UNIVERSITY OF MADRAS

MASTER OF COMPUTER APPLICATIONS (MCA) DEGREE PROGRAMME

SYLLABUS WITH EFFECT FROM 2023-2024

Title of the Paper	**Parallel and Distributed Computing**		
Elective- VI Theory	II Year & IV Semester	Credit:3	535E4C

Objectives:

- To analyze the current popular distributed systems such as peer-to-peer (P2P) systems will also be analyzed.
- To learn and apply knowledge of parallel and distributed computing techniques and methodologies.
- To learn the architecture and parallel programming in graphics processing units (GPUs).
- To understand the memory hierarchy and cost-performance tradeoffs.
- To gain experience in the design, development, and performance analysis of parallel and distributed applications

Outcomes:

- Develop and apply knowledge of parallel and distributed computing techniques and methodologies.
- Apply design, development, and performance analysis of parallel and distributed applications.
- Use the application of fundamental Computer Science methods and algorithms in the development of parallel applications.
- Explain the design, testing, and performance analysis of a software system, and to be able to communicate that design to others.
- Understand the requirements for programming parallel systems and how they can be used to facilitate the programming of concurrent systems.

Unit I: Introduction to Parallel Computing: The Idea of Parallelism, Power and potential of parallelism, examining sequential and parallel programs, Scope and issues of parallel and distributed computing, Goals of parallelism, Parallelism and concurrency using multiple instruction streams.

Unit II: Parallel Architecture: Pipeline architecture, Array processor, Multi-processor architecture, Systolic architecture, Dataflow architecture, Architectural classification schemes, Memory access classification, Memory Issues: Shared vs. distributed, Symmetric multiprocessing (SMP), SIMD, Vector processing, GPU co-processing, Flynn's Taxonomy, Instruction Level support for parallel programming, Multiprocessor caches and Cache Coherence, Non-Uniform Memory Access (NUMA).

Unit III: Parallel Algorithm Design Principles and Programming: Need for communication and coordination/synchronization, Scheduling and contention, Independence and partitioning, Task-Based Decomposition, Data Parallel Decomposition, Characteristics of task and interaction, Load balancing, Data Management, parallel algorithm models, Sources of overhead in parallel

programs, Performance metrics for parallel algorithm implementations, Parallel algorithmic patterns like divide and conquer, Map and Reduce, Specific algorithms like parallel Merge Sort, Parallel graph Algorithms.

Unit IV: Architectures Of Distributed Systems - Architectural Styles – System Architectures - Architectures Versus Middleware - Self-Management In Distributed Systems - Processes - Threads - Virtualization - Clients -Servers - Communication - Remote Procedure Call - Message-Oriented Communication - Stream-Oriented Communication - Multicast Communication

Unit V: Distributed Object Based Systems - Architecture - Processes - Communication - Naming - Synchronization - Fault Tolerance - Security - Distributed System Examples - File Systems And Web Based Systems

Recommended Texts:

1. Ananth Grama, Anshul Gupta, and George Karypis, Vipin Kumar; Introduction to Parallel Computing; Addition Wesley; 2nd Edition; 2003.

2. A.S. Tanenbaum; Distributed Operating Systems; Create Space Independent Publishing Platform; 3rd edition;2017

Reference Books:

1. Introduction To Parallel Programming, Steven Brawer, Academic Press
2. Introduction To Parallel Processing, M. Sasikumar, Dinesh Shikhare and P. Ravi Prakash,PHI
3. Randy Chow, T. Johnson, Distributed Operating Systems and Algorithms, Addison Wesley
4. Ian Foster: Designing and Building Parallel Programs – Concepts and tools for Parallel Software Engineering, Pearson Publisher, 1st Edition, 2019.
5. Parallel Programming in C with MPI and OpenMP Michael J. Quinn, McGrawHill Higher Education.

Web References:

1. https://www.youtube.com/watch?v=qbQCQ0U6H0o&list=PLbMVogVj5nJQRvzENlvMKA9q70ScSRZBQ

Course Outcomes

On the successful completion of the course, students will be able to

CO1	Develop and apply knowledge of parallel and distributed computing techniques and methodologies	K1-K6
CO2	Apply design, development, and performance analysis of parallel and distributed applications	K1-K6
CO3	Use the application of fundamental Computer Science methods and algorithms in the development of parallel applications	K1-K6
CO4	Explain the design, testing, and performance analysis of a software system, and to be able to communicate that design to others	K1-K6
CO5	Understand the requirements for programming parallel systems and how they can be used to facilitate the programming of concurrent systems	K1-K6

K1- Remember, K2- Understand, K3- Apply, K4- Analyze, K5 Evaluate, K6- Create

Mapping with Programme Outcomes:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	M	S	S	L	S	M	L	M	L	L
CO2	S	S	M	L	M	L	M	L	M	S
CO3	S	S	L	M	S	S	L	M	L	M
CO4	M	L	S	M	M	L	S	L	M	S
CO5	S	S	M	S	L	M	L	M	M	L

S- Strong; M-Medium; L-Low