Parallel and Distributed Computing	Code & No:	CS 350
	Credits:	3 (3,0,1)
	Pre-requisite:	CS 312
	Co-requisite:	None
	Level:	8

Course Description:

Covers fundamental concepts of parallel computing not from the point of view of hardware, but from a more abstract view of algorithmic and implementation patterns. The aim is to facilitate the parallel programming by surveying some key algorithmic structures and programming models, together with an abstract representation of the underlying hardware and core concepts in parallel computing. SIMD, shared memory, and distributed memory machine models are covered, along with a brief discussion of what their execution starting with a naive example, and continuing with a discussion of some key algorithmic structures. Important programming models are presented in depth, as well as important concepts of performance analysis, including work-depth analysis of task graphs, communication analysis of distributed memory algorithms and key performance metrics. Fundamentals of distributed computing will be covered.

Course Aims:

1. Fundamentals of Parallel computing
2. Parallel Machines and parallel execution models such as SMID,
3. Algorithms used for parallel processing and their structures.
4. Parallel Program structures.
5. Performance analysis matrix and its optimizations
6. Fundamentals of distributed computing
7. Models of distributed computations

Student Outcomes (SOs):

☐(a) An ability to apply knowledge of computing and mathematics appropriate to the program's student outcomes and to the discipline

☒(b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution

☐(c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs

☐(d) An ability to function effectively on teams to accomplish a common goal

☐(e) An understanding of professional, ethical, legal, security and social issues and responsibilities

☐(f) An ability to communicate effectively with a range of audiences

☐(g) An ability to analyze the local and global impact of computing on individuals, organizations, and society

☐(h) Recognition of the need for and an ability to engage in continuing professional development

☒(i) An ability to use current techniques, skills, and tools necessary for computing practice.

☒(j) An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices. [CS]

☒(k) An ability to apply design and development principles in the construction of software systems of varying complexity. [CS]

☐(j) An ability to use and apply current technical concepts and practices in the core information technologies of human computer interaction, information management, programming, networking, and web systems and technologies. [IT]

☐(k) An ability to identify and analyze user needs and take them into account in the selection, creation, evaluation, and administration of computer-based systems. [IT]

☐(l) An ability to effectively integrate IT-based solutions into the user environment. [IT]

☐(m) An understanding of best practices and standards and their application. [IT]

☐(n) An ability to assist in the creation of an effective project plan. [IT]

Course Learning Outcomes (CLOs):

1. Appreciate and understand the importance of parallel computing in solving practical problems in science and engineering (parallel thinking).
2. Be able to select the proper parallel processing strategy that is expected to work best for solving a particular problem in given parallel processing platform.
3. Be able to develop a parallelization strategy for numerical and other algorithms for science and engineering problems, such as sorting the records of a large database and any other relevant problems (parallel algorithms)
4. Be able to develop parallel program structures and also understand existing parallel program structures available in parallel programming.
5. Appreciate and understand the fundamentals of distributed computing.

SOs and CLOs Mapping:

CLO/SO	a	b	c	d	e	f	g	h	i	j	k	l	m	n
CLO1		√												

CLO2		√											
CLO3										√			
CLO4									√				
CLO5								√					

No.	Topics	Weeks	Teaching hours
1	Overview of Parallel Computing INTRODUCTION TERMINOLOGY EVOLUTION OF PARALLEL COMPUTERS EXAMPLE: WORD COUNT PARALLEL PROGRAMMING MODELS Implicit Models Semi-Implicit Models Explicit Models Thinking in Parallel	01	03
2	PARALLEL DESIGN PATTERNS Structural Patterns Computational Patterns Patterns in the Lower Layers WORD COUNT IN PARALLEL Parallel Machine and Execution Models PARALLEL MACHINE MODELS SIMD Shared Memory and Distributed Memory Computers	02	6
3	Distributed Memory Execution Shared Memory Execution Summary PARALLEL EXECUTION MODEL Task Graph Model Parallel Algorithmic Structures HISTOGRAM EXAMPLE	02	6

	Guidelines for Parallel Algorithm Design EMBARRASSINGLY PARALLEL		
4	REDUCTION SCAN DIVIDE AND CONQUER PIPELINE DATA DECOMPOSITION Parallel Program Structures LOAD BALANCE SIMD: STRICTLY DATA PARALLEL FORKJOIN PARALLEL LOOPS AND SYNCHRONIZATION Shared and Private Variables Synchronization Thread Safety	02	6
5	TASKS WITH DEPENDENCIES SINGLE PROGRAM MULTIPLE DATA MASTERWORKER DISTRIBUTED MEMORY PROGRAMMING Distributed Arrays Message Passing Map-Reduce	02	6
6	Performance Analysis and Optimization WORKDEPTH ANALYSIS PERFORMANCE ANALYSIS	01	3
7	Performance Metrics Communication Analysis BARRIERS TO PERFORMANCE MEASURING AND REPORTING PERFORMANCE	01	3
8	Fundamentals of distributed computing Definition Relation to computer system components Motivation	01	3

	Relation to parallel multiprocessor/multicomputer systems Message-passing systems versus shared memory systems Primitives for distributed communication Synchronous versus asynchronous executions Design issues and challenges		
9	Model of distributed computations A distributed program A model of distributed executions Models of communication networks	02	4
10	Global state of a distributed system Cuts of a distributed computation Past and future cones of an event Models of process communications		2
	Total	14	42

Textbook:

- Elements of Parallel Computing by Eric Aubanel, Chapman and Hall/CRC ,December 6, 2016 ISBN 9781498727891. (Required)

Essential references:

- Distributed Computing Principles, Algorithms, and Systems by Ajay D. Kshemkalyani and Mukesh Singhal, Cambridge university press, 2008, ISBN : 9780521876346
- David Kirk and Wen-Wei Hwu Programming Massively Parallel Processors Morgan Kaufmann (2nd Edition), 2012. (recommended)