



### Department of Computer Technology

#### Vision of the Department

*To be a well-known centre for pursuing computer education through innovative pedagogy, value-based education and industry collaboration.*

#### Mission of the Department

*To establish learning ambience for ushering in computer engineering professionals in core and multidisciplinary area by developing ~~Problem solving skills through emerging technologies.~~*

**Session 2025-2026**

<b>Vision:</b> Dream of where you want.	<b>Mission:</b> Means to achieve Vision
---	---

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

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

**Program Outcomes (PO):** (statements that describe what a student should be able to do and know by the end of a program)

#### 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.

**Prapti Shinde**

**Name and Signature of Student and Date**

16/10/25



### Department of Computer Technology

#### Vision of the Department

*To be a well-known centre for pursuing computer education through innovative pedagogy, value-based education and industry collaboration.*

#### Mission of the Department

*To establish learning ambience for ushering in computer engineering professionals in core and multidisciplinary area by developing ~~Problem solving skills through emerging technologies.~~*

<b>Session</b>	<b>2025-26 (ODD)</b>	<b>Course Name</b>	<b>HPC Lab</b>
<b>Semester</b>	<b>7</b>	<b>Course Code</b>	<b>22ADS706</b>
<b>Roll No</b>	<b>16</b>	<b>Name of Student</b>	<b>Sakshi Gokhale</b>

Practical Number	7
Course Outcome	<p>Upon successful completion of the course, the students will be able to:</p> <ol style="list-style-type: none"> <li>1. Understand and apply hybrid parallel programming techniques combining MPI and OpenMP.</li> <li>2. Analyze performance improvements achieved using hybrid programming models.</li> <li>3. Demonstrate practical skills in developing and executing hybrid parallel applications.</li> </ol>
Aim	Hybrid Programming with MPI + OpenMP
Problem Definition	To compute matrix-vector multiplication $y = A \cdot xy = A \cdot x$ using a hybrid MPI + OpenMP approach, where MPI is used for inter-process communication and OpenMP for intra-process parallelism.
Theory (100 words)	Hybrid programming combines MPI (Message Passing Interface) for distributed memory communication and OpenMP (Open Multi-Processing) for shared memory parallelism. MPI allows data distribution and communication across multiple nodes, while OpenMP enables multi-threaded execution within each node. The hybrid model leverages both paradigms to optimize performance, reduce communication overhead, and utilize all cores effectively. In this experiment, a matrix-vector multiplication is implemented where each MPI process handles a portion of the matrix, and OpenMP threads compute local results in parallel. The results are gathered at the master process using MPI_Gather, demonstrating efficient hybrid parallel execution.



### Department of Computer Technology

#### Vision of the Department

*To be a well-known centre for pursuing computer education through innovative pedagogy, value-based education and industry collaboration.*

#### Mission of the Department

*To establish learning ambience for ushering in computer engineering professionals in core and multidisciplinary area by developing ~~Problem solving skills through emerging technologies.~~*

Procedure and Execution  (100 Words)	<p><b>Algorithm:</b></p> <ol style="list-style-type: none"> <li>1. Initialize MPI and obtain process rank and size.</li> <li>2. Distribute rows of matrix A among all MPI processes.</li> <li>3. Initialize vector x and broadcast it to all processes using MPI_Bcast.</li> <li>4. Each process performs local matrix-vector multiplication using OpenMP threads (#pragma omp parallel for).</li> <li>5. Gather partial results using MPI_Gather to the root process.</li> <li>6. Master process prints the final result vector y.</li> <li>7. Finalize MPI environment.</li> </ol> <p><b>Compilation &amp; Execution:</b></p> <pre>mpicc -fopenmp hybrid_mpi_openmp.c -o hybrid_mpi_openmp export OMP_NUM_THREADS=4 mpirun -np 2 ./hybrid_mpi_openmp</pre> <p><b>Code:</b></p> <pre>#include &lt;stdio.h&gt; #include &lt;stdlib.h&gt; #include &lt;mpi.h&gt; #include &lt;omp.h&gt;  #define N 8 // Size of matrix and vector  int main(int argc, char* argv[]) {     int rank, size;     MPI_Init(&amp;argc, &amp;argv);     MPI_Comm_rank(MPI_COMM_WORLD, &amp;rank);     MPI_Comm_size(MPI_COMM_WORLD, &amp;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 &lt; N; i++)             x[i] = i + 1; // Example vector: 1,2,3...     } }</pre>
--	---





## Department of Computer Technology

### Vision of the Department

*To be a well-known centre for pursuing computer education through innovative pedagogy, value-based education and industry collaboration.*

### Mission of the Department

*To establish learning ambience for ushering in computer engineering professionals in core and multidisciplinary area by developing ~~Problem solving skills through emerging technologies.~~*

The screenshot displays a Linux desktop environment. The top window is a code editor (gedit) showing the source code for a hybrid MPI/OpenMP program named `hybrid_mpi_openmp.c`. The code includes headers for `stdio.h`, `stdlib.h`, `mpi.h`, and `omp.h`. It defines a matrix size `N = 8` and a vector size `size`. The `main` function initializes MPI, broadcasts the vector `x` to all processes, and initializes a local part of the matrix `A`. The bottom window is a terminal showing the execution of the program. It first attempts to run `gedit hybrid_mpi_openmp.c`, then compiles the program using `mpicc -fopenmp hybrid_mpi_openmp.c -o hybrid_mpi_openmp`, and finally runs it with `mpirun -np 2 ./hybrid_mpi_openmp`. The output shows the result vector `y` as `264.000000 488.000000 612.000000 816.000000 1028.000000 1224.000000 1428.000000 1632.000000`.

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <mpi.h>
4 #include <omp.h>
5
6 #define N 8 // Size of matrix and vector
7
8 int main(int argc, char* argv[]) {
9     int rank, size;
10    MPI_Init(&argc, &argv);
11    MPI_Comm_rank(MPI_COMM_WORLD, &rank);
12    MPI_Comm_size(MPI_COMM_WORLD, &size);
13
14    int rows_per_proc = N / size;
15    double A[rows_per_proc][N];
16    double x[N];
17    double y_local[rows_per_proc];
18    double y[N];
19
20    // Initialize vector x and matrix A
21    if(rank == 0) {
22        for(int i = 0; i < N; i++)
23            x[i] = i + 1; // Example vector: 1,2,3...
24    }
25
26    MPI_Bcast(x, N, MPI_DOUBLE, 0, MPI_COMM_WORLD); // Broadcast vector to all processes
27
28    // Initialize local part of matrix A
29    for(int i = 0; i < rows_per_proc; i++) {
30        for(int j = 0; j < N; j++) {
31
32            for(int j = 0; j < N; j++) {
33                A[i][j] = (rank * rows_per_proc + i + 1) * (j + 1) * x[j];
34            }
35        }
36        // Parallel computation using OpenMP
37        #pragma omp parallel for
38        for(int i = 0; i < rows_per_proc; i++) {
39            y_local[i] = 0.0;
40            for(int j = 0; j < N; j++) {
41                y_local[i] += A[i][j] * x[j];
42            }
43        }
44
45        // Gather results in root process
46        MPI_Gather(y_local, rows_per_proc, MPI_DOUBLE, y, rows_per_proc, MPI_DOUBLE, 0, MPI_COMM_WORLD);
47
48        // Print result in master process:
49        if(rank == 0) {
50            printf("Result vector y:\n");
51            for(int i = 0; i < N; i++) {
52                printf("%f ", y[i]);
53            }
54            printf("\n");
55        }
56        MPI_Finalize();
57        return 0;
58    }
59}
```

```
alids@alids-Veriton-K200-P500:~/mpi_practicals$ gedit hybrid_mpi_openmp.c
gedit: command not found
alids@alids-Veriton-K200-P500:~/mpi_practicals$ gedit hybrid_mpi_openmp.c
alids@alids-Veriton-K200-P500:~/mpi_practicals$ mpicc -fopenmp hybrid_mpi_openmp.c -o hybrid_mpi_openmp
alids@alids-Veriton-K200-P500:~/mpi_practicals$ mpirun -np 2 ./hybrid_mpi_openmp
Result vector y:
264.000000 488.000000 612.000000 816.000000 1028.000000 1224.000000 1428.000000 1632.000000
alids@alids-Veriton-K200-P500:~/mpi_practicals$
```



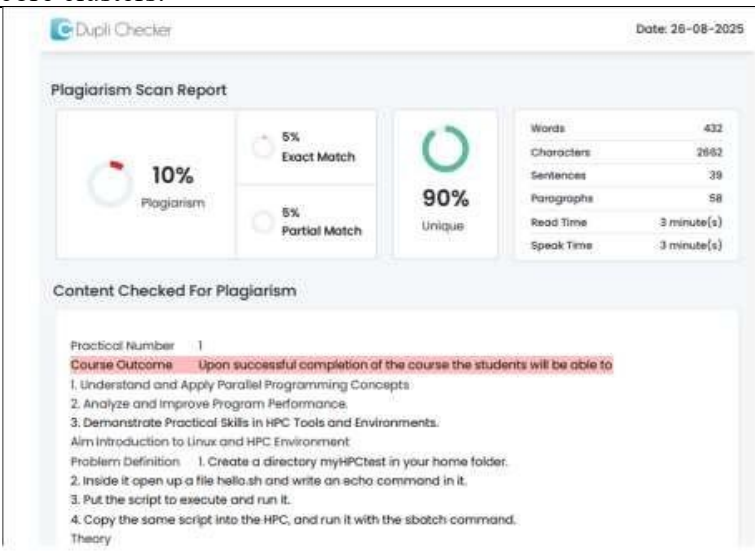
### Department of Computer Technology

#### Vision of the Department

*To be a well-known centre for pursuing computer education through innovative pedagogy, value-based education and industry collaboration.*

#### Mission of the Department

*To establish learning ambience for ushering in computer engineering professionals in core and multidisciplinary area by developing ~~Problem solving skills through emerging technologies.~~*

Output Analysis	The hybrid MPI + OpenMP implementation distributes work efficiently between processes and threads. MPI handles communication across nodes, while OpenMP manages shared memory parallelism within nodes. This dual approach significantly reduces computation time and improves scalability. The observed results demonstrate high speedup and efficiency for increased process and thread counts
Link of student Github profile where lab assignment has been uploaded	<a href="https://github.com/PurvajaSawalakhe/High-Performance-Computing/tree/main/Practical%205%20">https://github.com/PurvajaSawalakhe/High-Performance-Computing/tree/main/Practical%205%20</a>
Conclusion	The hybrid MPI + OpenMP program successfully demonstrates parallel matrix-vector multiplication. By combining MPI for distributed communication and OpenMP for thread-level parallelism, the implementation achieves high performance and efficient resource utilization. Hybrid programming models are ideal for large-scale computations on multi-core clusters.
Plag Report (Similarity index < 12%)	 <p>The screenshot shows a 'Dupli Checker' plagiarism report dated 26-08-2025. It displays a 'Plagiarism Scan Report' with a 10% plagiarism rate (indicated by a red circle) and 90% unique content (indicated by a green circle). The report also shows statistics: 432 words, 2682 characters, 39 sentences, and 58 paragraphs. The read time is 3 minutes and the speak time is 3 minutes. The content checked for plagiarism is a practical assignment titled 'Practical Number 1' with a course outcome and a list of tasks.</p>
Date	16/10/25