High Performance Computing

Course Code: CS72 **Credits: 3:0:0:1**

Unit I

Introduction to High–Performance Computers, Memory Hierarchy, CPU Design: Reduced Instruction Set Computers, Multiple–Core Processors, Vector Processors.

Self-Study: Parallel Semantics, Distributed Memory Programming.

Unit II

Programming Shared Address Space Platforms: Thread Basics, Why Threads? The POSIX Thread API, Thread Creation and Termination, Synchronization Primitives in Pthreads, Controlling Thread and Synchronization Attributes, Thread Cancellation, Composite Synchronization Constructs.

Self-Study: Tips for Designing Asynchronous Programs, OpenMP: a Standard for Directive Based Parallel Programming

.

Unit III

Programming using the Message-Passing Paradigm: Principles of Message-Passing Programming, The Building Blocks: Send and Receive Operations, MPI: the Message Passing Interface, Topologies and Embedding, Overlapping Communication with Computation, Collective Communication and Computation Operations.

Self-Study: Groups and Communicators.

Unit IV

Introduction: GPUs as Parallel Computers, Architecture of a Model GPU, Why More Speed or Parallelism? Parallel Programming Languages and Models, Overarching Goals. History of GPU Computing: Evolution of Graphics Pipelines, GPU Computing. Introduction to CUDA: Data Parallelism, CUDA Program Structure, A Matrix-Matrix Multiplication Example, Device Memories and Data Transfer.

Self-Study: Kernel Functions and Threading.

Unit V

CUDA Threads: CUDA Thread Organization, Using blockIdx and threadIdx, Synchronization and Transparent Scalability, Thread Assignment, Thread Scheduling and Latency Tolerance. CUDA Memories: Importance of Memory Access Efficiency, CUDA Device Memory Types, A Strategy for Reducing Global Memory Traffic.

Self-Study: Memory as a limiting Factor to Parallelism.