COMPUTER ENGINEERING (LM55)

(Lecce - Università degli Studi)

Teaching HIGH PERFORMANCE COMPUTING

GenCod A003136

Owner professor Giovanni ALOISIO

Teaching in italian HIGH PERFORMANCE COMPUTING

Teaching HIGH PERFORMANCE COMPUTING

SSD code ING-INF/05

Reference course COMPUTER ENGINEERING

Course type Laurea Magistrale

Credits 9.0

Teaching hours Front activity hours: 81.0

For enrolled in 2017/2018

Taught in 2018/2019

Course year 2

Language ENGLISH

Curriculum PERCORSO COMUNE

Location Lecce

Semester Second Semester

Exam type Oral

Assessment Final grade

Course timetable https://easyroom.unisalento.it/Orario

BRIEF COURSE DESCRIPTION

The architectures of cache-based microprocessors is discussed, with a special focus on their inherent performance limitations. Developments based on multicore chips and simultaneous multithreading are also considered. General optimization strategies for serial code on cache-based architectures are presented. Simple models are used to convey the concept of "best possible" performance of loop kernels, showing how to raise those limits by code transformations. Hands-on on parallel computing will be also organized through case studies to be developed using MPI and their parallel efficiency evaluation.

REQUIREMENTS

Good knowledge of the contents of first level courses on Informatics (Fondamenti di Informatica and Calcolatori Elettronici) and of the courses on "Parallel Algorithms" - 2nd Year (1st semester) of the 2nd Level Degree in Computer Engineering.

COURSE AIMS

After the course the student should be able to understand how to solve a number of central issues in high performance computing, starting from a deep knowledge of the basics of modern processor architectures and serial optimization techniques that can effectively exploit the architectural features for scientific computing.

TEACHING METHODOLOGY

Compulsory attendance is requested, since attendance at lectures and laboratory is mandatory because the course is based on the "learning by doing" approach on the advanced computing resources provided. The course will be held in the HPC Lab of the Engineering Faculty.

ASSESSMENT TYPE

The oral exam is aimed at verifying to what extent the student has gained knowledge and understanding of the selected topics of the course and is able to communicate about his understanding.

Students, divided into small groups (max two students), will also get hands-on experience, developing small projects on specific topics of the course. The max final vote is expressed as 30/30 with the possibility to get the laude

FULL SYLLABUS

Introduction of the course and review of previous topics covered in the first level course on Computer Architectures (2 hours).

General-purpose cache-based microprocessor architecture: Performance metrics and benchmarks. Moore's Law. Pipelining. Superscalarity. SIMD (6 hours).

Memory hierarchies: Cache. Cache mapping. Prefetch (6hours).

Advanced Solutions: Multicore processors. Multithreaded processors. Vector processors (6 hours).

Basic optimization techniques for serial code: Scalar profiling (Function and line-based runtime profiling, Hardware performance counters, Manual instrumentation). Common sense optimizations. Simple measure, large impact. Aliasing, Computational accuracy, Register optimizations, Using compiler logs). C++ optimizations (Temporaries, Dynamic memory management, Loop kernels and iterators) (6 hours).

Data access optimization: Balance analysis and lightspeed estimates (Bandwidth-based performance modeling, The STREAM benchmarks). Storage order. Case study: The Jacobi algorithm (6 hours).

Parallel computers: Taxonomy of parallel computing paradigms. Shared-memory computers (Cache coherence, UMA, ccNUMA). Distributed-memory computers. Hierarchical (hybrid) systems. Networks (Basic performance characteristics of networks, Buses, Switched and fat-tree networks, Mesh networks, Hybrids) (6 hours).

Basics of parallelization: Parallelism (Data parallelism, Functional parallelism). Parallel scalability (Factors that limit parallel execution, Scalability metrics, Simple scalability laws, Parallel efficiency, Serial performance versus strong scalability, Refined performance models, Choosing the right scaling baseline, Case study: Can slower processors compute faster? Load imbalance) (8 hours).

REFERENCE TEXT BOOKS

Georg Hager and Gerhard Wellein, " Introduction to High Performance Computing for Scientists and Engineers ", CRC Press, © 2011 by Taylor and Francis Group, LLC - ISBN 978-1-4398-1192-4.