INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)

Dundigal - 500 043, Hyderabad, Telangana

COURSE CONTENT

HIGH PERFORMANCE COMPUTING								
VI Semester: CSE (CS)								
Course Code	**Category**	**Hours / Week**			**Credits**	**Maximum Marks**		
ACCC20	Elective	L	T	P	C	CIA	SEE	Total
		3	0	0	3	40	60	100
Contact Classes: 48	**Tutorial Classes: Nil**	**Practical Classes: Nil**				**Total Classes: 48**		
Prerequisite: Operating systems								

I. COURSE OVERVIEW:

This course provides a comprehensive introduction to High-Performance Computing (HPC), covering the principles, architectures, and techniques used to solve computationally intensive problems. Students will explore parallel computing models, distributed computing, and GPU acceleration to optimize performance in scientific computing, engineering, and data-intensive applications. The course also includes hands-on experience with MPI (Message Passing Interface), Open MP, and CUDA programming.

II. COURSES OBJECTIVES:

The students will try to learn

I. *The* parallel computing principles, algorithm design methodologies including partitioning, task scheduling, load balancing, and structured parallel models used in modern parallel computers.

II. *The* performance modeling fundamentals such as scalability, overhead, bandwidth, efficiency, and to understand algorithmic case studies like shortest path solutions derived from methods including Floyd's algorithm.

III. *The* C/C++-based concurrency constructs, synchronization, locality, communication, and remote operations supported in development environments provided by vendors such as IBM.

IV. *The* message-passing programming models and practical bindings using libraries like MPI and performance analysis ecosystems such as Pablo.

III. COURSE OUTCOMES:

At the end of the course students should be able to:

CO1 Design and implement parallel algorithms (sorting, graph, reduction, network-aware methods) while selecting suitable partitioning and scheduling strategies.

CO2 Model and evaluate parallel system performance using parameters including scalability, overhead, bandwidth, speedup, and efficiency.

CO3 Develop concurrent programs in C/C++ using synchronization, mutual exclusion, and data transfer mechanisms with locality-aware thread/processor placement.

CO4 Apply message-passing programming using C and Fortran bindings with global and asynchronous operations through libraries such as MPI.

CO5 Analyze interconnection network impact including efficiency trade-offs in hypercube-style communication using models like Hypercube algorithms.

CO6 Use structured performance profiling and analysis workflows through tools such as Pablo, instrumentation traces, data collection, transformation, and visualization.

IV. COURSE CONTENT:

MODULE I: DESIGN OF PARALLEL ALGORITHMS (10)

Parallel computers and computation, a parallel machine model, a parallel programming model , parallel algorithm examples, partitioning, communication, agglomeration, mapping, load balancing algorithms, task, scheduling algorithms, case studies, random numbers generation, hypercube algorithms, vector reduction, matrix transposition, merge sort.

MODULE II: APPROACHES TO PERFORMANCE MODELING (09)

A quantitative basis for design, defining performance, approaches to performance modeling, developing models, performance parameters, time, scalability, overheads, bandwidth, efficiency, speed, interconnection networks, input/output; Case study: Shortest path algorithms, floyd's algorithm, dijkstra's algorithm, modular design review, modularity and parallel computing performance analysis; Case study: Convolution, tuple space and matrix multiplication.

MODULE III: PARALLEL COMPUTING DEVELOPMENT TOOLS: (10)

C++ review, C, C++ introduction, concurrency, locality, processor objects, global pointers thread placement, communication, remote operations.

Synchronization, mutual exclusion, data transfer functions, asynchronous communication, determinism, mapping, modularity performance issues.

MODULE IV: PARALLEL COMPUTING DEVELOPMENT TOOLS (09)

Fortran M, concurrency, communication, unstructured communication, asynchronous communication, determinism, argument passing, mapping, modularity, high performance Fortran, data parallelism, concurrency, data distribution, dummy arguments and modularity other HPF features, performance issues

MODULE V: ADD ON TOOLS FOR DEVELOPMENT (09)

Message passing libraries: The MPI programming model, MPI basics, C and Fortran language bindings with MPI functions, global operations, asynchronous communication, modularity, other MPI features, performance issues, performance tools, performance analysis, data collection, data transformation and visualization tools, paragraph, upshot—pablo, gauge, paraide, IBM's parallel environment, AIMS, custom tools.

V. TEXT BOOKS:

1. Ion Foster, ―Designing and Building Parallel Programs‖, Addison Wesley, 1st Edition, 2003.

VI. REFERENCE BOOKS:

2. Arjen Markus, "Modern Fortran in Practice", Cambridge University Press, 1 st Edition, 2012.
3. Charles H. Koelbe, "High Performance Fortran Handbook", MIT Press, 1 st Edition, 1993.
4. Michael J. Quinn, "Parallel Programming in C with MPI and Open MPI", Tata McGraw-Hill Publishing Company Ltd, 1st Edition, 2003.

VII. ELECTRONICS RESOURCES:

1. http://www.drdobbs.com/parallel/designing-parallel-algorithms-part-1/223100878.
2. http://searchcloudapplications.techtarget.com/tip/How-to-use-application-performance-modelingtechniques.
3. https://computing.llnl.gov/tutorials/parallel_comp/.

VIII. MATERIALS ONLINE

1. Course template
2. Tutorial question bank
3. Tech-talk topics
4. Open-ended experiments
5. Definitions and terminology
6. Assignments
7. Model question paper – I
8. Model question paper – II

9. Lecture notes
10. PowerPoint presentation
11. E-Learning Readiness Videos (ELRV)