Asignaturas Doctorado Profesores Visitantes:

SEMINAR 1

Issues in Computer Architecture and Microarchitecture for Future Computing Machines (3 credits – 3 ECTS – 30 Hours)
Prof. Yale Patt (University of Texas at Austin, USA)

Objectives:
Understand the relevant issues at the computer architecture and microarchitecture levels that will drive the design of future processors.

Description:
This course will focus on the Principles of Computer Architecture and Microarchitecture and the Tradeoffs that must be made in order to produce a microprocessor that satisfies a particular design point. We will look at the issues critical to future microarchitectures, and mechanisms that could prove useful to dealing with those issues. We will examine the newest microprocessors and latest cancellations, and see discuss the implications of that. The approach is always to stress the insights, rather than to memorize a collection of facts. A gross organization of topics is as follows:
1. Comprehensive overview of Architecture, Microarchitecture
2. Microarchitecture for high performance
3. Alternative approaches to concurrency
4. Relevant case studied
5. The future: Mechanisms that need more attention before they are used on real machines.

Evaluation method:
Continuous evaluation based on the assignments done during the course

Bibliography:
Conference and journal papers that will support the topics of the course and the assignments to be done

SEMINAR 2

Performance Engineering of Distributed e-Business Systems (3 credits – 3 ECTS – 30 Hours)
Dr. Samuel Kounov (University of Cambridge, UK)
Objectives:
Performance and scalability issues in modern e-business systems are gaining in importance as we move from prototypes to real operational systems. The course will provide an introduction to the state-of-the-art in the areas of benchmarking and performance modeling.

Description:

Overview of popular benchmarks Industry-standard benchmarks Proprietary benchmarks Open-source benchmarks Applications of benchmarks


Evaluation method:
Avaluació continuada de treballs proposats durant el curs pels professors de l'assignatura.

Bibliography:
Performance Engineering of Distributed Component-Based Systems – Benchmarking, Modeling and Performance Prediction


SEMINAR 3

Profiling and Optimization (3 credits – 3 ECTS – 30 Hours)
Prof David Kaeli (Northeastern University (Boston, USA))

Objectives:
To study how to obtain and use program runtime characteristics using both hardware and software techniques.

Description:
This course will discuss how to capture program runtime characteristics using both hardware and software techniques. The course will also discuss how to utilize these characteristics in the areas of architectural modelling, program optimization, binary translation, software fault recovery, software testing, and runtime monitoring. The subjects that will be covered in the course include:

1. A historical perspective of profiling and instrumentation
2. Instrumentation
   2.1 Hardware techniques
   2.2 Software techniques
3. Applications
   3.1 Performance
   3.2 Recovery
   3.3 Testing
   3.4 Monitoring
   3.5 JITs / Binary translators
4. Optimization
   4.1 Static
   4.2 Dynamic
5. Selected topics
6. Software/hardware interface
7. Program debugging and testing - tools and strategies

Evaluation method:
Continuous evaluation based on the assignments done during the course.

Bibliography:
Conference and journal papers that will support the topics of the course and the assignments to be done

SEMINAR 4

Ad Hoc and Sensor Networks (3 credits – 3 ECTS – 30 Hours)
Prof. Ian Akyldiz (Georgia Institute of Technology, Atlanta, USA)
Classes: starting 25/june (possibly from 15:00 to 18:00)

Objectives:
The course will be centered in research topics related with Sensor Networks. Issues such as MAC protocols, power consumption, routing protocols and network architectures will be studied. The course also will introduce those challenge aspects to be solved in this research area.

Description:
1. Introduction and applications
2. Factors affecting the design of sensor networks
3. Application layer, queries and network management
4. Transport Layer Protocols
5. Routing Protocols
6. MAC Protocols
7. Error Control Techniques and optimal packet size
8. Cross Layer Protocols solutions
9. Localization (Target Detection) algorithms
10. Synchronization algorithms
11. Topology control and connectivity maintenance
12. Actor/Sensor network (challenges and solutions)
13. Wireless Multimedia sensor network
14. Underwater sensor network (challenges and solutions)
15. Underground sensor network (challenges and solutions)
16. Grand challenges

Evaluation method:
Continuous evaluation based on the assignments done during the course

Bibliography:
Conference and journal papers that will support the topics of the course and the assignments to be done

SEMINAR 5

VLIW architectures and compilers (3 credits – 3ECTS – 20 Hours)
Dr. Paolo Faraboschi, (HP Barcelona)
Objectives:
Students will learn VLIW (Very Long Instruction Word) architectures as an alternative to the superscalar architectures, since they are widely used in embedded processors. Students will learn compiling techniques in order to obtain the maximum performance to VLIW architectures.

Description:
- Syllabus:
  - Introduction to ILP and VLIW
    VLIW and superscalars
    The VLIW philosophy
    Brief history of VLIW
  - VLIW Architectures
    Fetching and decoding wide instructions
    Code layout and compression techniques
    Register files and clustering
      Branch architecture and VLIW
    Hardware supported speculation
    Predication
  - VLIW Compilers
    Engineering of an ILP compiler
      Region formation: traces, superblocks, hyperblocks
    Acyclic scheduling
    Cyclic scheduling, software pipelining
    Phase ordering issues
  - VLIW in embedded computing
    Introduction to embedded computing, SoC, IP and cores
    VLIW and DSPs
      ILP and power/energy tradeoffs
    Use of VLIW as embedded accelerators
  - Special Topics
    Customizable processors
    Binary compatibility and dynamic compilation
      Real examples of processors using VLIW techniques
  - Lab
    Introduction to the VEX compiler and simulator
      Using VEX to optimize applications and architectures

Evaluation method:
Continuous evaluation based on the assignments done during the course

Bibliography:
Conference and journal papers that will support the topics of the course and the assignments to be done

SEMINAR 6

Multicore Systems Programming and Optimization (2 credits – 2 ECTS – 20 Hours)
Prof. Dimitrios S. Nikolopoulos (Virginia Polytechnic Institute and State University, USA)

Objectives:
This course teaches programming methodologies and tools for multicore processors. The first part of the course covers conventional programming models, such as threads and OpenMP, as well as emerging programming methodologies, such as transactional memory and non-blocking concurrent data structures. The second part of the course covers optimization methodologies for multicore processors, including optimizations for effective cache sharing between cores, data layout and memory access locality optimizations, multigrain parallelization, and scheduling. The third part of the course involves case studies with conventional multicore processor designs (Intel CoreDuo and IBM Power5) and unconventional heterogeneous multicore designs (IBM Cell).

Description:

1 Introduction to Multicore Architectures. Processor architectures and memory hierarchies
2 Multicore system software stack
   2.1 Operating System Support
   2.2 Thread libraries
   2.3 Memory management and memory allocators
   2.4 Synchronization and atomic operations
3 Programming with threads
   3.1 Thread creation
   3.2 Scheduling
   3.3 Synchronization
   3.4 Memory management issues
   3.5 Examples
4 Programming with OpenMP
   4.1 OpenMP parallelization
   4.2 OpenMP loop and functional parallelization
   4.3 OpenMP scheduling
   4.4 Extensions in the context of multicore processors
5 Transactional Memory
   5.1 Programming with transactions
   5.2 Optimizing transactions
   5.3 Non-blocking data structures and synchronization
6 Case Study I: Intel Core Duo
7 Case Study II: IBM Power5
8 Case Study III: IBM Cell

**Evaluation method:**
Continuous evaluation based on the assignments done during the course

**Bibliography:**
Conference and journal papers that will support the topics of the course and the assignments to be done

*Fast Simulation of Computer Architectures (1 credits – 1 ECTS – 10 Hours)*
Prof. Tom Conte (University of Illinois, Urbana-Champaign, USA)

**Objectives:**
The course describes fast, efficient and sophisticated techniques for quantitative computer architecture evaluation. It covers topics such as how to collect traces, emulate instruction sets, simulate multiprocessors using execution-driven techniques, evaluate memory hierarchies, apply statistical sampling to simulation, and how to augment simulation with performance bound models. The course targets to practicing computer architect designers seeking timely solutions to tough evaluation problems, and to advanced graduate level students doing research in this area.

**Description:**
Introduction
Fast Instruction-Set Simulator for Execution Profiling
Instrumentation Tools
Stack-Based Single-Pass Cache Simulation
Non-Stack Single-Pass Simulation
Execution Driven Simulation of Shared Memory Multiprocessors
Sampling for Cache and Processor Simulation
Performance Bounds for Rapid Computer System Evaluation

**Evaluation method:**
Continuous evaluation based on the assignments done during the course
**Bibliography:**
Conference and journal papers that will support the topics of the course and the assignments to be done

**SEMINAR 7**

*Autonomic Computing: Concepts to Systems (1 credits – 1 ECTS – 10 Hours)*
Prof. Omer Rana  (Cardiff University, UK)

**Objectives:**
The course focuses on the knowledge required to take decisions in the current Autonomic Computing systems. The Autonomic Computing vision delegates more autonomy to individual components, allowing such components to adapt and modify its behaviour based on changes in their operating environment. Topics: knowledge representation and annotation techniques - knowledge inferencing and expert systems - The Java expert system shell - different approaches - neural information processing techniques.

**Description:**
The course focuses on the knowledge required to take decisions in the current Autonomic Computing systems. The Autonomic Computing vision delegates more autonomy to individual components, allowing such components to adapt and modify its behaviour based on changes in their operating environment. Topics: knowledge representation and annotation techniques - knowledge inferencing and expert systems - The Java expert system shell - different approaches - neural information processing techniques.

**Evaluation method:**
Continuous evaluation based on the assignments done during the course

**Bibliography:**
Conference and journal papers that will support the topics of the course and the assignments to be done

*Autonomic Computing and Applications (2 credits – 2 ECTS – 20 Hours)*
Prof. Manish Parashar (Rutgers University, USA)

**Objectives:**
The course will start with an overview of Autonomic Computing with the motivations, origins, characteristics, challenges and opportunities. The course will address the key research issues in developing autonomic grid systems and
wll discuss potential paradigms and current approaches. Finally the course will review existing and emerging projects in industry and academia and detailed case studies of grid environments.

**Description:**
The course will start with an overview of Autonomic Computing with the motivations, origins, characteristics, challenges and opportunities. The course will address the key research issues in developing autonomic grid systems and will discuss potential paradigms and current approaches. Finally the course will review existing and emerging projects in industry and academia and detailed case studies of grid environments.

**Evaluation method:**
Continuous evaluation based on the assignments done during the course

**Bibliography:**
Conference and journal papers that will support the topics of the course and the assignments to be done

XML and Semantic Web Technologies and Applications (1 credits – 1 ECTS – 10 Hours)
Prof. Mark Baker (University of Reading, UK)

**Objectives:**
XML was developed by the W3C and aims to provide a universal format for describing structured documents and data; in other words, it allows data to be self-describing. XML is the basis of a number of emerging web-based technologies including Web Services and the Semantic Web. XML and these technologies are fundamental to the success of the new service oriented architectural approach to loosely coupled wide-area distributed systems

**Description:**
The aim of this Module is to:

i) Introduce XML (Extensible Markup Language) and related XML technologies and their application in representing and transforming of data and text.

ii) Introduce the technologies that support Web Services, including SOAP, Web Services Description Language (WSDL) and the Universal Description, Discovery and Integration (UDDI) protocol.

iii) Provide an outline of basic Semantic Web technologies, including the
Resource Description Framework (RDF), RDF Schema, and (OWL) the Web Ontology Language.

**Evaluation method:**
Continuous evaluation based on the assignments done during the course

**Bibliography:**
Conference and journal papers that will support the topics of the course and the assignments to be done

---

**SEMINAR 8**

HiPEAC Summer School  (3 credits – 3 ECTS – 30 Hours)

The "HiPEAC Summer School" is a **one week summer school for computer architects and compiler builders** working in the field of high performance computer architecture and compilation for embedded systems. The school aims at the **dissemination of advanced scientific knowledge and the promotion of international contacts** among scientists from academia and industry.

A distinguishing feature of this Summer School is its broad scope ranging from low level technological issues to advanced compilation techniques. In the design of modern computer systems one has to be knowledgeable about architecture as well as about the quality of the code, and how to improve it. This summer school offers the ideal mix of the two worlds – both at the entry level and at the most advanced level.

The HiPEAC Summer School is organized by the HiPEAC Network of Excellence.

The summer school is open to everybody but previous training and/or experience in computer science as well as a background in computer architecture or compilation is indispensable

[http://www.hipeac.net/summerschool/](http://www.hipeac.net/summerschool/)