Specialization: Advanced Computing (AC)

Program Elective-I

COURSE TITLE: Compiler for HPC

Course Code:				CSE-241041EAC		**Examination**	**T**	**P**
Total number of Lecture Hours:56 **Total number of Practical Hours:-**						**External**	**80**	
						Internal	**20**	
Lecture (L):	**4**	**Practicals(P):**	**-**	**Tutorial (T):**	**-**	**Total Credits**	**4**	

Course Objectives:

- Understand the key principles of high-performance computing, focusing on compiler design and optimization techniques for parallel systems.
- Learn to analyze data dependencies and apply loop restructuring techniques to optimize performance in scalar and array operations.
- Develop skills in optimizing data locality, concurrency, and vectorization for efficient parallel processing.
- Explore advanced parallel computing architectures (SIMD, MIMD) and optimize data layouts and memory access for high-performance systems.

Course Content	**TEACHING HOURS**
UNIT 1: High Performance Systems and Compiler Design for Parallel Computing	**-14 Hrs**
High Performance Systems, Structure of a Compiler, Programming Language Features, Languages for High Performance, **Data Dependence:** Data Dependence in Loops, Data Dependence in Conditionals, Data Dependence in Parallel Loops, Program Dependence Graph.	
UNIT 2: Advanced Compiler Optimization and Data Dependence Analysis for Parallel Systems	**- 14 Hrs**
Scalar Analysis with Factored Use-Def Chains: Constructing Factored Use- Def Chains, FUD Chains for Arrays, Induction Variables Using FUD Chains, Constant Propagation with FUD Chains, Data Dependence for Scalars. Data Dependence Analysis for Arrays. Array Region Analysis, Pointer Analysis, I/O Dependence, Procedure Calls, Inter-procedural Analysis. **Loop Restructuring:** Simple Transformations, Loop Fusion, Loop Fission, Loop Reversal, Loop Interchanging, Loop Skewing, Linear Loop Transformations, Strip-Mining, Loop Tiling, Other Loop Transformations, and Inter-procedural Transformations. .	
UNIT 3: Optimization Techniques for Locality, Concurrency, and Vectorization in High Performance Computing	**-14 Hrs**
Optimizing for Locality: Single Reference to Each Array, Multiple References, General Tiling, Fission and Fusion for Locality **Concurrency Analysis:** Concurrency from Sequential Loops, Concurrency from Parallel Loops, Nested Loops, Round off Error, Exceptions and Debuggers. **Vector Analysis:** Vector Code, Vector Code from Sequential Loops, Vector	

Code from For all Loops, Nested Loops, Round off Error, Exceptions, and Debuggers, Multi-vector Computers.	
UNIT 4: Parallel Architecture and Data Layout Optimization for High Performance Systems	**-14 Hrs**
SIMD Machines, MIMD Machines, Data Layout, Parallel Code for Array Assignment, Remote Data Access, Automatic Data Layout, Multiple Array Assignments.	

Textbooks

1. Michael Wolfe, High-Performance Compilers for Parallel Computing, Pearson

COURSE OUTCOMES (CO):

CO1: Analyze data dependencies and optimize scalar and array operations using techniques like factored use-def chains (FUD chains).

CO2: Apply loop optimizations, such as fusion, fission, and tiling, to improve parallelism and locality in code.

CO3: Implement concurrency analysis and vectorization techniques to optimize performance in parallel loops and reduce execution time.

CO4: Optimize data layouts and memory access in SIMD and MIMD architectures to enhance the performance of parallel computing systems.

LEVEL OF CO-PO MAPPING TABLE

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

To be effective from year-2024

COURSE TITLE: Operating System Design

Course Code:			**CSE-241042EAC**		**Examination Scheme**	
Total number of Lecture Hours: 56					**External**	**80**
					Internal	**20**
Lecture (L):	**4**	**Practicals(P):**	**Tutorial (T):**	**-**	**Total Credits**	**4**

Course Objectives

- To understand the fundamental principles, design approaches, and types of advanced operating systems.
- To study process abstraction, process management, and system architecture, including threads and micro-kernels.
- To analyze scheduling techniques, concurrency mechanisms, and deadlock handling strategies.
- To explore memory management, I/O systems, file management, and security aspects of operating systems.

Course Content	**TEACHING HOURS**
Unit 1: Fundamentals of Operating Systems	**21 Hrs**
Computer system and operating system overview, Operating system functions and design issues, Design approaches, Types of advanced operating systems.	
Unit 2: Process Management and System Architecture	**14 Hrs**
Process abstraction, Process management, system calls, Threads, Symmetric multiprocessing and micro-kernels.	
Unit 3: Scheduling, Concurrency, and Deadlock Handling	**14 Hrs**
Scheduling: Uniprocessor, Multiprocessor and Real time systems, concurrency, classical problems, mechanisms for synchronization: semaphores, monitors, Process deadlock and deadlock handling strategies.	
Unit 4: Memory, I/O, and Security Management	**14 Hrs**
Memory management, Virtual memory concept, Virtual machines, I/O management, File and disk management, Operating system security.	

Textbooks

- Advanced concept in operating system: M. Singhal, N.G. Shivratri

Reference Books

- Operating system internal and design principles: William Stallings Watermarking and Steganography", Margan Kaufmann Publishers, New York, 2008.
-

COURSE OUTCOMES (CO):

CO1: Demonstrate a comprehensive understanding of operating system principles and advanced system types.

CO2: Analyze and apply process management and system architecture concepts in modern computing systems.

CO3: Develop solutions for scheduling, synchronization, and deadlock issues in various computing environments.

CO4: Evaluate and implement strategies for memory, I/O, and security management in operating systems.

LEVEL OF CO-PO MAPPING TABLE

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

COURSE TITLE: Optimization Techniques

Course Code:				**CSE-241043EAC**		**Examination Scheme**	
Total number of Lecture Hours: 48						**External**	**80**
						Internal	**20**
Lecture(L):	**4**	**Practical (P):**	**0**	**Tutorial(T):**	**0**	**Total Credits**	**4**

Course Objectives

- To introduce linear programming and optimization techniques, covering both graphical and simplex methods for solving linear problems.
- To explore unconstrained one-dimensional optimization techniques, including search methods and interpolation techniques for finding optimal solutions.
- To study unconstrained multi-dimensional optimization methods, such as random search, pattern search, and descent algorithms like steepest descent and quasi-Newton methods.
- To understand constrained optimization techniques, including conditions for optimality, Kuhn-Tucker conditions, and methods like gradient projection, cutting plane, and penalty function.

Course Content	**TEACHING HOURS**
UNIT 1:	**-Hrs**
Linear programming –formulation-Graphical and simplex methods-Big-M method Two phase method-Dual simplex method-Primal Dual problems.	**12**
UNIT 2:	**- Hrs**
Unconstrained one dimensional optimization techniques -Necessary and sufficient conditions – Unrestricted search methods -Fibonacci and golden section method Quadratic Interpolation methods, cubic interpolation and direct root methods.	**12**
UNIT 3:	**- Hrs**
Unconstrained n dimensional optimization techniques – direct search methods – Random search – pattern search and Rosen brooch's hill claiming method- Descent methods-Steepest descent, conjugate gradient, quasi - Newton method.	**12**
UNIT 4:	**- Hrs**
Constrained optimization Techniques- Necessary and sufficient conditions – Equality and inequality, constraints-Kuhn-Tucker conditions-Gradient projection method-cutting plane method- penalty function method.	**12**

Textbooks

1. B. V. D. Babu, A. K. M. N. Islam, and P. C. Babu. Optimization Techniques and Applications. 1st ed., Springer, 2022.
2. J. K. Gupta. Optimization Techniques in Operations Research. 3rd ed., Sultan Chand & Sons, 2020.
3. S. S. Rao. Engineering Optimization: Theory and Practice. 5th ed., Wiley, 2019.
4. K. Deb. Optimization for Engineering Design: Algorithms and Examples. 2nd ed., Prentice Hall, 2019.

To be effective from year-2024

Reference Books

1. R. L. S. Rardin. Optimization in Operations Research. 1st ed., Pearson, 2021.
2. A. Ravindran, D. T. Phillips, and J. J. Solberg. Operations Research: Principles and Practice. 2nd ed., Wiley, 2020.
3. F. S. Hillier and G. J. Lieberman. Introduction to Operations Research. 10th ed., McGraw-Hill Education, 2015.

COURSE OUTCOMES(CO):

CO1: Students will be able to formulate and solve linear programming problems using methods like the simplex method, Big-M method, and dual simplex method.

CO2: Students will acquire skills in applying one-dimensional optimization techniques, including Fibonacci, golden section, and interpolation methods for function optimization.

CO3: Students will demonstrate proficiency in using multi-dimensional optimization techniques, including random search, pattern search, and descent algorithms for solving complex problems.

CO4: Students will understand and apply constrained optimization techniques, including the Kuhn-Tucker conditions, gradient projection, and penalty function methods to engineering and real-world problems.

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
4	2	2	2	2	3	1	1	1	3	3	2	2

To be effective from year-2024

Program Elective-II

COURSE TITLE: Cluster and Grid Computing

Course Code:				**CSE-241051EAC**		**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

- Understand the foundational concepts and technologies involved in Grid computing, including the structure and standards governing large data grids and web services.
- Explore the Open Grid Services Architecture (OGSA) and Web Services Resource Framework (WSRF) in the context of resource distribution, management, and data choreography.
- Analyze the architecture, categories, and middleware components of cluster computing, with an emphasis on protocols, interconnection, and resource management.
- Learn how to set up and manage high-availability clusters, examining failure recovery mechanisms and advanced cluster configurations through case studies.

Course Content	TEACHING HOURS
UNIT 1: Introduction to Cluster and Grid Computing and Web Services Architecture	**14 Hrs**
Introduction: Cluster and Grid computing, Meta-computing, Web services and Grid Computing, e-Governance and the Grid Technologies and Architectures for Grid Computing: Issues in Data Grids, Functional requirements in Grid Computing, Standards for Grid Computing, Recent technology trends in Large Data Grids. Web Services and the Service Oriented Architecture: Service Oriented Architecture, SOAP and WSDL, Creating Web Services, Server Side.	
UNIT 2: Open Grid Services Architecture (OGSA), Web Services Resource Framework (WSRF), and Grid Databases	**14 Hrs**
OGSA and WSRF: OGSA for Resource Distribution, Stateful Web Services in OGSA, WSRF, WSRF Specification, Globus Toolkit: History, version, Applications, Approaches and Benefits, Infrastructure Management, Monitoring and Discovery, Security, Data Choreography and Coordination, GT4 Architecture, GT4 Containers. The Grid and Databases: Requirements, Storage Request Broker, Integration of Databases with the Grid, Architecture of OGSADAI for offering Grid Database services.	
UNIT 3: Cluster Computing Architecture and Middleware	**14 Hrs**
Cluster Computing: Approaches to Parallel Computing, Definition and Architecture of a Cluster, Categories of clusters. Cluster Middleware: Levels and Layers of Single System Image, Design objectives, Resource Management and Scheduling, Cluster programming Environment and Tools. Networking, Protocols & I/O for clusters: Networking and Interconnection/Switching Devices, Design Issues, Design Architecture, HiPPI, ATM, Myrinet, Memory Channel	

UNIT 4: Cluster Setup, Administration, and High-Availability Clusters	14 Hrs
Setting Up and Administering a Cluster: Setup of simple cluster, setting up nodes, clusters of clusters, System monitoring, Global Clocks Sync. Cluster Technology for High Availability: High availability clusters, high availability parallel computing, types of failures and errors, cluster architectures and configurations for high availability, Failure/Recovery clusters. Recent trends: technologies and attributes in Cluster and Grid computing. Case study of various cluster architectures, load balancing and scheduling policies.	

Textbooks

1. Grid and Cluster Computing by C.S.R. Prabhu, PHI.

Reference Books

1. **Grid and Cloud Computing: Concepts and Practical Applications** by Katarina Stanoevska-Slabeva, Thomas Wozniak, and Santi Ristol, 1st Edition, 2010
2. **Grid Computing** by Joshy Joseph and Craig Fellenstein, 1st Edition, 2004 — Covers the architecture, infrastructure, and technologies of Grid computing, including practical aspects of Web Services and OGSA

COURSE OUTCOMES (CO):

CO1: Demonstrate an understanding of Grid computing and service-oriented architecture (SOA), explaining their impact on web services, e-Governance, and large data grids.

CO2: Apply the concepts of OGSA and WSRF to resource distribution and database integration within grid environments, and evaluate the infrastructure benefits provided by the Globus Toolkit.

CO3: Assess the components and functions of cluster middleware, identify suitable networking protocols, and describe different categories of clusters for various computational tasks.

CO4: Successfully set up and administer a basic cluster configuration, identifying and applying methods for ensuring high availability and load balancing in complex cluster systems

LEVEL OF CO-PO MAPPING TABLE

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

Lab Manual Topics: Cluster & Grid Computing and Web Services Architecture

Lab Topics

1) Set up a basic cluster simulation using Python's multiprocessing module and virtualization tools.
2) Simulate distributed task execution by splitting a computation across multiple processes.
3) Create a simple grid computing simulation that assigns tasks to simulated nodes.
4) Build a basic RESTful web service using Flask.
5) Develop a simple SOAP-based web service (optional exercise for SOAP/WSDL).
6) Implement a Python script to simulate resource distribution across grid nodes.
7) Demonstrate OGSA/WSRF concepts by simulating stateful web service behavior with Python.
8) Connect a simple database to a grid simulation for basic operations.
9) Develop a Python-based scheduler to simulate resource management and job assignment in a cluster.
10) Use Python socket programming to simulate inter-node communication in a cluster.
11) Create a script that monitors simulated node status and logs performance metrics.
12) Simulate a failover mechanism by reassigning tasks from a "failed" node to active nodes.
13) Implement a simple load balancing algorithm in Python for task distribution among cluster nodes.
14) Set up a mini "cluster lab" using Docker or VirtualBox and deploy your simulation scripts.
15) Conduct a mini case study by running your cluster/grid simulations, collecting results, and summarizing your findings in a brief report

Computing Resources

- **Operating Systems:** Windows, Linux (e.g., Ubuntu)
- **Programming Language:** Python
- **Virtualization Tools :** VirtualBox or Docker

To be effective from year-2024

(162)

COURSE TITLE: Parallel Programming Tools and Model							
Course Code:				**CSE-241052EAC**		**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

- Classify parallel architectures parameters that are essential for the classification of modern parallel processing systems.
- Describe the methodologies employed for synchronization and memory consistency and cache coherence in shared memory systems.
- Describe and compare the different types of interconnects employed in parallel processing systems.
- Outline and analyse the features of micro-architecture parallel systems such as superscalar, VLIW, vector, multithreading, CMP multi-core and tile processors.

Course Content	**TEACHING HOURS**
UNIT 1: Introduction to Parallel Computing Architectures and Hardware	**14 Hrs**
Introduction to Parallel Computing Architectures, parallel hardware/multi-cores, Processes and threads, Programming models: shared memory and message passing, Amdahl's Law. Introduction to parallel hardware: Multi-cores and multiprocessors, shared memory and message passing architectures, cache hierarchy and coherence, sequential consistency.	
UNIT 2: Parallel Software Development and Performance Optimization	**14 Hrs**
Introduction to parallel software: Steps involved in developing a parallel program, Dependence analysis, Domain decomposition, Task assignment: static and dynamic, Performance issues: 4C cache misses, inherent and artifactual communication, false sharing, computation-to-communication ratio as a guiding metric for decomposition, hot spots and staggered communication.	
UNIT 3: Shared Memory Parallel Programming and Synchronization	**14 Hrs**
Shared memory parallel programming: Synchronization Locks and barriers, Hardware primitives for efficient lock implementation, Lock algorithms, Relaxed consistency models, High-level language memory models (such Java and/or C++), Memory fences. Developing parallel programs with UNIX fork model: IPC with shared memory and message passing, UNIX semaphore and its all-or-none semantic. Developing parallel programs with POSIX thread library, Thread creation, Thread join, Mutex, Condition variables. Developing parallel programs with OpenMP directives: Parallel for, Parallel section, Static, dynamic, guided, and runtime scheduling, Critical sections and atomic operations, Barriers Reduction	

UNIT 4: GPU Programming and Advanced Parallel Models	14 Hrs
Introduction to GPU programming: GPU architecture, Introduction to CUDA programming, Concept of SIMD and SIMT computation, Thread blocks, Warps, Global memory, Shared memory, Thread divergence in control transfer. Unit 6: Recent trends in Parallel Programming Models and Paradigms. Case study of parallel hardware which include shared memory architecture and message passing architectures for efficient computing.	

Textbooks

1. Peter S Pacheco, An Introduction to Parallel Programming, Morgan Kaufmann,2011.

Reference Books

2. M Herlihy and N Shavit, The Art of Multiprocessor Programming Morgan Kaufmann, 2008.
3. JL Hennessy and DA Patterson, Computer Architecture: A Quantitative Approach,4th Ed., Morgan Kaufmann/Els India, 2006.

COURSE OUTCOMES (CO):

CO1: Describe the core concepts of parallel computing architectures, including multi-core processors, shared memory, and message-passing models, and analyze their implications on performance using Amdahl's Law.

CO2: Develop and optimize parallel software by applying domain decomposition, task assignment strategies, and performance metrics to manage communication and computation trade-offs.

CO3: Implement shared memory parallel programs using synchronization techniques such as locks, barriers, and semaphores, and evaluate performance implications within UNIX and POSIX environments.

CO4: Write and execute GPU-based parallel programs using CUDA, demonstrating an understanding of GPU architecture, memory management, and thread-level parallelism for efficient SIMD/SIMT computation.

LEVEL OF CO-PO MAPPING TABLE

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

Parallel Programming Tools and Models- Lab Manual

The lab should cover the following topics:

1. **Introduction to Parallel Computing** – Basics of parallel architectures, processes, and threads.
2. **Exploring Multi-core Processors** – Understanding multi-core and multiprocessor architectures.
3. **Process and Thread Creation** – Using POSIX threads (pthreads) to create and manage threads.
4. **Shared Memory vs. Message Passing** – Implementing communication between processes using shared memory and message passing models.
5. **Performance Analysis with Amdahl's Law** – Experimenting with parallel speedup and efficiency.
6. **Cache Hierarchy and Coherence** – Analyzing the impact of cache coherence on performance.
7. **Dependency Analysis and Domain Decomposition** – Implementing static and dynamic task assignment.
8. **Synchronization Using Locks and Barriers** – Implementing mutexes, condition variables, and barriers in pthreads.
9. **Memory Consistency and Fences** – Understanding relaxed consistency models in parallel execution.
10. **Parallel Programming Using UNIX Fork Model** – Implementing inter-process communication (IPC) using shared memory.
11. **Parallel Programming Using OpenMP** – Basics of OpenMP, including `parallel for` and scheduling techniques.
12. **Load Balancing and Scheduling Strategies** – Experimenting with static, dynamic, and guided scheduling in OpenMP.
13. **False Sharing and Performance Bottlenecks** – Analyzing false sharing and its impact on performance.
14. **Parallel Reduction and Critical Sections** – Implementing atomic operations and reduction techniques.
15. **Introduction to GPU Programming** – Basics of CUDA programming and GPU architecture.
16. **Memory Management in CUDA** – Working with global, shared, and local memory.
17. **Thread Organization in CUDA** – Implementing thread blocks, warps, and divergence control.
18. **Optimizing CUDA Performance** – Experimenting with memory coalescing and bank conflicts.
19. **Parallel Sorting Algorithms** – Implementing parallel sorting techniques (e.g., parallel quicksort, merge sort).
20. **Case Study: Real-world Parallel Computing Applications** – Analyzing parallel computing in scientific simulations, graphics, or machine learning.

Lab Setup Notes:

- **Programming Languages**: C/C++, Python
- **Compilers & IDEs**: GCC, Clang, NVCC (CUDA), Visual Studio Code, Eclipse CDT
- **Parallel Libraries**: POSIX Threads (pthreads), OpenMP, MPI, CUDA
- **Debugging & Profiling**: Valgrind, GDB, CUDA Debugger, NVIDIA Nsight
- **Virtualization & Simulation**: Docker (CUDA), VirtualBox, SimGrid
- **Operating System**: Linux (preferred), Windows (WSL), macOS

COURSE TITLE: Big Data Analytics

Course Code:			**CSE-241053EAC**		**Examination Scheme**	**T**		**P**
Total number of Lecture Hours:56 **Total number of Practical Hours:-**					**External**	**80**		
					Internal	**20**		
Lecture (L):	**4**	**Practicals(P):**	**4**	**Tutorial (T):**	**-**	**Total Credits**	**6**	

Course Objectives:

- Understand the significance and applications of big data across various industries, including marketing, healthcare, and finance.
- Explore the fundamentals of NoSQL databases, including different data models and advanced concepts such as sharding and replication.
- Gain proficiency in data management and analysis using Hadoop, including its architecture, HDFS, and data processing techniques.
- Learn MapReduce workflows and integrate NoSQL databases, such as HBase and Cassandra, with Hadoop for effective data analytics

Course Content	**TEACHING HOURS**
UNIT 1: Introduction to Big Data and Its Applications Across Industries	**-14 Hrs**
What is big data, why big data, convergence of key trends, unstructured data, industry examples of big data, web analytics, big data and marketing, fraud and big data, risk and big data, credit risk management, big data and algorithmic trading, big data and healthcare, big data in medicine, advertising and big data, big data technologies, introduction to Hadoop, open source technologies, cloud and big data, mobile business intelligence, Crowd sourcing analytics, inter and trans firewall analytics.	
UNIT 2: NoSQL Databases and Data Processing Techniques	**- 14 Hrs**
Introduction to NoSQL, aggregate data models, aggregates, key-value and document data models, relationships, graph databases, schemaless databases, materialized views, distribution models, sharding, master-slave replication, peer peer replication, sharding and replication, consistency, relaxing consistency, version stamps, map-reduce, partitioning and combining, composing map-reduce calculations.	
UNIT 3: Data Management and Analysis with Hadoop	**-14 Hrs**
Data format, analyzing data with Hadoop, scaling out, Hadoop streaming, Hadoop pipes, design of Hadoop distributed file system (HDFS), HDFS concepts, Java interface, data flow, Hadoop I/O, data integrity, compression, serialization, Avro, file-based data structures	
UNIT 4: MapReduce Workflows and NoSQL Integration	**-14 Hrs**
MapReduce workflows, unit tests with MRUnit, test data and local tests, anatomy of MapReduce job run, classic Map-reduce, YARN, failures in classic Map-reduce and YARN, job scheduling, shuffle and sort, task execution, MapReduce types, input formats, output formats. Hbase, data model and implementations, Hbase clients, Hbase examples, praxis. Cassandra, Cassandra data model, Cassandra examples, Cassandra clients, Hadoop integration.	

To be effective from year-2024

Lab Manual

Lab should be covered on the following topics:

- Analyze datasets of varying size and structure (CSV, JSON, unstructured text). Identify volume, velocity, and variety characteristics.
- Use web analytics tools (e.g., Google Analytics) on sample data to perform customer segmentation and understand behavior patterns.
- Apply data mining techniques on a financial dataset to detect anomalies indicative of fraud (e.g., clustering for anomaly detection).
- Explore a healthcare dataset (e.g., patient records) to identify trends, assess risk factors, and understand applications in predictive analytics.
- Set up a Hadoop cluster on a local machine or cloud environment. Store and retrieve data on HDFS, exploring storage distribution.
- Use MongoDB or Redis to create and manipulate key-value pairs and document structures. Experiment with CRUD operations and indexing.
- Build a basic graph database with Neo4j. Load and query data with relationships (e.g., social network connections) to see graph traversal.
- Implement sharding and replication in MongoDB or Cassandra. Evaluate how sharding improves performance and replication enhances reliability.
- Run MapReduce tasks on NoSQL data, such as counting document types or aggregating values in a collection.
- Upload and retrieve files in HDFS. Experiment with replication factors, block sizes, and permissions to see their effect on storage and performance.
- Develop MapReduce workflows for a specific task (e.g., log parsing), set up unit tests with MRUnit, and perform local tests.
- Set up Hbase and Cassandra clients with Hadoop. Create data models and run queries, observing data distribution and access.

Program Elective-III

COURSE TITLE: Distributed Database

Course Code:				**CSE-242041EAC**		**Examination Scheme**		
Total number of Lecture Hours:						**External**	**80**	
						Internal	**20**	
Lecture (L):	**4**	**Practicals(P):**	**-**	**Tutorial (T):**	**-**	**Total Credits**		**4**

Course Objectives

- Understand the fundamental concepts and challenges of distributed database systems, including relational database principles and integrity constraints.
- Explore the architecture of distributed DBMSs, design strategies, and issues related to data fragmentation, allocation, and security in distributed systems.
- Learn the fundamentals of query processing and transaction management in a distributed environment, emphasizing concurrency control and query optimization.
- Gain insight into the structure and functioning of parallel and distributed object-oriented databases, addressing database interoperability, query processing, and recent performance-enhancing trends.

Course Content	**TEACHING HOURS**
UNIT 1: Introduction to Distributed Data Processing and Relational Database Systems	**16 Hrs**
Unit 1: Introduction: Distributed Data processing, Distributed database system (DDBMS), Promises of DDBMSs, Complicating factors and Problem areas in DDBMSs, Overview Of Relational DBMS Relational Database concepts, Normalization, Integrity rules, Relational Data Languages, Relational DBMS.	
UNIT 2: Distributed DBMS Architecture and Database Design	**14 Hrs**
Unit 2: Distributed DBMS Architecture: DBMS Standardization, Architectural models for Distributed DBMS, Distributed DBMS Architecture. Distributed Database Design: Alternative design Strategies, Distribution design issues, Fragmentation, Allocation. Semantic Data Control: View Management, Data security.	
UNIT 3: Query Processing and Transaction Management in Distributed Databases	**14 Hrs**
Overview of Query Processing: Query processing problem, Objectives of Query Processing, Complexity of Relational Algebra operations, characterization of Query processors, Layers of Query Processing. Introduction to Transaction Management: Definition of Transaction, Properties of transaction, types of transaction. Distributed Concurrency Control: Serializability theory, Taxonomy of concurrency control mechanisms, locking bases concurrency control algorithms.	
UNIT 4: Parallel and Distributed Object Database Management Systems	**14 Hrs**
Parallel Database Systems: Database servers, Parallel architecture, Parallel DBMS techniques, Parallel execution problems, Parallel execution for hierarchical architecture. Distributed Object Database Management systems: Fundamental Object concepts and Object models, Object distribution design. Architectural issues, Object management, Distributed object storage, Object query processing. Transaction management. Database Interoperability: Database Integration, Query processing. Recent approaches, models and current trends in improving the performance of Distributed Database.	

To be effective from year-2024

Textbooks

3. Principles of Distributed Database Systems, Second Edition, M. Tamer Ozsu Patrick Valduriez

Reference Books

3. Distributed Databases principles and systems, Stefano Ceri, Giuseppe Pelagatti, Tata McGraw Hill.

COURSE OUTCOMES (CO):

CO1: Describe the core principles of distributed database systems (DDBMS), including the architecture of relational databases, normalization, and the promises and challenges of distributed data processing.

CO2: Evaluate distributed database architectural models and design strategies, applying knowledge of fragmentation, allocation, and security to ensure effective database distribution.

CO3: Apply transaction management principles and concurrency control techniques to maintain database integrity and efficiency in distributed environments.

CO4: Analyze the architecture and processing techniques of parallel and distributed object databases, exploring current trends to enhance performance and interoperability in complex database systems.

LEVEL OF CO-PO MAPPING TABLE

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

COURSE TITLE: Concurrence, Parallelism and Distributed System

Course Code:				**CSE-242042EAC**		**Examination Scheme**	**T**	**P**
Total number of Lecture Hours:56 **Total number of Practical Hours:-**						**External**	**80**	
						Internal	**20**	
Lecture (L):	**4**	**Practicals(P):**	**-**	**Tutorial (T):**	**-**	**Total Credits**	**4**	

Course Objectives:

- To understand the fundamental concepts of distributed computing, including architectural models, distributed operating systems, and communication protocols.
- To explore various synchronization mechanisms, process concepts, and security aspects within distributed systems.
- To analyze and design distributed databases, focusing on data storage, query processing, transaction management, and concurrency control.
- To introduce parallel processing techniques, including the design of parallel algorithms and parallel databases, and to evaluate parallel query processing methods.

Course Content	TEACHING HOURS
UNIT 1: Fundamentals of Distributed Computing and Communication Systems	**-14 Hrs**
Fundamentals of Distributed Computing: Architectural models for distributed and mobile computing systems, Basic concepts in distributed computing. Distributed Operating Systems: Overview, network operating systems, Distributed file systems, Middleware, client/server model for computing. Communication: Layered protocols, RPC, RMI, Remote objects. Basic Algorithms in Message Passing Systems, Leader Election in Rings, and Mutual Exclusion in Shared Memory, Message Passing, PVM and MPI.	
UNIT 2: Process Concepts, Synchronization, and Security in Distributed Systems	**- 14 Hrs**
Process Concepts: Threads, Clients and Servers, Code migration, Agent based Synchronization: Clock synchronization, Logical clocks, Election algorithms, Mutual exclusion, Distributed transactions, Naming concepts, Security in distributed systems, Distributed objects, CORBA, Distributed COM.	
UNIT 3: Distributed Databases and Transaction Management	**-14 Hrs**
Distributed Databases: Distributed Data Storage, Fragmentation & Replication, Transparency, Distributed Query Processing and Optimization, Distributed Transaction Modelling and concurrency Control, Distributed Deadlock, Commit Protocols.	
UNIT 4: Parallel Processing and Parallel Databases	**-14 Hrs**
Parallel Processing: Basic Concepts: Introduction to parallel processing, Parallel processing terminology, Design of parallel algorithms, Design of Parallel Databases, Parallel Query Evaluation.	

Textbooks

- **"Distributed Systems: Concepts and Design"**, George Coulouris, Jean Dollimore, Tim Kindberg, and Gordon Blair, 5th Edition (2011)
- **"Principles of Distributed Database Systems"**, M. Tamer Özsu, Patrick Valduriez, 4th Edition (2019), Springer

Reference Books

- **"Distributed Operating Systems: Concepts and Design", Pradeep** K. Sinha, 2nd Edition (2004)
- **Distributed Systems: Principles and Paradigms", Andrew** S. Tanenbaum, Maarten van Steen, 2nd Edition (2007).
- **Distributed Computing: Principles, Algorithms, and Systems"**, Ajay D. Kshemkalyani, Mukesh Singhal, 1st Edition (2008), Cambridge University Press

COURSE OUTCOMES (CO):

CO1: Understand and explain the core concepts of distributed computing, distributed operating systems, and communication protocols.

CO2: Demonstrate the ability to apply synchronization techniques, process management, and security measures in distributed systems.
CO3: Analyze and implement distributed databases, including data fragmentation, replication, query processing, and transaction management.
CO4: Design and evaluate parallel processing algorithms and parallel databases for efficient data handling and query evaluation.

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	1	1	2	2	2	1	3
2	2	3	3	3	3	2	2	2	2	2	1	2
3	3	3	3	2	3	2	2	2	2	2	2	3
4	3	2	3	2	3	1	1	2	3	3	2	2

To be effective from year-2024

COURSE TITLE: HPC Architecture, and Ecosystem

Course Code:				**CSE-242043EAC**		**Examination Scheme**	
Total number of Lecture Hours: 56						**External**	**80**
						Internal	**20**
Lecture (L):	**4**	**Practical (P):**	-	**Tutorial (T):**	-	**Total Credits**	**4**

Course Objectives

- Introduce the fundamentals of parallel system organization and explore message passing models with a specific emphasis on MPI programming.
- Examine problems suitable for parallelization, with a focus on embarrassingly parallel problems, problem decomposition, graph partitioning, and load balancing.
- Introduce shared memory programming models, including OpenMP, and explore their applications in scientific computing and parallel languages.
- Explore recent trends in parallel programming with OpenMP, including advanced programming techniques and applications in scientific computing.

Course Content	TEACHING HOURS
UNIT 1: Introduction to Parallel System Organization and Message Passing	**17 Hrs.**
Overview of parallel system organization; Introduction to message passing and MPI programming.	
UNIT 2: Decomposition and Load Balancing in Parallel Computing	**14 Hrs.**
Embarrassingly parallel problems; Problem decomposition, graph partitioning, and load balancing.	
UNIT 3: Shared Memory Systems and OpenMP Programming	**14 Hrs.**
Introduction to shared memory and OpenMP programming. Examples of scientific computing; Parallel Languages.	
UNIT 4: Advanced OpenMP Techniques and Applications in Scientific Computing	**14 Hrs.**
Recent trends in OpenMP programming, application areas of scientific computing.	

Textbooks

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

Reference Books

2. Scientific Parallel Computing by Scott, Clark, and Bagheri.
3. Using OpenMP: Portable Shared Memory Parallel Programming by Chapman, Jost, and van derPas.

COURSE OUTCOMES (CO):

After completion of course, students would be able to:

CO1: Demonstrate a strong understanding of the key concepts and architecture of parallel systems.

CO2: Implement basic MPI functions for communication between processes in a distributed

memory system.
CO3: Identify and solve problems that are inherently parallel, requiring minimal inter-process communication. Apply efficient decomposition strategies for embarrassingly parallel problems.
CO4: Use techniques like graph partitioning and load balancing to optimize parallel programs.

LEVEL OF CO-PO MAPPING TABLE

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

To be effective from year-2024