Course code.	Course Name	L-T-P - Credits	Year of Introduction
CS372	**HIGH PERFORMANCE COMPUTING**	**3-0-0-3**	**2016**

Pre-requisites : CS202 Computer Organization and Architecture

Course Objectives

- To introduce the concepts of Modern Processors.
- To introduce Optimization techniques for serial code.
- To introduce Parallel Computing Paradigms.
- To introduce Parallel Programming using OpenMP and MPI.

Syllabus

Modern processors - pipelining-superscalarity-multicore processors- Mutithreaded processors- vector processors- basic optimization techniques for serial code - taxonomy of parallel computing paradigms- shared memory computers- distributed-memory computers- Hierarchical Systems-networks- basics of parallelization - data parallelism - function parallelism- Parallel scalability-shared memory parallel programming with OpenMp - Distributed-memory parallel programming with MPI.

Expected Outcome

The students will be able to

i. appreciate the concepts used in Modern Processors for increasing the performance.
ii. appreciate Optimization techniques for serial code.
iii. appreciate Parallel Computing Paradigms.
iv. identify the performance issues in Parallel Programming using OpenMP and MPI.

Text Book

1. Georg Hager, Gerhard Wellein, Introduction to High Performance Computing for Scientists and Engineers, Chapman & Hall / CRC Computational Science series, 2011.

References

1. Charles Severance, Kevin Dowd, High Performance Computing, O'Reilly Media, 2nd Edition, 1998.
2. Kai Hwang, Faye Alaye Briggs, Computer Architecture and Parallel Processing, McGraw Hill, 1984.

Course Plan

Module	Contents	Hours	End Sem. Exam Marks
I	Modern Processors : Stored Program Computer Architecture- General purpose cache- based microprocessor-Performance based metrics and benchmarks- Moore's Law- Pipelining- Superscalarity- SIMD- Memory Hierarchies Cache- mapping- prefetch- Multicore processors- Mutithreaded processors- Vector Processors- Design Principles- Maximum performance estimates- Programming for vector architecture.	**07**	**15%**

II	Basic optimization techniques for serial code : scalar profiling-function and line based runtime profiling- hardware performance counters- common sense optimizations- simple measures, large impact- elimination of common subexpressions- avoiding branches-using simd instruction sets- the role of compilers - general optimization options- inlining - aliasing- computational accuracy-register optimizations- using compiler logs- c++ optimizations -temporaries- dynamic memory management- loop kernels and iterators data access optimization: balance analysis and light speed estimates- storage order- case study: jacobi algorithm and dense matrix transpose.	**07**	**15%**
FIRST INTERNAL EXAM			
III	Parallel Computers : Taxonomy of parallel computing paradigms-Shared memory computers- Cache coherance- UMA - ccNUMA-Distributed-memory computers- Hierarchical systems- Networks-Basic performance characteristics- Buses- Switched and fat- tree networks- Mesh networks- Hybrids - Basics of parallelization - Why parallelize - Data Parallelism - Function Parallelism- Parallel Scalability- Factors that limit parallel execution- Scalability metrics- Simple scalability laws- parallel efficiency - serial performance Vs Strong scalability- Refined performance models-Choosing the right scaling baseline- Case Study: Can slow processors compute faster- Load balance.	**07**	**15%**
IV	Distributed memory parallel programming with MPI : message passing - introduction to MPI – example - messages and point-to-point communication - collective communication – nonblocking point-to-point communication- virtual topologies - MPI parallelization of Jacobi solver- MPI implementation - performance properties	**08**	**15%**
SECOND INTERNAL EXAM			
V	Shared memory parallel programming with OpenMp : introduction to OpenMp - parallel execution - data scoping- OpenMp work sharing for loops- synchronization - reductions - loop scheduling - tasking - case study: OpenMp- parallel jacobi algorithm- advanced OpenMpwavefront parallelization- Efficient OpenMP programming: Profiling OpenMP programs - Performance pitfalls-Case study: Parallel Sparse matrix-vector multiply.	**08**	**20%**
VI	Efficient MPI programming : MPI performance tools-communication parameters- Synchronization, serialization, contention- Reducing communication overhead- optimal domain decomposition- Aggregating messages – Nonblocking Vs Asynchronous communication- Collective communication-Understanding intra-node point-to-point communication.	**08**	**20%**
END SEMESTER EXAM			

Question Paper Pattern

1. There will be *five* parts in the question paper – A, B, C, D, E
2. Part A
   a. Total marks : 12

   b. *Four* questions each having *3* marks, uniformly covering modules I and II; All*four* questions have to be answered.
3. Part B
   a. Total marks : 18
   b. *Three* questions each having *9* marks, uniformly covering modules I and II; *Two* questions have to be answered. Each question can have a maximum of three subparts.
4. Part C
   a. Total marks : 12
   b. *Four* questions each having *3* marks, uniformly covering modules III and IV; All*four* questions have to be answered.
5. Part D
   a. Total marks : 18
   b. *Three* questions each having *9* marks, uniformly covering modules III and IV; *Two* questions have to be answered. Each question can have a maximum of three subparts
6. Part E
   a. Total Marks: 40
   b. *Six* questions each carrying 10 marks, uniformly covering modules V and VI; *four* questions have to be answered.
   c. A question can have a maximum of three sub-parts.
7. There should be at least 60% analytical/numerical questions.