master process. 5. Master process prints the final result.   **Steps for execution**  **Step 1** - Compile: mpicc -fopenmp hybrid\_mpi\_openmp.c -o hybrid\_mpi\_openmp  **Step 2** - Execute (using 2 MPI processes, adjust threads with OMP\_NUM\_THREADS): export OMP\_NUM\_THREADS=4    # Set number of OpenMP threads per process  mpirun -np 2 ./hybrid\_mpi\_openmp | | | |
| Code: | | #include <stdio.h>  #include <stdlib.h>  #include <mpi.h>  #include <omp.h>  #define N 8  // Size of matrix and vector  int main(int argc, char\* argv[]) {      int rank, size;      MPI\_Init(&argc, &argv);      MPI\_Comm\_rank(MPI\_COMM\_WORLD, &rank);      MPI\_Comm\_size(MPI\_COMM\_WORLD, &size);      int rows\_per\_proc = N / size;      double A[rows\_per\_proc][N];      double x[N];      double y\_local[rows\_per\_proc];      double y[N];      // Initialize vector x and matrix A      if(rank == 0) {          for(int i = 0; i < N; i++)              x[i] = i + 1;  // Example vector: 1,2,3...      }      MPI\_Bcast(x, N, MPI\_DOUBLE, 0, MPI\_COMM\_WORLD); // Broadcast vector to all processes      // Initialize local part of matrix A      for(int i = 0; i < rows\_per\_proc; i++) {          for(int j = 0; j < N; j++) {              A[i][j] = (rank \* rows\_per\_proc + i + 1) \* (j + 1);          }      }      // Parallel computation using OpenMP      #pragma omp parallel for      for(int i = 0; i < rows\_per\_proc; i++) {          y\_local[i] = 0.0;          for(int j = 0; j < N; j++) {              y\_local[i] += A[i][j] \* x[j];          }      }      // Gather results to root process      MPI\_Gather(y\_local, rows\_per\_proc, MPI\_DOUBLE, y, rows\_per\_proc, MPI\_DOUBLE, 0, MPI\_COMM\_WORLD);      // Print result in master process      if(rank == 0) {          printf("Result vector y:\n");          for(int i = 0; i < N; i++) {              printf("%lf ", y[i]);          }          printf("\n");      }      MPI\_Finalize();      return 0;  } | | | |
| Output | | A screenshot of a computer program  AI-generated content may be incorrect.  A screenshot of a computer  AI-generated content may be incorrect. | | | |
| Output Analysis | | The program executes successfully and gives us the resultant vector y as an output. | | | |
| Link of student Github profile where lab assignment has been uploaded | |  | | | |
| Conclusion | | The experiment successfully compiled and ran a hybrid MPI/OpenMP parallel program using 2 MPI processes. The program executed correctly, as confirmed by the output of the 8-element result vector y demonstrating that the system and the parallel code are properly configured for hybrid high-performance computing. | | | |
| Plag Report (Similarity index < 12%) | |  | | | |
| Date | | 24/10/2025 | | | |