Parallel Computing

Code: CS32124CS	Course Type: Elective	Semester: I
Evolution Scheme (TA-MSE-ESE): 20-30-100	Total Marks: 150	L-T-P (Credits): 3-0-0 (3)

Pre-Requisites: NIL

Course Objectives:

To understand the fundamentals of parallel computing, including architectures, programming models, and performance metrics, and to design and apply parallel algorithms and techniques for solving real-world problems.

Course Outcomes: At the end of the course, the student will be able to

CO-1	Understand fundamental concepts and performance metrics of parallel computing
CO-2	Analyze different parallel architectures
CO-3	Design and implement programming models for real-world problems
CO-4	Apply parallel algorithms and computing techniques in modern applications like AI, GPU computing, and cloud systems

Course Articulation Matrix:

PO	PEO-1	PEO-2	PEO-3	PEO-4	PEO-5
CO-1	3	2	1	1	2
CO-2	3	3	2	1	2
CO-3	3	3	3	1	3
CO-4	3	3	2	2	3

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Unit-1: Fundamentals of Parallel Computing

Introduction to Parallel Computing, Definition, need, and scope, Applications in scientific computing, AI/ML, big data, Flynn's Taxonomy: SISD, SIMD, MISD, MIMD, Performance Metrics: Speedup, Efficiency, Scalability, Amdahl's Law and Gustafson's Law.

Unit-2: Parallel Architectures

Multicore and Manycore Processors, Types of Parallelism: Data Parallelism, Task Parallelism, Pipeline Parallelism, Dependence and Synchronization: Data dependency, control dependency, Parallel Computer Architectures: Shared Memory (UMA, NUMA), Distributed Memory Systems.

Unit-3: Parallel Programming Models

Memory Models, Consistency models, Parallel Programming Models: Threads (Pthreads, OpenMP), Message Passing Interface (MPI), CUDA/OpenCL basics, Synchronization Mechanisms: Locks, Semaphores, Barriers.

Unit-4: Case Studies of Parallel Algorithms

Parallel Algorithms for: Matrix multiplication, Sorting (Parallel Merge, Quick Sort), Searching, GPU Computing, Cloud and Cluster Computing basics**,** Parallel Computing for AI/ML.

Learning Resources:

Text Books:

1. M. J. Quinn, Parallel Programming in C with MPI and OpenMP. New York, NY, USA: McGraw-Hill, 2004.
2. A. Grama, A. Gupta, G. Karypis, and V. Kumar, Introduction to Parallel Computing, 2nd ed. Boston, MA, USA: Addison-Wesley, 2003.
3. K. Hwang, Advanced Computer Architecture: Parallelism, Scalability, Programmability. New York, NY, USA: McGraw-Hill, 2012.

Optional Materials: Reference Books

1. P. Pacheco, An Introduction to Parallel Programming. Burlington, MA, USA: Morgan Kaufmann, 2011.
2. B. Wilkinson and M. Allen, Parallel Programming: Techniques and Applications Using Networked Workstations and Parallel Computers, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall, 2005.
3. D. B. Kirk and W.-m. W. Hwu, Programming Massively Parallel Processors: A Hands-on Approach, 3rd ed. Cambridge, MA, USA: Morgan Kaufmann, 2016.