SILVER OAK UNIVERSITY

Engineering and Technology (M.Tech.)

Computer Engineering (Software Engineering)
Subject Name: High Performance Computing
Subject Code:
Semester: II

Prerequisite:
Basic Knowledge of Procedure and Object-Oriented Programming.

Objective:
Objective of this course is to teach the students high performance computing concepts with which they can speed up their computational processes as well as can manage operations by utilizing resources properly using related libraries and tools.

Teaching and Examination Scheme:

Teaching Scheme			Credits	Evaluation Scheme				Total Marks
L	T	P	C	Internal		External		
				Th	Pr	Th	Pr	
3	0	2	4	30	20	70	30	150

Content:

Unit No.	Course Contents	Teaching Hours	Weightage %
1	**Modern Processors:** Stored Program Computer Architecture, Architecture of microprocessor based on cache- Performance based metrics and benchmarks, Moore's Law, Pipelining, Super scalarity, SIMD, Different classes of Memory- cache, mapping, pre-fetch, Introduction to different types of processor such as Multicore processors, Multithreaded processors, Vector Processors	5	14
2	**Requirements and General Issues:** Scalable parallel computer Architectures, A cluster computer and its Architecture, clusters classifications, Commodity components for clusters, Network services/Communication SW, Cluster middleware and single system Image(SSI), Resource Management and Scheduling(RMS),Programming environments and Tools, Representative cluster Systems. High speed Networks: Design issues, Fast Ethernet, High Performance parallel interface (HPPI), Asynchronous transfer mode (ATM), Myrinet.	7	18

3	**Parallel Computers:** Taxonomy of parallel computing paradigm, Different types of memory computers such as Shared memory computers, Distributed memory computers, Hierarchical systems, Basics of parallelization.	7	18
4	**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 mechanism	7	18
5	**Distributed memory parallel programming with MPI:** Brief introduction to MPI such as messages and point-to-point communication – collective communication – Non-blocking point-to-point communication- virtual topologies, MPI parallelization of Jacobi solver- MPI implementation - performance properties .MPI performance tools, communication parameters, Synchronization, serialization.	7	18
6	**Advance Computing:** Introduction to Petascale computing, Optical Computing, Quantum computing and its issues	5	14

Course Outcome:

Sr. No.	CO statement	Unit No
CO-1	Analyze the functionality of Modern Processor	1
CO-2	Comprehend and implement various optimization techniques for serial code.	2
CO-3	Design the concept of parallel computing paradigm.	3 & 4
CO-4	Understanding the concepts of Distributed memory parallel programming with MPI	5
CO-5	Concepts of advance computing	6

Teaching & Learning Methodology: -
Problem identification
Problem based learning
Competency based learning

List of Experiments/Tutorials:

Sr. No.	Title
1	UsingDivide and Conquer Strategies design a class for Concurrent Quick Sort using C++.
2	Perform concurrent ODD-Even Merge sort using HPC infrastructure using Python/ Scala/ Java/ C++.
3	Write a program to calculate sum of an array elements using direct scheduling.
4	Write a program for prime number that includes MPI calls for parallel processing.
5	Write a program to perform one to one message passing.
6	Write a program to demonstrate MPI_Send() and MPI_Recv().
7	Write a program to demonstrate MPI group functions.
8	Write a program that performs blocking message passing.
9	Write a program which calculates Sum of numbers from 1 to 10, by dividing the job into two processes (parent and one child, Using Shared Memory)
10	Write a program in which two or more threads are available, one thread invokes function funcA() and another thread calls funcB() using functionalities of omp.h and pragma directive.
11	Introduction and Installation of CUDA toolkit.
12	Create your first CUDA program.
13	Perform Fourier Transformation using wrap level programming in CUDA.

Major Equipment:
Laptop/Personal Computer, For message passing it may require small network of computers.

Books Recommended: -
1. Georg Hager, Gerhard Wellein, Introduction to High Performance Computing for Scientists and Engineers, Chapman & Hall / CRC Computational Science series
2. Gene Wagenbreth and John Levesque, High performance Computing: Programming and Application,CRC press, Taylor and francis group
3. MaciejBrodowicz, Matthew Anderson, and Thomas Sterling, High Performance Computing: Modern Systems and Practices,Morgankaufmann publishers

List of Open Source Software/learning website:
https://nptel.ac.in/courses/106/108/106108055/
https://computing.llnl.gov/tutorials/mpi/#Exercise1
https://www.sciencedirect.com/topics/computer-scienc