GPU Computing

Credits: 4

Pre-requisites:
Algorithms and data structures - CSE102, CSE222
Programming in C/C++ - CSE101, CSE201

Post Condition:
This course will expose students to design and write high performance code on the Graphics Processing Unit (GPU). A student who successfully completes this course will

- understand concepts behind parallel computing,
- understand parallel computing paradigms, GPU architecture and GPGPU development frameworks (CUDA, OpenCL, and GLSL),
- be able to develop parallel solutions to computationally challenging problems,
- be able to implement such solutions on GPU using CUDA, and
- be able to analyse the effectiveness of the GPU based solutions using standard benchmarks and tools.

Description:
This course will introduce parallel computing paradigms with focus on GPGPU programming to harness the massively parallel GPU architecture in solving computationally demanding tasks. The NVIDIA CUDA and industry standard OpenCL frameworks will be introduced and used with most of the labs. This is a project based course where the students will work on scientific computational problems.

Week	**Topics covered**
1	**Introduction and overview**: advances in architecture and technology, need for parallel computing, examples, and challenges.
2-3	**Basics on architecture and programming:** CPU/GPU architecture, multicore architecture, Flynn's taxonomy, SIMT execution model
4-5	**Introduction to CUDA C**: kernel based data parallel execution model, memory model and locality, CUDA threads, atomics, GPU utilisation
6-7	**Parallel programming paradigms**: parallel algorithm design, analytical modelling of parallel programs, limits on achievable performance, Amdahl's law, Gustafson's law, scalability, work optimality, message passing, shared address space machines, basic communication operations, concurrency

8	Project discussions and critical analyses
9-10	**Parallel computing using CUDA**: data transfer and CUDA streams, performance considerations, floating-point accuracy, synchronisation, communication, reduction trees, parallel prefix sum, optimisations
11-12	Case studies, Multi-GPU systems, GPGPU-computing using OpenCL and OpenGL
13	Project evaluations

Evaluation

This is a project based course and there will be regular project evaluations throughout the course. The students are expected to attend lectures and complete programming assignments. The course is programming intensive. The provisional breakdown for various elements is:

Assignments	20 %
Quizzes	5 %
Project	30 %
Mid-sem	15 %
End-sem	30 %

Texts/Other Resources

1. David B. Kirk, and Wen-mei W. Hwu, Programming massively parallel processors: a hands-on approach, Elsevier.
2. A. Grama, A. Gupta, G. Karypis, and V. Kumar, Introduction to parallel computing, 2nd edition.