Master CANS

Advanced Subjects

Processor Architecture (34320)

Objectives

This course focuses on the study of processor microarchitecture. It analyzes different microarchitectural schemes to improve performance, diminish complexity, reduce energy consumption and increase reliability of microprocessors. The different contemporary families of microarchitectures are studied: superscalar, multithreaded, multicore and vector processors, with a special emphasis on future trends. The course also covers the design of virtual machines. These are program execution environments implemented through a combination of hardware and software, which have emerged as a powerful approach to attack various important challenges in computers, such as software security and mobility, processor performance and energy consumption.

Contents

- Processor evolution and future trends
- Superscalar processors
- Design alternatives for superscalar processors
- Clustered microarchitectures
- Vector processors
- Multithreaded processors
- Multicore processors
- Reducing energy consumption
- Reliability
- Virtual machines

Methodology

Lectures: Explanation and discussion of the main concepts and analysis of alternative implementations.

Case studies: Study of concrete designs, usually taken from papers, with the objective of consolidating the main concepts discussed in the lectures.

Evaluation

Course assignment to be performed individually or in small groups.

Bibliography

- Computer Architecture: A Quantitative Approach, 4th. Edition, J. Hennessy and D. Patterson, Morgan Kaufmann

- Modern Processor Design, J. Shen and M. Lipasti, Mc Graw Hill

Multiprocessor Architecture (34321)

Compilers for high performance computers (34322)

Objectives

- Know the machine independent compiler optimizations and transformations.
- Know the machine dependant optimizations and transformations that increase the instruction level parallelism.
- Know the main techniques of dependence analysis, dependence removal and parallelization
- Know the differences between the main programming models for thread level parallelism.
- Know the main services of a thread library and its impact on the compiler transformations.
- Know the restructuring techniques to improve the performance of the memory subsystem.
- Rewrite code in order to help the compiler to better optimize code.

Contents

1. Introduction
2. Instruction Level Parallelism Optimizations
    - Instruction Level Parallelism
    - Machine Independent Optimizations
    - Instruction Scheduling
    - Register Allocation
3. Memory Hierarchy Optimizations (Marta Jiménez)
    - Basic Concepts
    - Basic transformations
    - Combination of transformations
4. Thread Level Parallelism Optimizations (Marc González)
    - Thread level Parallelism
    - Analysis and detection of parallelism
    - Programming models

- Parallel execution
- Memory models

Methodology

The course combines theory classes with interactive classes (exercises and discussions) and with the presentation of works from the students. The course has around 60 class hours, of which about 20% are devoted to the presentation of topics by the students. The rest comprises the theory and interactive classes.

The students devote the rest of the time (out of class) to individual and collective activities (study, solve exercises, research information and preparing works and presentations).

Grading

The course grading is done based on the students presentations, the corresponding student works and on the students participation to class.

Bibliography

Steven Muchnick. Advanced Compiler Design and Implementation. Morgan Kaufmann. ISBN-10: 1558603204 ISBN-13: 978-1558603202

Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman. Compilers: Principles, Techniques, and Tools, 2/E. Addison-Wesley. 2007. ISBN-10: 0321486811. ISBN-13: 9780321486813

Randy Allen, Ken Kennedy. Optimizing Compilers for Modern Architectures: A Dependence-based Approach. Morgan Kaufman, ISBN: 1-55860-286-0

Computer Design and Technology (34323)

Objectives

1. VLSI Design with MOS Technology
2. MOS circuits analysis and evaluation
3. Fabrication of VLSI devices
4. Power and thermal analysis of microprocessors
5. Power and temperature reduction techniques
6. Variability and Reliability aspects of future technologies

Contents

0. Historical Perspective
1. Introduction to MOS and VLSI Technology
2. Advanced MOS Design
3. VLSI Design
4. Thermal and Energy Analysis of Microprocessors
5. Design Implications of Temperature and Power
6. Variations and Reliability

Methodology

- Lectures

The professor presents the students the content of each topic and leads the discussion on different alternative solutions.

- Assignments / Problem sets

Students improve their knowledge through different kinds of assignments during the semester.

- Lab sessions

Students perform the specified lab sessions and submit their results for evaluation.

Evaluation

All assignments, problem sets and lab sessions are handed in for evaluation.

Resources

- Access to computers with the software to perform the lab sessions.
- Access to the tools used: simulators, editors, compilers, libraries, ...
- Description of the assignments to be completed.

Bibliography

CMOS VLSI Design: A Circuits and Systems Perspective (3rd Edition), Neil H.E. Weste, David Harris, Addison Wesley (2004)

Principles of CMOS VLSI Design: A Systems Perspective with Verilog/VHDL Manual (2nd Edition), Neil H. E. Weste, Kamran Eshraghian, Michael John Sebastian Smith, Addison Wesley (2000)

Jan M. Rabaey, Anantha Chandrakasan, Borivoje Nikolic; Digital Integrated Circuits: A Design Perspective; Editorial Prentice Hall, 2003

John P. Uyemura; CMOS Logic Circuit Design; Editorial KAP, 1999

Measurement Tools and Techniques (34324)

Objectives

Knowledge of the different techniques involved in performing research on computer architecture and technology: code instrumentation, obtaining execution traces, performance analysis and trace visualization, and performance simulation. Knowledge of the tools required to perform these tasks: instrumentation libraries and packages, trace visualizers, performance analyzers, and simulators. Knowledge about the different performance metrics relevant for a computer system and their actual meaning.

Contents

- Code instrumentation and trace generation

- Source code instrumentation

- Binary instrumentation

- Performance analysis and characterization, trace visualization

- Simulation

- Simulator taxonomy

- Simulation techniques

- Performance metrics

Methodology

• Lectures

The teacher present theoretical knowledge to the students, performs demonstrations of the different related tools, and moderates discussion on the alternatives.

• Practice

Students develop a practical work with assistance of the teacher.

• Student work

Students develop a practical work without direct supervision of the teacher.

Evaluation

Delivery of reports related to theoretical and / or practical aspects of the course.

Resources

- access to a personal computer for code development
- access to a multiprocessor system to execute parallel applications, perform measurements, and run simulations
- access to the related tools: simulators, instrumentation libraries and packages, visualizers
- description of the practical work
- bibliographical references (conference and journal papers, web sites, ...)

Parallel Programming Models and Algorithms (34325)

Objectives

During the course the student will: 1) develop, through a collaborative work group, a parallel code to solve a numerical problem that should run efficiently on a parallel architecture using a current parallel programming model, and 2) write an article/report outlining the results of the research work. This will involve a study of parallel programming models commonly used.

Contents

- Basic concepts
- Programming models for shared-memory architectures: OpenMP
- Programming models of distributed-memory architectures: MPI and PGAS
- Programming Models for Accelerator-based architectures

Methodology

Learning is based on solving a supercomputing problem (similar to those appearing in professional or academic applications). The solution of this problem guides all the learning process. The final result of the course is a parallel program that solves the problem in the most efficient and an article/report detailing the research findings. The teaching method adopted is the cooperative work within a group of 3 students, mostly using the technique called "puzzle". Group work creates a positive interdependence, individual demand and face-to-face interaction. In addition, interpersonal and teamwork skills are developed and auto-evaluation on the functioning of the group.

At the beginning of the course a detailed study guide for the course is made public, which includes: list of the specific objectives, detailed plan of activities to achieve these objectives (description of activities in class, outside class, individual, group, results of activities, deliverables, time commitment, delivery dates, etc..), evaluation plan (which activities are evaluated, how they are evaluated, whom and what is it for this evaluation).

The course plan 4 hours a week for 15 weeks. Of these 60 hours, only 15 are devoted to lectures, the rest are for the student to perform activities with members of its group or with the professor (discussions, questions, explanations to other members of the group, group evaluation, etc.). The students spend the remaining hours to complete the load of the course (90 hours) to perform activities outside the class, both individual (study, solving exercises, search for information, individual programming, etc..) and group (programming in pairs, and some activities similar to those in the classroom). It uses the virtual campus Athena, based on free software Moodle for the complete management of the course (planning, communication, documentation, forums, managing deliverables and rating).

Evaluation

- 60% for the evaluation of the final assignment and 40% for the continued evaluation of the training period.

Resources

- Atenea UPC virtual Campus
- Documentation about programming models: OpenMP, MPI, …
- Computing resources available at the Computer Architecture Department (DAC) at UPC and the Barcelona Supercomputing Center (BSC)
- Compilers and analysis Tools available in the target machines

Optimization of numerical simulations in engineering (34326)

NOT TO BE OPEN IN 2009-10

Optimization of non-numerical applications (34327)

NOT TO BE OPEN IN 2009-10

Operating Systems for New (Computer) Architectures (34328)

Objectives

The Operating Systems for New (Computer) Architectures (SONAR) course goal is to acquire a critical view of the issues and relations between the operating systems, runtime environments and the computer architecture. Operating systems evolution has always been behind the architecture We want to reverse this trend by analyzing the current use of the hardware features, the way abstractions are exported, how to characterize and manage correctly the system and applications needs. Performance and other metrics are considered at each design point. The student should be able to propose better uses of current architectures and to start influencing how new ones are designed.

Contents

Introduction to new generation operating systems

Abstractions and mechanisms for execution scheduling

Efficient management of the memory hierarchy

Power management

Operating systems and runtime support for parallelism

OS in the heterogeneous multi-core era

Case studies and evaluation

Methodology

Some topics are introduced at theory classes. Then next sessions are devoted to case and published paper discussions and student's presentations on topics they are interested.

Around half of the 150 hours that are expected to be devote to this course do not require class presence, they include: reading research papers, work on and prepare presentations and personal study. Each student takes responsibility of bringing an area of his/her interest to the course, and present the state of the art and the emerging trends in research and products.

Evaluation

The course grades are based on the participation at class discussions, presentations and work reports.

Resources

Conference papers, documentation on OS and runtime frameworks, specifications of computer architectures.

Execution environments for parallel architectures (34329)

Objectives

The course basic goal is to study and to design the services that the system will offer in order to achieve an efficient parallel execution. We will envision the application needs depending on the application execution patterns and depending on the architecture on top of which the application is running. The main discussion is centered on the operating environment and how this environment can affect the application performance. Other topics are: which services are more appropriate; in which way we have to offer the services; which are the best configuration parameter values for a given environment and how can we modify these parameters. Each course we will focus in some emergent technologies as SMP's and cluster computing, multicores or heterogeneous multiprocessors.

Contents

Distributed object technologies and protocols, Internal organization of middleware, Grid middleware, Middleware for eBusiness applications, Middleware Performance and Analysis, New Middleware Challenges: Autonomic Computing

Methodology

This course has an important experimental component. The main basic concepts previously known and we will assume they have been deeply studies in a previous basic course. With this assumption, we will start from the beginning of the course discussing about main existing research works, those works needed to have a good horizontal background concerning parallel architectures.

Approximately half of the classes will be guided by the professor presentations and the rest will be guided by the participation of students. Professors will propose research papers to make group discussions and will also propose some projects to be done individually or by groups. Both, paper discussion and shepherding of these works will be done during the practical classes.

Evaluation

The course evaluation will be done based on student participation in paper discussion and mainly in the realization of the projects. These projects will be specified during the first weeks of the course, realized (individually or in group depending on the project) during the course, and presented to the rest of the course during the last two weeks. All the parts of the project (specification, realization, and presentation) are components of the evaluation.

Resources

Conference and Journal Papers

Campus virtual

Manuals

Hardware resources from DAC and BSC-CNS

Open source Software and tools DAC or BSC-CNS provided.

Execution Environments for Distributed Computing (34330)

Objectives

Nowadays, business and scientific organisations have a large amount of critical workflow processes that depend upon a set of heterogeneous applications. This set of applications can range from transactional applications, with databases, to non interactive applications such as those that perform scientific CPU-intensive computing, document indexing or intensive I/O. To efficiently execute all these kinds of applications, a new execution environment (or middleware) is needed to manage the available resources and simplify the development and integration of the different types of applications and services. These middleware components are crucial to new systems and architectures because of their direct impact on the quality of the service offered by the application. This course provides an overview of the wide scope of this area and introduces past and current research focusing on conceptual and practical aspects of Middleware.

The course also has the objective of introducing the student to research. In general the EEDC course is focused on developing skills rather than content. For this reason each year we focus the course on some of the dichotomies which come to mind when looking at the strategies available to deal with concrete problems in the wider EEDC space and in relation to some problem of the utmost relevance. This year's special focus is on the green computing.

Content

MODULE 1- MIDDLEWARE BASICS

MODULE 2- TODAY SCENARIO: CLOUD COMPUTING

MODULE 3- TOPICAL FOCUS: GREEN COMPUTING

Methodology

Based on the previous concepts there will be paper readings assigned for some of the classes. This is an important part of the course. We will have two main reading parts in the course, one for general papers and one for more in depth research in the area (related with the research project). Discussions will be led by one or more students and may include a brief presentation of the paper.

All students taking the course are required to complete a research project (or "state-of-the-field" review). The project is intended to provide the student with an opportunity to gain experience with research in a topic related to the content of the course. Moreover we will introduce the student to the research information resources.

Finally we will learn how a research conference is organized. The students will be involved in the organization committees of the Third Workshop on Execution Environments for Distributed Computing (EEDC 2009). All the research projects will be presented at this conference after a standard review process. Finally all the projects will be published as a book chapter in a book.

On completion of this course you will be able to:

- Students will have a solid understanding of the concepts used in the course. This understanding will provide them with the foundation necessary to be able to pursue further learning on their own.
- Students will be an expert in this important area of computer science that support current internet world.
- Students will have an understanding of the impact of computers in society.
- Students will gain experience to orally communicate ideas and concepts clearly and in an organized manner.
- Students will gain experience to write research reports and books.
- This course will help students to be prepared to enter the scientific computation community in academia or in industry.

Evaluation

- Paper Readings/Presentations: 35%
- Research Project: 35%
- Participation: 30%

Resources

- UPC Virtual Campus
- Research papers
- Internet resources

Next Generation Internet (34331)

Objectives

This course presents the key aspects and concepts of the Internet technology which are already consolidated. These technical aspects are the background of many mechanisms being used in new generation networks in order to provide quality of service.

The course introduces the evolution of the networking technology using MPLS (Multiprotocol Label Switching). Research topics about MPLS network management and inter-domain issues are presented.

The next block of the course deals with the technologies for Quality of Service (QoS) provisioning. It starts with the basic principles about QoS and the available mechanisms, and follows with the description and evaluation of the architectures proposed by IETF. The open research challenges are presented.

The course to be continued with research topics related to Network Monitoring, Measurements and Traffic Analysis.

Finally, resilience routing and energy-aware network optimization are introduced pointing out its novel aspects and the problems it poses when deploying Internet.

Special stress is given on research topics and future trends.

Contents

- Concepts about Next Generation Internet.
- Multiprotocol Label Switching (MPLS).
- Quality of Service Principles. QoS in packet networks
- Switching and Routing paradigms.
- Mobility and "always-connected" Internet.
- Pervasive Internet "all-IP-networks".
- New network architectures for the future internet.
- Traffic monitoring and QoS assessment.
- Network Monitoring, Measurements and Traffic Analysis.
- Resilient networks
- Energy-aware network optimization

Methodology

Active participation of the students is envisaged. During the first part of the course lectures are taught introducing the main topics of the course with integration between expositive classes and collaborative work using a Virtual Campus. During the second part of the course students present several more specific topics, preferably related with open issues and research activities. Students prepare their presentation, individually or in groups of two, choosing the topic from a given list. These topics are selected so that they allow focusing in aspects that complement the subjects taught in the first part of the course. In this second part advisors follow closely the preparation of the student's presentations.

Evaluation and grading

The evaluation of the course includes not only the knowledge acquired by the students in the topics of the course but their level of participation in the sessions.

Grading includes the following aspects:

- Attendance and participation (10%)

- Preparation and resolution of workshops (reading research papers) (20%)

- Talk and presentation of 20-25 minutes (slides plus notes) related with some of the topics covered in the course. (30%)

- Final exam (40%)

Bibliography

"MPLS and Label Switching Networks". Uyless Black. Ed. Prentice-Hall. 1st. Edition, 2001. "Engineering Internet QoS". Sanjay Jha, Mahbub Hassan. Artech House 2002. ISBN 1-58053-341-8.

"Internet Performance Survival Guide. QoS Strategies for Multiservice Networks". Geoff Huston. John Wiley & Sons Inc. 2000. ISBN 0-471-37808-9.

"IPv6 Essentials. Integrating IPv6 into your IPv4 Network". Silvia Hagen. O'Reilly 2002. ISBN 0-596-00125-8.

"Distributed Multimedia through Broadband Communications Services". Daniel Minoli, Robert Keinath. Artech House. 1994. ISBN 0-89006-689-2.

"Differentiated Services for the Internet". Kalevi Kilkki. MacMillan Technical Publishing. 1999. ISBN 1-57870-132-5.

Complementary references: RFC (IETF) and a selection of journal papers.

Optical Internet (34332)

Objectives

This course is tailored for students to acquire knowledge on the optical networking panorama (short, medium and long term evolution of the optical transport network infrastructures), and its impact on the present and future Internet. The students will learn how to transport IP traffic over the current networks of optical fibre and over the near future WDM (Wavelength Division Multiplexing) networks, the future trends of the optical networks architectures based on the statistical multiplexing paradigm, and the main hints of the role of the optical technology in the access networks.

Outline

1. Transporting IP Traffic over Optical Networks

1.1 Networks of Optical Fibre:

- Current technologies: IP/ATM (short summary)
- In deployment technologies: IP/SDH
- Migration to WDM: IP/WDM
- IP over (1-10) GigabitEthernet/WDM

1.2 Optical Core Networks:

- IP/OTN (Optical Transport Networks)
- IP/ASON (Automatic Switched Optical Networks)
- GMPLS (Generalized MultiProtocol Label Switching
- Routing and Wavelengths assignment in WDM networks

2. Long Term Optical Network Architectures

2.1 Optical Burst/Packet Switching

- Fast packet switching architectures: evolution to optical burst/packet switching
- Optical packet switching (OPS) concepts and preliminary prototypes
- Optical MPLS (MPLS over OPS)
- QoS provisioning in OPS

2.2 Optical Burst Switching

- Traffic handling, contention resolution and bursts assembling
- The off-set time emulated architecture (E-OBS)
- Integration of GMPLS and OBS

2.3 Metropolitan Optical Networks (MONs)

- Optical Metro Networks Architectures
- QoS provisioning in MONs

3. Optical Access Networks

- XDSL (X-Digital Subscriber Loop)
- HFC (Hybrid Fibre Coaxial)
- PONs (Passive Optical Networks)

Methodology

The methodology of this course will follow the criteria established by the new *Superior European Education Space (SEES)*, which objective is that the students "*learn to learning*".

Evaluation

The evaluation and grading of the students registered to this course will be done according with the following items:

1. *Attendance to the lectures: 20% of the score.*
2. *Technical Report (elaboration and presentation): 40% of the score.*
3. *Instead of a final exam, students will be required to* elaborate *three sets (one set per chapter) of 2 or 3 questions that could be appropriate for an exam of the course (i.e., not extremely easy or too difficult questions, but fitting with the level reached in the course): 40% of the score.*

Resources

Basic Bibliography:

- *B. Mukherjee, "Optical WDM Networks", Ed.Springer, 2006*
- *Harry G. Perros, "Connection-Oriented Networks", Ed. Wiley, 2005*
- *Peter Tomsu, Christian Schmutzer, "Next Generation Optical Networks", Ed. Prentice Hall, 2002*
- *Uyless Black, "Optical Networks, Third Generation Transport Systems", Prentice Hall, 2002*
- *Jean Walrand, et altri. "High Performace Communication Networks", Ed. Morgan Kaufmann, 2000 (2nd. Edition)*

Notes and teaching material:

A copy of the slides, additional tutorial and white papers, and any other teaching material required in the course will be available through Internet

Wireless LANs, Ad-hoc and Sensor Networks (34333)

Objectives

The subject will be oriented to the analysis of local area wireless networks and wireless ad hoc networks. The course will focus on the problems derived from wireless transmission, description of the protocols, and analytical models that have been proposed for these type of networks in the literature.

Contents

1. Signal transmission in wireless channels
2. Medium access protocols
3. Routing algorithms

Methodology

There will be 4 hours per week, dedicated to magistral classes to explain the theory and solve problems. The students' activities will consist of reading papers, solving practical problems that will be proposed, and doing some presentation of case studies. The problems will be collected and corrected during the course.

Evaluation

The evaluation will be based on the correction of the problems proposed to the students during the course and the presentation of case studies.

Resources

Papers, problems and other material that will be provided during the course.

References

- Andrea Goldsmith, "Wireless Communications", Cambridge University Press, ISBN-13: 9780521837163 | ISBN-10: 0521837162
- Stephen Boyd, Lieven Vandenberghe, "Convex Optimization", Cambridge University Press, ISBN-13: 978-0521833783, ISBN-10 0521833787
- U. H. Gerlach, "Linear Mathematics in Infinite Dimensions", "http://www.math.ohio-state.edu/~gerlach/math/BVtypset/BVtypset.html

Mathematical Concepts Applied to Computer Networks (34334)

Objectives

Provide the student with general probability concepts focused on the mathematical analysis of computer networks.

Contents

1- Combinatory.
2- Mathematical model of probability
3- Inclussion-exclussion formula
4- Continous models
5- Conditioned probability and expectation
6- Probability distributions
7- Repeated trials
8- Autosimilar distributions
9- Information theory
10- Stochastic processes
11- Markov Chains
12- Queueuing Theory

Methodology

There will be 5 hours per week, dedicated to magistral classes to explain the theory and solve problems. The students' activities will consist of reading material and solving practical problems that will be proposed at each theoretical unit. The problems will be collected and corrected during the course.

Evaluation

The evaluation will be based on the correction of the problems proposed to the students during the course and a final exam.

References

• An Introduction to Probability Theory and Its Applications, William Feller, Wiley; (1971)

• Probability, Random Variables and Stochastic Processes, Athanasios Papoulis McGraw-Hill (2001)

• Understanding Probability: Chance Rules in Everyday Life, Henk Tijms, Cambridge University Press (2004)

• Cálculo de probabilidades 1. Ricado Vélez, Víctor Hernández, UNED 1995.

• Finite Markov Chains. John G Kemeny, J Laurie Snell. Springer, 1960

• Probability, Stochastic Processes, and Queuing Theory. Randolph Nelson. Springer 1995.

Decentralized Systems (34335)

Objectives

Learn the fundamental concepts, operation, design and evaluation of decentralized systems.

Contents

Fundamental concepts: peer-to-peer and overlay networks.

Routing: unstructured and structured overlay networks.

Techniques and models: publish/subscribe, group communication, self-properties, incentives, management, resource allocation, security and anonymity, characterization and evaluation.

Applications: content and media distribution, storage, file sharing, communication, computing, social networks.

Methodology

Theory classes, readings of research papers, presentation of topics by students.

Evaluation

The evaluation of the course will be based on the participation of students in class, reading reports and a project work on specific topics.

References

George Coulouris, Jean Dollimore, Tim Kindberg. Distributed Systems: Concepts and Designs. Addison Wesley, 3 edition. 2001.

Ralf Steinmetz, Klaus Wehrle, Peer-to-Peer Systems and Applications, Springer, 2005. LNCS 3485.

Tanenbaum, A., Steen, M. (2007). Distributed Systems: Principles and Paradigms, 2 / E. New Jersey: Prentice Hall.

Birman, K. (2005). Reliable Distributed Systems. Technologies, Web Services, and Applications. New York: Springer Verlag.

A collection of research papers.

Computer Security and Secure Distributed Applications (34336)

Communication Networks Optimization (NEW)

Objectives

The course is a comprehensive introduction to the theory and algorithms of integer optimization. It is organized in four parts: mathematical programming, heuristic algorithms, network flows, and communication network problems.

The goals of the course are the following:

- To present students with a knowledge of the state-of-the art in the theory and practice of communication network applied integer optimization,
- To provide students with a rigorous analysis of heuristic algorithms,
- To help each student develop his or her own intuition about ILP modeling and algorithm development and analysis.

Contents

1. Mathematical Programming:

This introductive part presents the mathematical programming basics and discusses how to formulate integer optimization problems, the duality of integer optimization and how to solve the resulting relaxations both practically and theoretically.

Contents:

o Linear Programming (LP) basics

o The SIMPLEX algorithm

o LP basics models

o Duality

o Integer programming (ILP)

o The branch-and-bound algorithm

o Relaxations

o Non-linear models

o Network flows models

o Multi-commodity models

o Modeling and solving software: AMPL/CPLEX

2. Heuristics methods:

The last two decades have witnessed a tremendous growth in the area of heuristic algorithms. Two benefits of heuristics have spearheaded this growth: simplicity and speed. This part presents the basic concepts of heuristic algorithms and some advanced meta-heuristics. Finally, a discussion about the application of heuristics to solve integer optimization problems is presented.

Contents:

o Greedy procedures

o Improvement methods: exchanges and local search.

o Meta-heuristics: GRASP, Simulated Annealing, Tabu Search, Genetic

algorithms, Ant colony, Scatter Search, Path Relinking.

o Mathematical programming vs. heuristics

3. Network flows:

Network flow problems form a subclass of linear programming problems with applications to transportation, logistics, manufacturing, computer science, project management, finance as well as a number of other domains. This part surveys some of the applications of network flows and focus on key special cases of network flow problems.

Contents:

o Max flow-min cut

o Shortest path problems

o Spanning-Trees

o Max flow vs. k-shortest path

o Node-Arc and Arc-path formulations

4. Communication Networks applications

This part concludes the course centering the previously learned concepts into communication networks problems. Research level problems are presented and ILP modeling and algorithms to solve it are analyzed.

Contents:

o Network planning

o Traffic engineering and quality of service: Load balancing, Maximum Delay minimization o Protection and restoration

o Re-optimization

o Examples in IP/MPLS, optical (OCS, OBS, OPS), and wireless networks

Methodology and grading

Collaboration is encouraged on all aspects of the class.

Groups of two may collaborate and hand in a single applied project. Each project consists on define a problem drawn from your own research area, design an ILP model, and design and implement some heuristic algorithms to solve the problem. Students should compare results obtained with both methods.

The grading for the subject will be determined using the following weights, part assignments: 30%, applied project: 70%.

References

1. Williams, H. P. Model building in mathematical programming. John Wiley & Sons, 1993.

2. Winston, Wayne L. Operations research: applications and algorithms. PWS-KENT, 2004.

3. Luenberger, D.G. Linear and nonlinear programming. Kluwer Academic Publishers, 2004.

4. Hillier F.S., Lieberman G.J. Introduction to operations research. McGraw-Hill, 2005.

5. Glover, F.; Kochenberger, G.A. Handbook of metaheuristics. Kluwer Academic Publishers, 2003.

6. Michalewicz, Z.; Fogel, D.B. How to solve it modern heuristics. Springer, 1999.

7. Glover, F; Laguna, M. Tabu search. Kluwer Academic Publishers, 1997.

8. Ahuja, Magnanti, and Orlin. Network Flows: Theory, Algorithms, and Applications. Prentice Hall, 1993.

9. Medhi, D; Ramasamy, K, Network Routing: Algorithms, Protocols, and Architectures. Elsevier Morgan Kaufmann, 2007

10. Grover, W.D., Mesh-based survivable networks, Prentice Hall, 2004.

Scalable Distributed Systems (NEW)

Objectives

Learn the fundamental concepts, operation and design guidelines, patterns and methods for developing software systems with good scalability and predictability, and the evaluation of scalable and Internet-scale systems.

Content

- Fundamental concepts: The effect of scale in system properties (functional and operational).

- Issues in large-scale systems: virtualization, service orientation and composition, availability, locality, performance and adaptation (host-spots, autonomic computing).

- Models for large-scale systems: system models for analysis (game-theoretic, economic, evolutionary, control, complex networks), architectural models (multi-tier, cluster, farm, grid, cloud, SaaS).

- Scaling techniques: basic techniques (caching, distribution, replication), scalable computing techniques for architectural models.

- Middleware and Applications: computing, storage, web, content distribution, Internet-scale systems or services

Methodology

Theory classes,

Reading and discussion of research papers,

Presentation of topics by students,

Laboratory activities

Evaluation

The evaluation of the course will be based on the participation of students in class, reading reports and an experimental project work on specific topics.

References

A collection of research papers.