Course objective

With the rise of computationally expensive machine learning models and availability of cheap computing resources such as GPUs, accelerated computing can bring a huge potential to the field of data science and artificial intelligence. The goal of this course is to provide a solid foundation for developing, analyzing, and implementing parallel algorithms using GPU-accelerated computing, that is using a GPU in combination with a CPU.

Course outcomes

The first few lectures will cover the important aspects of computer architecture, algorithms and data structure required for this course. Subsequently the course aims to cover the important topics of GPU programming such as design of parallel algorithms, GPU and CPU architecture, data parallelism, CUDA programming model, GPU memory model, memory performance and optimization, parallel complexity analysis, performance modeling, and case studies focused on important parallel patterns supported by lab assignments. The course will also cover uses of the GPU in machine learning. Lab work will require significant programming. (Note: A working knowledge of the C programming will be very helpful.)

Course Contents

- Know-how of principles of parallel algorithm design and ability to program on heterogeneous computing systems using CUDA C.
- Ability to express data and instruction level parallelism in applications using CUDA. Understanding of important parallel patterns.
- Hands-on experience with the fundamental tools and techniques for accelerating/optimizing applications (C language) on GPUs with CUDA (taking into account processor architecture features).

References:

2. CUDA by Example: An Introduction to General-Purpose GPU Programming by Jason Sanders and Edward Kandrot

3. CUDA Programming: A Developer's Guide to Parallel Computing with GPUs (Applications of GPU Computing Series) by Shane Cook

4. Web material:  

**Module 1 - Introduction to Accelerated Computing:**

**Module 2 - GPU/CUDA Hardware and first CUDA program**
CPUs and GPUs. GPU nVIDIA architecture. Setting up CUDA. Compile and execute simple CUDA program structure, Timing execution time. Kernel functions and threading. Global memory and data transfer.

**Module 3 - GPU/CUDA programming model**
Basic CUDA program structure, kernel calls, threads, blocks, grid, thread addressing, predefined variables.

**Module 4 - CUDA based Design and Parallelization of Important Parallel Algorithms** - Scan, Reduction, Convolution etc.

**Module 5 - Performance Analysis and Optimization of Parallel Algorithms.** Advanced Topics in CUDA Programming.
Components of course

Lectures and Labs

Delivery Mechanism

Online Lectures - 2 lectures/discussion sessions per week supplemented by course materials in the form of video lectures/notes. Recordings of lectures will be provided to students. LMS- Moodle.

1 lab session of 2 hours to be conducted by course TAs.

Assessment method/Grading

Exam: One mid-semester examination and a final examination: 30% + 35%

Lab Assignments - 20%

Project - 15%

Grading scheme is relative and depends on both: class performance and minimum expectation from a student.