Department of Computer Science & Engineering

Academic Year	:	2026-27	Course Name	:	Parallel computing
Semester	:	7B	Course Code	:	BCS702
Scheme	:	22	L: T: P: S	:	3:0:2:1 Credit : 4
Total Contact hours	:	40 hours Theory + 10 Slots (20 Hours)	CIE Marks	:	50
Course Plan Author	:	Latha A	SEE Marks	:	50
Date	:	25/6/26	Total Marks	:	100

Course Prerequisites:

- Familiarity with programming languages like C, n is typically required, along with an understanding of how computers work at a hardware level.

Learning Objectives:

This course will enable to,

- Explore the need for parallel programming
- Explain how to parallelize on MIMD systems
- To demonstrate how to apply MPI library and parallelize the suitable programs
- To demonstrate how to apply OpenMP pragma and directives to parallelize the suitable programs
- To demonstrate how to design CUDA programs

Course Outcomes:

CO		*At the end of the course, student should be able to . . .*	**Blooms' Level**
CO1	:	Understand the need for parallel programming	L2
CO2	:	Demonstrate parallelism in MIMD system.	L2
CO3	:	Apply MPI library to parallelize the code to solve the given problem.	L3
CO4	:	Apply OpenMP pragma and directives to parallelize the code to solve the given problem.	L3
CO5	:	Design a CUDA program for the given problem.	L3

Blooms' Taxonomy:

L1	L2	L3	L4	L5	L6
Remembering	**Understanding**	**Applying**	**Analyzing**	**Evaluating**	**Creating**

Program Outcomes:

PO1	:	Engineering knowledge	**PO7**	:	Environment and sustainability
PO2	:	Problem analysis	**PO8**	:	Ethics
PO3	:	Design/development of solutions	**PO9**	:	Individual and team work
PO4	:	Conduct investigations of complex problems	**PO10**	:	Communication
PO5	:	Modern tool usage	**PO11**	:	Project management and finance
PO6	:	The engineer and society	**PO12**	:	Life-long learning

Program Specific Outcomes:

PSO1:	Model computational problems by applying mathematical concepts and design solutions using suitable data structures & algorithmic techniques.
PSO2:	Demonstrate basic knowledge of computer science in efficient design of problem solutions of varying complexity.
PSO3:	Create career path to become a successful computer science professional, entrepreneur and relish for higher studies

CO-PO-PSO Mapping:

CO	Program Outcomes														
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 1 2	PSO 1	PSO 2	PSO 3
CO1	3	2	1		2							2	2	2	1
CO2	3	3	2	2	2				1			2	3	3	1
CO3	3	3	3	2	3				2	1		2	3	3	2
CO4	3	3	3	2	3				2	1		2	3	2	2
CO5	3	3	3	2	3				2	1	1	2	3	3	3

CO-PO Mapping justification

CO	PO	Correlation Weightage	Justification
CO1	PO1	3	Applies engineering fundamentals to understand parallel architectures and computing models.
	PO2	2	Analyzes computational limitations of sequential programs and identifies opportunities for parallelism.

	PO3	1	Demonstrates introductory understanding of designing basic parallel solutions.
	PO5	1	Uses modern simulation and programming tools to study parallel computing concepts.
	PO12	2	Understanding emerging parallel computing technologies encourages continuous professional development and self-learning.
	PSO1	2	Understands computational models for solving engineering problems.
	PSO2	2	Builds foundation in parallel computing concepts.
	PSO3	1	Introduces students to advanced computing domains useful for future careers.
CO2	PO1	3	Applies computer engineering principles to MIMD architectures and memory models
	PO2	3	Analyzes task decomposition, synchronization, scalability, and performance.
	PO3	2	Designs basic parallel algorithms using suitable decomposition techniques.
	PO4	2	Students evaluate system performance using speedup, efficiency, and scalability measures.
	PO5	2	Students use simulation tools and parallel programming environments to demonstrate MIMD execution.
	PO9	1	Understands collaborative execution models involving multiple processes and threads
	PO12	1	Students keeps pace with evolving multicore and many-core architectures.memory models, and interconnection networks.
	PSO1	3	Models computational problems suitable for parallel execution.
	PSO2	3	Applies computer science principles to efficient parallel system design.
	PSO3	1	Builds awareness of current high-performance computing technologies.
CO3	PO1	3	Students apply concepts of distributed-memory systems and message-passing paradigms.
	PO2	3	Students analyze computational problems and identify suitable data partitioning and communication mechanisms.
	PO3	3	Students develop efficient distributed-memory solutions using MPI functions and collective communication operations.
	PO4	2	Students investigate program performance through execution time, communication overhead, and scalability analysis.
	PO5	3	Students extensively use MPI libraries, compilers, and performance evaluation tools.
	PO9	2	MPI-based projects often require collaboration in designing and testing parallel applications.
	PO10	1	Documents and explains MPI program implementation and results.

	PO12	1	Students acquire skills in a widely used parallel programming framework, supporting lifelong learning.
	PSO1	3	Designs efficient distributed algorithms using appropriate computational models.
	PSO2	3	Implements scalable solutions using modern programming techniques.
	PSO3	2	Develops practical HPC programming skills valuable for professional careers.
CO4	PO1	3	Students apply knowledge of shared-memory systems, cache coherence, and thread management.
	PO2	3	Students analyze dependencies, synchronization issues, and workload distribution in parallel programs.
	PO3	3	Students design shared-memory parallel solutions using OpenMP directives and pragmas.
	PO4	2	Students investigate performance issues such as scheduling, false sharing, and thread safety.
	PO5	3	Students effectively use OpenMP programming tools and compiler directives.
	PO9	1	Demonstrates thread coordination and synchronization concepts.
	PO10	1	Communicates program design and performance evaluation effectively.
	PO12	2	Updates knowledge of multicore programming technologies.
	PSO1	3	Applies algorithmic techniques for shared-memory systems.
	PSO2	3	Develops efficient software using OpenMP programming.
	PSO3	2	Enhances employability in multicore software development.
CO5	PO1	2	Students apply knowledge of GPU architecture, heterogeneous computing, and CUDA programming models
	PO2	3	Students analyze computational problems to identify portions suitable for GPU acceleration.
	PO3	3	Students design and implement optimized CUDA solutions using threads, blocks, and grids.
	PO4	2	Students evaluate GPU performance through speedup analysis, memory optimization, and execution profiling.
	PO5	3	Students utilize CUDA toolkits, GPU programming environments, and performance analysis tools.
	PO9	1	CUDA application development may involve collaborative coding, testing, and optimization activities.
	PO10	1	Students understand CUDA implementation and performance analysis clearly.
	PO11	2	Appreciates resource optimization and efficient utilization of computing infrastructure.

	PO12	1	Acquires advanced GPU computing skills supporting lifelong learning and emerging technologies
	PSO1	3	Designs optimized GPU-based algorithms for computational problems.
	PSO2	3	Applies advanced programming techniques to solve complex computing problems efficiently
	PSO3	3	Builds expertise in GPU computing, supporting careers, entrepreneurship, and higher education in High Performance Computing, Artificial Intelligence, and Data Science.

Course Content (Syllabus)

Module 1	**CH**
Introduction to parallel programming, Parallel hardware and parallel software – Classifications of parallel computers, SIMD systems, MIMD systems, Interconnection networks, Cache coherence, Shared-memory vs. distributed-memory, Coordinating the processes/threads, Shared-memory, Distributed-memory.	
Module 2	
GPU programming, Programming hybrid systems, MIMD systems, GPUs, Performance – Speedup and efficiency in MIMD systems, Amdahl's law, Scalability in MIMD systems, Taking timings of MIMD programs, GPU performance.	
Module 3	
Distributed memory programming with MPI – MPI functions, The trapezoidal rule in MPI, Dealing with I/O, Collective communication, MPI-derived datatypes, Performance evaluation of MPI programs, A parallel sorting algorithm.	
Module 4	
Shared-memory programming with OpenMP – openmp pragmas and directives, The trapezoidal rule, Scope of variables, The reduction clause, loop carried dependency, scheduling, producers and consumers, Caches, cache coherence and false sharing in openmp, tasking, tasking, thread safety.	
Module 5	
GPU programming with CUDA - GPUs and GPGPU, GPU architectures, Heterogeneous computing, Threads, blocks, and grids Nvidia compute capabilities and device architectures, Vector addition, Returning results from CUDA kernels, CUDA trapezoidal rule I, CUDA trapezoidal rule II: improving performance, CUDA trapezoidal rule III: blocks with more than one warp.	**08**

Schedule of Instruction:

Class No	Topic	Date	RBT	CO	Mode
Module-1	**Chapter Name Introduction to parallel programming**				
1.	Introduction to CO, Subject,Subject Prerquisites	11-08-25	L2	CO1	PPT
2.	Introduction and classifications of Parallel computers	12-08-25	L2	CO1	C&B
3.	SIMD systems	14-08-25	L2	CO2	C&B
4.	MIMD systems	18-08-25	L2	CO2	C&B
5.	Cache coherence	19-08-25	L2	CO1	C&B
6.	Cache coherence	19-08-25	L2	CO1	PPT
7.	Shared-memory	21-08-25	L2	CO1	C&B
8.	Distributed-memory	25-08-25	L2	CO1	C&B
9.	Coordinating the processes/threads	28-08-25	L2	CO1	C&B
10.	Shared-memory	1-09-25	L2	CO2	Tutorial
11.	Distributed-memory	2-09-25	L2	CO2	Self Study
Module-2	**Chapter Name**				
12.	Speedup	4-09-25	L2	CO2	Demonstration
13.	Efficiency in MIMD systems	8-09-25	L2	CO2	C&B
14.	Amdahl's law	9-09-25	L3	CO2	Brainstorming
15.	Scalability in MIMD systems	11-09-25	L2	CO2	C&B
16.	Scalability in MIMD systems	15-09-25	L2	CO2	Self study
17.	Taking timings of MIMD programs	16-09-25	L2	CO1	C&B
18.	GPU performance	18-09-25	L2	CO1	Tutorial
19.	GPU performance	22-09-25	L3	CO1	Demonstration
Module-3	**Chapter Name**				
20.	MPI functions	23-09-25	L2	CO3	C&B
21.	The trapezoidal rule in MPI	29-09-25	L2	CO3	C&B
	Open Book Assessment				
22.	Dealing with I/O	30-09-25	L2	CO3	C&B
23.	Collective communication	06-10-25	L2	CO3	Demonstration
24.	Revision				Group

Class No	Topic	Date	RBT	CO	Mode
					Discussion
25.	Collective communication	07-10-25	L2	CO3	C&B
26.	MPI-derived datatypes	09-10-25	L3	CO3	C&B
27.	Performance evaluation of MPI programs	13-10-25	L3	CO3	Demonstration
28.	Performance evaluation of MPI programs	13-10-25	L3	CO3	Demonstration
29.	A parallel sorting algorithm	14-10-25	L2	CO3	Self study
30.	CIE Scheme Discussion				
Module-4	**Chapter Name**				
31.	openmp pragmas and directives	16-10-25	L2	CO4	C&B
32.	The trapezoidal rule	17-10-25	L2	CO4	Tutorial
33.	Scope of variables	23-10-25	L3	CO4	Demonstration
34.	The reduction clause	24-10-25	L3	CO4	C&B
35.	loop carried dependency	28-10-25	L3	CO4	PPT
36.	scheduling, producers and consumers	30-10-25	L2	CO4	PPT and Mindmaps
37.	Caches, cache coherence and false sharing in openmp	3-11-25	L2	CO4	Tutorial
38.	Poster presentation				Presentation
Module-5	**Chapter Name**				
39.	GPUs and GPGPU, GPU architectures	4-11-25	L2	CO5	C&B
40.	Heterogeneous computing	6-11-25	L2	CO5	PPT
41.	Threads, blocks, and grids	7-11-25	L2	CO5	C&B
42.	Nvidia compute capabilities and device architectures	10-11-25	L3	CO5	PPT
43.	Vector addition	11-11-25	L2	CO5	PPT
44.	Returning results from CUDA kernels,	13-11-25	L2	CO5	C&B
45.	CUDA trapezoidal rule-I	14-11-25	L2	CO5	C&B
46.	CUDA trapezoidal rule - II	17-11-25	L2	CO5	Tutorial
47.	CUDA trapezoidal rule-III	17-11-25	L3	CO5	Brainstrom
48.	Poster presentation				Presentation

Class No	Topic	Date	RBT	CO	Mode
49.	Revision				Discussion

Practical Component of IPCC

Lab	Topic	Date	RBT	CO	Mode
	Introduction to Linux Based Terminals, Basic Commands, Editor Commands		**L2**		
	Introduce gcc compiler, Hello Program in C, Demonstration of looping statements and conditional statements,Omp directive		**L2**		
	Write a OpenMP program to sort an array on n elements using both sequential and parallel mergesort(using Section). Record the difference in execution time.	18-08-25	L3	CO4	EBL
	Write an OpenMP program that divides the Iterations into chunks containing 2 iterations,respectively(OMP_SCHEDULE=static, 2). Its input should be the number of iterations, and its output should be which iterations of a parallelized for loop are executed by which thread. For example, if there are two threads and four iterations, the output might be the following: 1. `Thread 0 : Iterations 0 -- 1` 2. Thread 1 : Iterations 2 – 3	25-08-25	L3	CO4	EBL
	Write an OpenMP program to calculate n Fibonacci numbers using tasks.	01-08-25	L3	CO4	EBL
	Write an OpenMP program to find the prime numbers from 1 to n employing parallel for directive. Record both serial and parallel execution times.	08-09-25	L3	CO4	EBL
	Write an MPI Program to demonstrate MPI_Send and MPI_Recv.	15-09-25	L3	CO3	EBL
	Write a MPI program to demonstration of deadlock using point to point communication and avoidance of deadlock by altering the call sequence	22-09-25	L3	CO3	EBL
	Write a MPI Program to demonstrate Broadcast operation.	29-09-25	L3	CO3	EBL
	Write a MPI Program demonstration of MPI_Scatter and MPI_Gather	06-09-25	L3	CO3	EBL
	Write a MPI Program to demonstration of MPI_Reduce and MPI_Allreduce (MPI_MAX, MPI_MIN, MPI_SUM, MPI_PROD)	13-10-25	L3	CO3	EBL

Textbooks:

T2	Michael J Quinn – Parallel Programming in C with MPI and OpenMp, McGrawHill.

Reference books:

R1	Calvin Lin, Lawrence Snyder – Principles of Parallel Programming, Pearson
R2	Barbara Chapman – Using OpenMP: Portable Shared Memory Parallel Programming Scientific and Engineering Computation

Web links and Video Lectures (e-Resources):

1.	https://sites.google.com/skit.org.in/latha
2.	Introduction to parallel programming: https://nptel.ac.in/courses/106102163

Assessment Schedule:

S.N.	Assessment Type	Content	CO	Duration	Marks	Date
1.	CIE 1 (Theory)	M1,M2,M3	CO1,CO2,CO3	1:30 hrs	50	
2.	CIE 2 (Theory)	M3,M4,M5	CO3,CO4,CO5	1:30 hrs	50	
CIE Scaled down to					15	
3.	CIE (Lab) - Record	M1,M2,M3 M4,M5	CO1,CO2,CO3 CO4,CO5	8-9 labs of each 2 hrs	15	
4.	CIE (Lab) - Test	M1,M2,M3 M4,M5	CO1,CO2,CO3 CO4,CO5	2:00 hrs	10	
5.	Open Book Assessment	M1,M2	CO1,CO2	1.00 hrs	5	
6.	Poster presentation using questionaries	M3 M4,M5	CO3 CO4,CO5	3.00 hrs	5	
7.	Semester End Examination	M1,M2,M3 M4,M5	CO1,CO2,CO3 CO4,CO5	3:00	50	
Total					100	

***RB** – Text Book/**R**eference **B**ook, **L** – **L**ecture, **V**- Videos or any other mode, **RBT** – **R**evised **B**looms' **T**axonomy, **L: T: P: C** – Theory/**L**ecture: **T**utorial: **P**ractical/Drawing: **C**redits, **SEE**: **S**emester **E**nd **E**xamination, **CIE**: **C**ontinuous **I**nternal **E**valuation, **Seminar**: Group of 6-8 students, **Module** 1,2,3,4 & 5,*

Assessment Details (both CIE and SEE)

The weightage of Continuous Internal Evaluation (CIE) is 50% and for Semester End Exam (SEE) is 50%.The minimum passing mark for the CIE is 40% of the maximum marks (20 marks out of 50) and for the SEE minimum passing mark is 35% of the maximum marks (18 out of 50 marks). A student shall be deemed to have satisfied the academic requirements and earned the credits allotted to each subject/course if the student secures a minimum of 40% (40 marks out of 100) in the sum total of the CIE (Continuous Internal Evaluation) and SEE (Semester End Examination) taken together.

CIE for the theory component of the IPCC (maximum marks 50)

- *IPCC means practical portion integrated with the theory of the course.*
- *CIE marks for the theory component are 25 marks and that for the practical component is 25 marks.*
- *25 marks for the theory component are split into 15 marks for two Internal Assessment Tests (Two Tests, each of 15 Marks with 01-hour duration, are to be conducted) and 10 marks for other assessment methods mentioned in 22OB4.2. The first test at the end of 40-50% coverage of the syllabus and the second test after covering 85-90% of the syllabus.*
- *Scaled-down marks of the sum of two tests and other assessment methods will be CIE marks for the theory component of IPCC (that is for 25 marks).*
- *The student has to secure 40% of 25 marks to qualify in the CIE of the theory component of IPCC.*

CIE for the practical component of the IPCC

- *15 marks for the conduction of the experiment and preparation of laboratory record, and 10 marks for the test to be conducted after the completion of all the laboratory sessions.*
- *On completion of every experiment/program in the laboratory, the students shall be evaluated including viva-voce and marks shall be awarded on the same day.*
- *The CIE marks awarded in the case of the Practical component shall be based on the continuous evaluation of the laboratory report. Each experiment report can be evaluated for 10 marks. Marks of all experiments' write-ups are added and scaled down to 15 marks.*
- *The laboratory test (duration 02/03 hours) after completion of all the experiments shall be conducted for 50 marks and scaled down to 10 marks.*
- *Scaled-down marks of write-up evaluations and tests added will be CIE marks for the laboratory component of IPCC for 25 marks.*
- *The student has to secure 40% of 25 marks to qualify in the CIE of the practical component of the IPCC.*

SEE for IPCC

Theory SEE will be conducted by University as per the scheduled timetable, with common question papers for the course (duration 03 hours)

1. The question paper will have ten questions. Each question is set for 20 marks.

2. There will be 2 questions from each module. Each of the two questions under a module (with a maximum of 3 sub-questions), should have a mix of topics under that module.

3. The students have to answer 5 full questions, selecting one full question from each module.

4. Marks scored by the student shall be proportionally scaled down to 50 Marks

The theory portion of the IPCC shall be for both CIE and SEE, whereas the practical portion will have a CIE component only. Questions mentioned in the SEE paper may include questions from the practical component.

Faculty In charge **Course Coordinator** **HoD**