Prof. William Spataro
Parallel Algorithms and Distributed Systems
Number of ECTS credits: 6 (of which 2 of laboratory activity)

- Objectives of the course

The course aims to provide students of the “Triennale” (Bachelor) Degree in Computer Science:

- Knowledge of the fundamentals and practical aspects of parallel computing;
- Knowledge of techniques and methods for the design and implementation of parallel algorithms;
- Thorough knowledge of the principles, structures and use of parallel processing systems and, in particular, of techniques for programming parallel computers using the shared-memory and message passing paradigms;
- Acquaintance of different fields of application of parallel computing and, in particular, of HPC (High Performance Computing).

Course Contents

Introduction to Parallel Computing

Aims, Concepts and Terminology
Flynn's taxonomy

Parallel Architectures

Overview of parallel machines
Shared Memory machines (Multiprocessors)
Distributed Memory machines (Multicomputers)
Multi-core architectures
Cache Coherence
Hybrid Architectures

Parallel Programming Models

Thread Model
Shared Memory Model
Distributed Memory Model
Parallel Data Models
Other Models

Design of Parallel Programs

Automatic and manual parallelization
Partitioning
Communication
Synchronization

Load Balancing
Granularity
Parallel I / O

Overview of OpenMP - The reference language for programming environments in shared memory / data parallel models
General Concepts
Parallel loop
Private and shared type variables
Critical Sections
Functional parallelism

MPI - The reference language for programming distributed memory environments
General Concepts
Environment Management Routines
Point-to-point communications: MPI_Recv and MPI_Send
Non-blocking communications
Collective Communications
Derived data-types
Communicators and Groups
Virtual topologies

GPGPU programming - The new frontier of Parallel Computing
CUDA C - The reference language for programming of graphics cards
General concepts, limits
CUDA, CUDA C
Thread and Memory Hierarchy
Concepts of threads, blocks and grid kernels
Optimization strategy: use of shared memory, coalescing

Performance Analysis
Parallel Overhead
Speedup and Efficiency
Superlinear speedup
Effect of Granularity on Performance
Scalability
Amdahl's Law
Iso-efficiency

Parallel Algorithms
Matrix calculation
Numerical integration

Searching and sorting algorithms
Performance analysis of parallel algorithms

Laboratory
Parallel software development in OpenMP and MPI on parallel computers
Development of GPGPU parallel software components in CUDA-C on NVIDIA graphics cards
Performance measurements

Recommended reading, books
- Ananth Grama et al., Introduction to Parallel Computing, 2/E, Addison-Wesley.
- Peter Pacheco, Parallel Programming with MPI. Morgan Kaufmann, 1997
- G. Spezzano, D. Talia "Calcolo parallelo, automi cellulari e modelli per sistemi complessi", Franco Angeli, Milano, 1999.
- I. Foster. Designing and Building Parallel Programs. Addison-Wesley, 1995, online version http://www-unix.mcs.anl.gov/dbpp
- David B. Kirk, Wen-mei W. Hwu, Programming Massively Parallel Processors, Second Edition: A Hands-on Approach 2nd Edition

NVIDIA CUDA C Programming Guide.
http://developer.download.nvidia.com/compute/cuda

NVIDIA CUDA C Best Practices Guide.
http://developer.download.nvidia.com/compute/cuda

- Online material (also available from the teacher's personal website)

Assessment methods
The final exam consists in a written test regarding all studied topics and development of a MPI/OpenMP/CUDA project chosen among proposed .