Program Elective-IV

COURSE TITLE: Parallel Algorithms

Course Code:				**CSE-242051EAC**		**Examination Scheme**	
Total number of Lecture Hours: 56						**External**	**80**
						Internal	**20**
Lecture (L):	**4**	**Practicals(P):**	4	**Tutorial (T):**	**0**	**Total Credits**	6

Course Objectives

1. **Knowledge**: Apply computing principles.
2. **Analysis**: Identify and validate parallel algorithms.
3. **Design:** Develop cost-effective parallel solutions.
4. **Research**: Conduct algorithm simulations for performance.
5. **The** course focuses on an alternative to sequential model, parallel algorithm performance measure, and application of parallel algorithm in different domain.

Course Content	TEACHING HOURS
UNIT 1	**14 Hrs**
Sequential model, need of alternative model, parallel computational models such as PRAM, LMCC, Hypercube, Cube Connected Cycle, Butterfly, Perfect Shuffle Computers, Tree model, Pyramid model, Fully Connected model, PRAMCREW, EREW models, simulation of one model from another one.	
UNIT 2	**14 Hrs**
Performance Measures of Parallel Algorithms, speed-up and efficiency of PA, Cost optimality, an example of illustrate Cost-optimal algorithms- such as summation, Min/Max on various models. Parallel Sorting Networks, Parallel Merging Algorithms on CREW/EREW/MCC/, Parallel Sorting Networks on CREW/EREW/MCC/, linear array	
UNIT 3	**14 Hrs**
Parallel Searching Algorithm, Kth element, Kth element in X+Y on PRAM, Parallel Matrix Transportation and Multiplication Algorithm on PRAM, MCC, VectorMatrix Multiplication, Solution of Linear Equation, Root finding.	
UNIT 4	**14 Hrs**
Graph Algorithms - Connected Graphs, search and traversal, Combinatorial Algorithms- Permutation, Combinations, Derangements.Case study of parallel sorting networks, which include CREW, EREW and their performance analysis for various problems.	

Textbooks

1. **Parallel Computing: Theory and Practice" by Michael J. Quinn** - McGraw-Hill, 2nd Edition, 2018.
2. **"Introduction to Parallel Computing" by Ananth Grama et al.** - Addison-Wesley, 2nd Edition, 2020.
3. **"Elements of Parallel Computing" by V. Rajaraman and C. Siva Ram Murthy** - PHI Learning, 2021 (Indian Author).

Reference Books

1. **"Fundamentals of Parallel Processing" by Jordan H. Kerkhoff** - Prentice Hall, 2008.
2. **"Parallel and Distributed Computing" by M.L. Liu** - Wiley, 2004.

COURSE OUTCOMES (CO):

- CO1: Explain various parallel models and alternatives to sequential models.
- CO2: Analyze performance measures and cost-optimality in parallel algorithms.
- CO3: Implement parallel algorithms for sorting, searching, and matrix operations.
- CO4: Apply parallel algorithms to graph and combinatorial problems, comparing model performance.
- CO5 at the end of this course the student will know about parallel computing model PRAM, LMCC etc., efficiency of parallel algorithms, parallel sorting network, parallel search algorithm, Permutation, graph algorithm, combinations.

LEVEL OF CO-PO MAPPING TABLE

COs	POs											
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
1	3	2	–	1	–	–	–	–	–	2	–	–
2	3	3	2	–	–	3	2	3	–	2	–	2
3	2	3	2	–	–	–	–	2	1	3	–	3
4	2	2	3	2	1	–	–	–	–	3	2	–

Lab Manual-Parallel Algorithms

Students are encouraged to perform hands-on lab exercises on the following topics:

- Hands-on with distributed operating systems, network operating systems, and distributed file systems.
- Working with Middleware and client/server model for computing.
- Implementation of Layered Protocols, Remote Procedure Calls (RPC), and Remote Method Invocation (RMI).
- Developing and interacting with remote objects.

- Implementation of basic algorithms in message-passing systems.
- Implementing multithreading in distributed systems.
- Development of client-server models and code migration techniques.
- Working with distributed objects using CORBA (Common Object Request Broker Architecture) and Distributed COM.
- Simulating transparency in distributed databases.
- Designing distributed query processing and optimization.
- Implementing commit protocols for distributed transactions.
- Hands-on with parallel processing concepts and terminology.
- Implementing parallel algorithms for distributed computing applications.
- Simulating parallel query evaluation techniques.

COURSE TITLE: Threaded and Message-Passing Programming

Course Code:				**CSE-242052EAC**		**Examination Scheme**		
Total number of Lecture Hours:						**External**	**120**	
						Internal	**30**	
Lecture (L):	**4**	**Practicals(P):**	**4**	**Tutorial (T):**	**-**	**Total Credits**	**6**	

Course Objectives

- Be familiar with the common attributes and design concerns of message-passing systems
- Be able to evaluate the suitability of different message-passing approaches for a particular application
- Understand the benefits and costs of formal verification of a concurrent system, and the situations in which it is appropriate
- Be able to design multi-core and distributed applications using several modern message-passing programming paradigms
- Have experience of implementing multi-core and distributed applications using a variety of message-passing systems.
- Be able to familiarise themselves rapidly with new programming languages.

Course Content	**TEACHING HOURS**
UNIT 1: Introduction to MPI and Advanced I/O Techniques	**14 Hrs**
Introduction to MPI, Topologies, Remote Memory Access, Dynamic Process Management, Parallel I/O, Non-contiguous Accesses, Collective I/O, Arrays, Distributed Arrays, Non-blocking I/O, Split Collective I/O, Shared File Pointers, Consistency Semantics, File Interoperability.	
UNIT 2: Synchronization and Dynamic Process Management	**14 Hrs**
Synchronization, Remote Memory Operations. Dynamic Process Management: Creating and Connecting MPI Processes, Design of the MPI Dynamic Process Routines.	
UNIT 3: Thread Safety and Mixed-Model Programming in MPI	**14 Hrs**
Thread, Thread Safety, Mixed-Model Programming: MPI for SMP Clusters, Decoding Data types, Generalized Requests, Adding New Error Codes and Classes, Attribute Caching, Error Handling.	
UNIT 4: Case Studies: Remote Memory Access and Mixed-Model Programming	**14 Hrs**
Case study of Remote Memory Access, Dynamic Process Management and Mixed-Model Programming.	

Textbooks

Pacheco, P. (2011). *An introduction to parallel programming* (1st ed.). Morgan Kaufmann.

Reference Books

1. William Gropp Ewing Lusk Rajeev Thakur, Using MPI-2: Advanced Features of the Message-Passing Interface, MIT Press.

COURSE OUTCOMES (CO):

CO1: Design MPI for distributed applications.

CO2: Familiarize themselves rapidly with new programming languages.

CO3:Demonstrate an understanding of MPI topologies, dynamic process management, and advanced parallel I/O techniques, including shared file pointers and consistency semantics.

CO4: Apply synchronization techniques and manage remote memory operations through dynamic process management in MPI environments.

CO5: Implement thread-safe MPI programs using mixed-model programming for SMP clusters, handling custom data types, error codes, and attribute caching.

CO6: Analyze real-world case studies of remote memory access and mixed-model programming to enhance parallel computing solutions.

LEVEL OF CO-PO MAPPING TABLE

	POs											
COs	1	2	3	4	5	6	7	8	9	10	11	12
1	3	2	3	3	3	2	1	-	2	2	3	2
2	3	2	2	2	3	-	-	-	2	2	2	3
3	3	3	3	3	3	-	1	-	2	3	2	2
4	3	3	3	3	3	1	2	-	2	2	2	2
5	3	2	3	2	3	2	-	-	2	3	2	3
6	3	3	3	3	3	3	2	1	3	3	3	3

LAB MANUAL

- Setting up an MPI environment.
- Writing and running a basic "Hello World" MPI program.
- Visualization and analysis of topology-based communication patterns.
- Introduction to `MPI_Put`, `MPI_Get`, and `MPI_Accumulate`.
- Performing file I/O operations using MPI I/O functions (`MPI_File_open`, `MPI_File_write`).
- Implementation of collective I/O operations using `MPI_File_read_all` and `MPI_File_write_all`.
- Implementing synchronization using barriers (`MPI_Barrier`) and locks.
- Practical exercises on managing deadlocks and avoiding race conditions.
- Creating and connecting MPI processes dynamically using `MPI_Comm_spawn`.
- Configuring and running MPI in multi-threaded environments using `MPI_Init_thread`.
- Developing a distributed matrix multiplication program using MPI RMA.
- Performance comparison between RMA and traditional send/receive methods.
- Analyzing the impact of hybrid programming on performance and resource utilization.

COURSE TITLE: Human Centered Computing

Course Code:				**CSE-242053EAC**		**Examination Scheme**	
Total number of Lecture Hours: 56						**External**	**80**
						Internal	**20**
Lecture (L):	**4**	**Practicals(P):**	**4**	**Tutorial (T):**	**-**	**Total Credits**	**6**

Course Objectives

- **Understanding Fundamental Concepts**: Explore the foundational principles of mathematical and computational modeling, including parameter estimation, model selection, and non-parametric methods.
- **Analyzing Search Mechanisms**: Investigate classical and advanced information retrieval techniques, contextual search paradigms, and temporal information retrieval to enhance information discovery and serendipity.
- **Exploring Human-Centered Systems**: Examine recommender systems, sentiment analysis, and affect measurement techniques, integrating human preferences and emotions into computational frameworks.
- **Developing Goal-Oriented Agents**: Understand the principles of goal-directed agents, reinforcement learning, and intrinsic motivation while exploring applications like gamification and emerging trends in human-centered computing.

Course Content	**TEACHING HOURS**
UNIT 1: Prelims	**-Hrs**
Intro, logistics, overview, Introduction to small data, Different flavors of mathematical/computational models, Model fitting, evaluation metrics, Parameter estimation, model selection and non-parametric methods.	14
UNIT 2: Search	**- Hrs**
Classical search/information retrieval, Query completion, Contextual/topical search foci, Information scent and other foraging models, Temporal information retrieval, Serendipity, discovery.	14
UNIT 3: Preferences & Emotions	**-Hrs**
Recommender systems, Collaborative filtering, Feature selection, Theories and schema, Sentiment analysis, Affect measurement (computer vision, survey instruments, activity monitoring), Chatbots to emotebots, Brain-computer interface, Boredom/ennui.	14
UNIT 4: Goals	**-Hrs**
Basic goal-directed agents, Hebbian/reinforcement learning, Explore-exploit dilemma, Curiosity, perseverance, intrinsic motivation as goals, Gamification, Deep principles – flow, connectedness, homeostasis, etc. Recent trends and applications of Human Centered Computing.	14

Textbooks

2. Croft, Metzler, Strohman. Search engines: Information Retrieval in practice. Pearson Education
3. Pang, B., & Lee, L. (2008).Opinion mining and sentiment analysis. Foundations and trends in information retrieval, 2(1-2), 1-135.
4. Shapira, B., Ricci, F., Kantor, P. B., & Rokach, L. (2011). Recommender Systems Handbook. Springer Press.

To be effective from year-2024

5. Sutton, R. S., & Barto, A. G. (1998).Reinforcement learning: An introduction (Vol. 1, No. 1). Cambridge: MIT press.

Reference Books

2. Picard, R. W., & Picard, R. (1997).Affective computing (Vol. 252). Cambridge: MIT press.

COURSE OUTCOMES (CO):

CO1: Demonstrate proficiency in applying mathematical and computational models for data fitting, evaluation, and parameter estimation in small data contexts.
CO2: Develop practical skills in implementing and evaluating information retrieval systems, including query completion, contextual search, and temporal retrieval.
CO3: Design and analyze human-centered systems incorporating sentiment analysis, affect measurement, and recommender systems to address real-world problems.
CO4: Apply concepts of goal-directed behavior, reinforcement learning, and intrinsic motivation to create innovative computational solutions with gamified elements and human-centric design principles.

LEVEL OF CO-PO MAPPING TABLE

	POs											
COs	1	2	3	4	5	6	7	8	9	10	11	12
1	3	3	2	3	-	2	2	-	-	2	2	3
2	3	3	3	3	2	2	2	-	-	3	3	3
3	3	3	3	2	2	3	3	3	2	3	3	3
4	3	3	3	2	3	2	3	3	3	3	2	3

Lab Manual

The lab should cover the following topics:

1. **Basic Curve Fitting**
2. **Evaluation Metrics**
3. **Parameter Estimation**
4. **Model Selection**
5. **Keyword Matching**
6. **Simple Query Completion**
7. **TF-IDF Computation**
8. **Basic Recommender System**
9. **Feature Selection**
10. **Sentiment Analysis**
11. **Basic Chatbot**
12. **User Engagement Analysis**
13. **Emotion Detection (Basic)**
14. **Interactive Content Suggestion**

Program Elective-V

COURSE TITLE: HIGH PERFORMANCE SCIENTIFIC COMPUTING

Course Code:				**CSE-243011EAC**		**Examination Scheme**	
Total number of Lecture Hours: 56						**External**	**80**
						Internal	**20**
Lecture (L):	**4**	**Practicals(P):**	**-**	**Tutorial (T):**	**-**	**Total Credits**	**4**

Course Objectives:

- Understand the foundational concepts of parallel system organization, message-passing paradigms, and MPI programming.
- Analyze and decompose embarrassingly parallel problems with a focus on efficient graph partitioning and load balancing techniques.
- Develop proficiency in shared memory programming using OpenMP and apply it to scientific computing scenarios.
- Apply knowledge of problem decomposition, graph partitioning, and load balancing through a practical case study using OpenMP.

Course Content	**TEACHING HOURS**
UNIT 1:	**-Hrs**
Overview of parallel system organization, Introduction to message passing and MPI programming	14
UNIT 2:	**- Hrs**
Embarrassingly parallel problems; Problem decomposition, graph partitioning, and load balancing	14
UNIT 3:	**-Hrs**
Introduction to shared memory and OpenMP programming; Examples of scientific computing	14
UNIT 4:	**-Hrs**
Case study of Problem decomposition, graph partitioning, and load balancing using OpenMP	14

Textbooks

- Parallel Programming for Multicore and Cluster Systems by Thomas Rauber and Gudula Runger.
- Scientific Parallel Computing by Scott, Clark, and Bagheri.
- Using OpenMP: Portable Shared Memory Parallel Programming by Chapman, Jost, and van der Pas.

Reference Books

- Parallel Programming for Multicore and Cluster Systems by Thomas Rauber and Gudula Runger.

COURSE OUTCOMES (CO):

CO1: Demonstrate an understanding of message-passing techniques and MPI programming in parallel systems.

CO2: Identify and implement strategies for problem decomposition, load balancing, and graph partitioning to optimize parallel computations.

CO3: Develop and execute shared memory solutions in scientific computing applications using OpenMP.

CO4: Integrate and apply parallel computing techniques to a complex case study, demonstrating problem-solving skills in practical parallel programming scenarios.

To be effective from year-2024

COURSE TITLE: Quantum Computing

Course Code:				CSE-243012EAC		Examination Scheme	
Total number of Lecture Hours: 56						External	80
						Internal	20
Lecture(L):	4	Practicasls(P):	0	Tutorial(T):	0	Total Credits	4

Course Objectives

- The course will provide an insight of basic of quantum physics from a computer scientist's perspective.
- Understand the principles of quantum mechanics and their application to computing.
- The course will also describe reality and understand the philosophical implications of quantum computing.

Course Content	TEACHING HOURS
UNIT 1:	- **Hrs**
Quantum bits (Qubits) & Quantum States: The Qubit, Vector Spaces. Linear Combination Of Vectors, Uniqueness of a spanning set, basis & dimensions, inner Products, orthonormality, gram-schmidt orthogonalization, bra-ket formalism, the Cauchyschwarez and triangle Inequalities.	14
UNIT 2:	**- Hrs**
Matrices & Operators: Observables, The Pauli Operators, Outer Products, The Closure Relation, Representation of operators using matrices, outer products & matrix representation, matrix representation of operators in two dimensional spaces, Pauli Matrix, Hermitian unitary and normal operator, Eigen values & Eigen Vectors, Spectral Decomposition, Trace of an operator, important properties of Trace, Expectation Value of Operator, Projection Operator, Positive Operators	14
UNIT 3:	**- Hrs**
Commutator Algebra, Heisenberg uncertainty principle, polar decomposition &singular values, Postulates of Quantum Mechanics. Tensor Products: Representing Composite States in Quantum Mechanics, Computing inner products, Tensor products of column vectors, operators and tensor products of Matrices. Density Operator	14
UNIT 4:	**- Hrs**
Quantum Measurement Theory: Distinguishing Quantum states & Measures, Projective Measurements, Measurement on Composite systems, Generalized Measurements, Positive Operator- Valued Measures. Recent trends in Quantum Computing Research, Quantum Computing Applications of Genetic Programming.	14

(189)

LEVEL OF CO-PO MAPPING TABLE

COs	POs											
	1	2	3	4	5	6	7	8	9	10	11	12
1	3	2	1	1	3	1	-	1	2	2	1	2
2	3	3	2	2	3	-	1	-	2	-	2	2
3	3	2	3	2	3	-	1	-	2	1	1	2
4	3	3	3	3	3	2	2	2	3	2	3	2

Textbooks

14. Hidary, Jack D. Quantum Computing: An Applied Approach. Springer, 2021.
15. Nielsen, Michael A., and Isaac L. Chuang. Quantum Computation and Quantum Information. Cambridge UP, 2000.

Reference Books

3. Soni, S. K. Quantum Programming for Beginners. Wiley, 2020.
4. Williams, E. M. R. M. C. A. Introduction to Quantum Computing for Engineers. Wiley, 2020.
5. Susskind, Leonard, and Art Friedman. Quantum Mechanics: The Theoretical Minimum. Basic Books, 2014.

COURSE OUTCOMES(CO):

CO1: Knowledge of Vector spaces, Matrices, Quantum state, Density operator and Quantum Measurement theory.

CO2: Explain quantum mechanics principles and apply them in quantum computing contexts.

CO3: Identify and assess the potential applications of quantum computing in fields like cryptography, AI, machine learning, and optimization.

LEVEL OF CO-PO MAPPING TABLE

	POs											
COs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
1	3	3	2	2	2	2	1	1	2	2	1	2
2	3	3	2	3	3	2	1	1	2	2	2	3
3	3	3	3	3	3	2	1	1	2	3	2	3