Elective course IDC xxx

Course Title: GPU Programming for Scientific Applications

Course No: IDC 600 level

Proposers : Mahendra K. Verma (PHY) + Swarnendu Biswas (CSE) + Rajesh Ranjan (Aerospace Engg.)

About the course: This course on GPU programming for scientific applications will enable students to harness massive parallelism for compute-intensive tasks. The course will provide an overview of GPU architectures, multithreading, and programming models, focusing on tools such as OpenMP, CuPy, JAX, Warp, and native CUDA, along with MPI for multi-GPU systems. Algorithms for key scientific applications —including Fast Fourier Transforms (FFT), spectral solvers, electromagnetic (EM) solvers, molecular dynamics, fluid dynamics, and parallel matrix solvers—will be covered alongside acceleration techniques. These skills will be invaluable to students across all science and engineering disciplines looking to scale their computational research.

Participating Departments for floating the course (possible ones): Aerospace engineering, Computer Science and Engineering, Kotak School of Sustainability, Mathematics and Statistics, Mechanical Engineering, Physics,

Departments from which students can take course for credit: Students from any department of the institute

Units: 3-0-0-0-9 [3 lectures, 9 credits]

Prerequisite: Basic knowledge in computer programming and mathematics

Who can take the course: Ph. D., M. Sc., and Advanced UG students.

Course Contents:

1. **GPU Architecture & Programming Framework**
    - Limitations of CPU in HPC
    - GPU design philosophies
    - GPU execution model, memory hierarchy, and compute units
    - GPU Paradigm: Threads, Blocks, and Grids; SIMT (Single Instruction Multiple Threads) framework
    - What makes GPUs fast?
2. **Python for HPC, Array Computing & Scientific Applications (CuPy)**
    - Overview and comparison of programming languages for HPC
    - Array computing and GPU acceleration using CuPy
3. **NVIDIA Warp: Programming, Profiling & Applications**
    - Fundamentals of Warp kernel programming
    - Performance profiling and bottleneck analysis
    - Implementing HPC algorithms and scientific benchmarks in Warp: Matrix Solvers, Molecular Dynamics, Working with large arrays

4. **Accelerated Python & Compiled Frameworks (JAX, PyTorch, Numba & Mojo)**
    - Vectorization, automatic differentiation, and GPU acceleration with JAX and PyTorch
    - JIT compilation for GPUs using Numba
    - Next-generation high-performance compiled computing with Mojo
5. **Core & Advanced CUDA Programming (C/C++)**
    - Kernel launching, thread hierarchy (grids, blocks, threads), and memory management
    - Memory optimization, shared memory utilization, and advanced CUDA kernels
6. **Distributed & Multi-GPU Computing with MPI (mpi4py)**
    - Point-to-point communication and non-blocking transfers
    - Collective communication, reductions, and multi-GPU coordination
    - Domain decomposition and parallel finite difference schemes
7. **Handling Large Data in Parallel & HDF5 Library**
    - I/O bottlenecks and data management in large-scale parallel simulations
    - Hierarchical data formatting using the HDF5 library (h5py)
    - Concurrent parallel read/write strategies across multi-GPU and multi-node systems
8. **Case Studies & Hands-on Exercises on Scientific Applications**
    - End-to-end implementation and performance tuning of real-world scientific codes
    - Case studies covering: Fast Fourier Transforms (FFT), spectral solvers, electromagnetic (EM) solvers, parallel matrix solvers, molecular dynamics, and fluid dynamics

The course will have major project component; the students would program on GPUs.

Overlapping Course: The course CS623 GPU Architecture and Programming offered in the CSE department delves deeply into GPU hardware and systems programming. In contrast, the present course focusses on applications. Our proposed course includes programming models Warp, Jax, PyTorch, Mumber etc., which are completely absent in CS623.

Textbooks and References:

1. J. L. Hennessy & D. A. Patterson: Computer Architecture: A Quantitative Approach, Morgan Kaufmann Publishers.
2. G. Zaccone. Python Parallel Programming Cookbook, Packt Publ. (2015)
3. D. Strorti and M. Yurtoglu, CUDA for engineers, Pearson Education India (2016)
4. W. W. Hwu, D. B. Kirk, and I. E. Hajj, Programming Massively Parallel Processors: A Hands-on Approach, Morgan Kaufmann Pub
5. Class notes

Signatures of the proposers:

		Rajesh Ranjan
Mahendra Verma	Swarnendu Biswas	Rajesh Ranjan

This Course is APPROVED/ NOT APPROVED: Convener, DPGC, Aerospace Engineering

This Course is APPROVED/ NOT APPROVED: Convener, DPGC, Comp. Sci. Engg.

This Course is APPROVED/ NOT APPROVED: Convener, DPGC, Kotak School of Sustainability Sarosh

This Course is APPROVED/ NOT APPROVED: Convener, DPGC, Physics