Scuola Superiore Sant'Anna
di Studi Universitari e di Perfezionamento

Università di Pisa

Master Program (Laurea Magistrale) in Computer Science and Networking

Academic Year 2009-2010

High Performance Computing Systems and Enabling Platforms

Marco Vanneschi

Department of Computer Science, University of Pisa

Scuola Superiore Sant'Anna
di Studi Universitari e di Perfezionamento

UNIVERSITÀ DI PISA

Master Program (Laurea Magistrale) in Computer Science and Networking

High Performance Computing Systems and Enabling Platforms

Marco Vanneschi

Course Introduction

My activity

- Research area
  - Computer Architecture
  - Parallel and Distributed Processing, High Performance Computing
  - Parallel Programming Models and Tools
  - Programmability of various HPC platforms
    - Multiprocessor, Cluster, Grid Computing, Multi-core, Pervasive Computing
  - Coordination of some National and European Projects (basic research and industrial research)
- Research group
  - Co-leaded with Prof. Marco Danelutto
  - Laboratory of Parallel Architectures
- Strong relationship research - teaching

This course (acronym: SPA)

My Personal Page: www.di.unipi.it/~vannesch section: Teaching

Link in DidaWiki to my Personal Page:
http://www.cli.di.unipi.it/doku/doku.php/magistraleinformaticanetworking/spa/start

- Fundamental course of *Laurea Magistrale in Computer Science and Networking*, 1st Year

- In common with *Laurea Magistrale in Informatica*
  - complementary course (study plan in Distributed Systems)

- *ASE* (vecchio ordinamento, 9 CFU): 6 CFU di SPA + 3 CFU di integrazione (vedi in seguito)

SPA

6 CFU = 48 hours

(12 weeks)

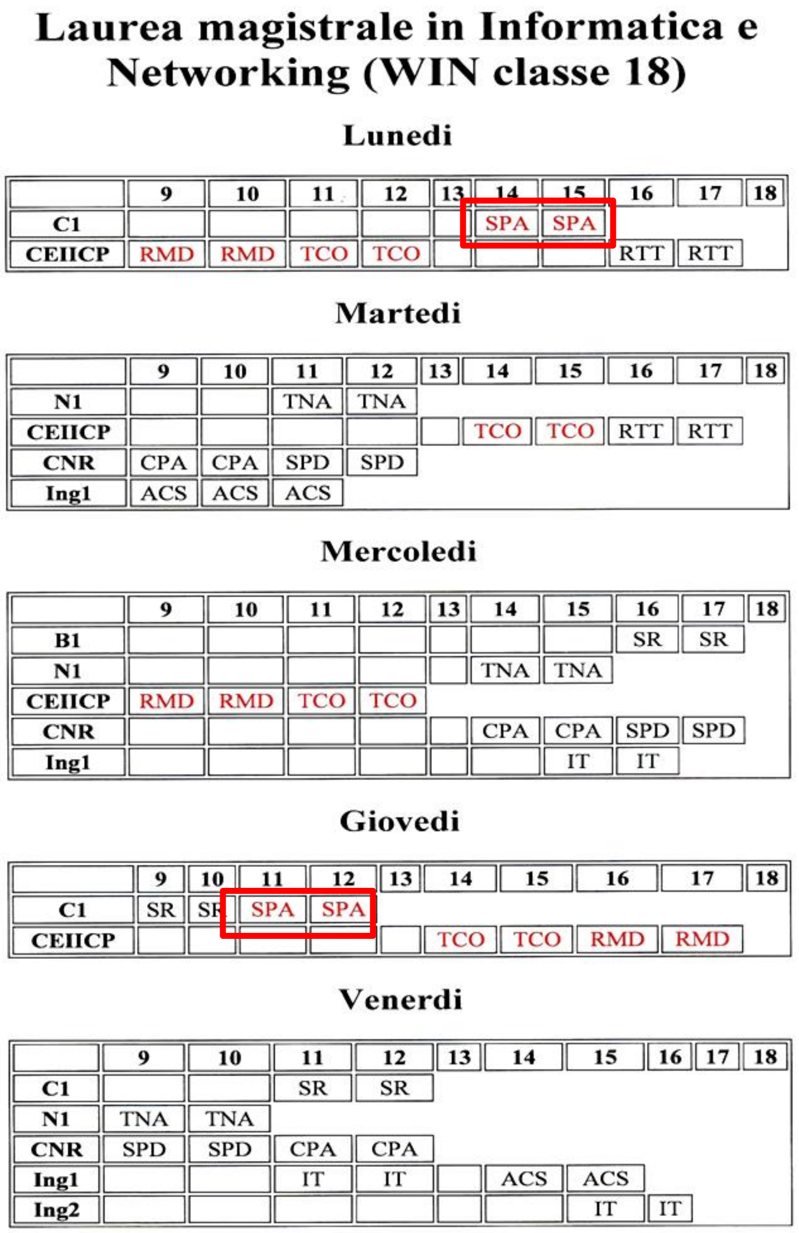
CREDIT DEFINITION:

1 CFU = 25 hours =

8 hours for lectures / class activities (lab, practical, etc)

+

17 hours for individual study



Laurea magistrale in Informatica e Networking (WIN classe 18)

Lunedi

	9	10	11	12	13	14	15	16	17	18
C1						SPA	SPA			
CEIICP	RMD	RMD	TCO	TCO				RTT	RTT	

Martedi

	9	10	11	12	13	14	15	16	17	18
N1			TNA	TNA						
CEIICP						TCO	TCO	RTT	RTT	
CNR	CPA	CPA	SPD	SPD						
Ing1	ACS	ACS	ACS							

Mercoledi

	9	10	11	12	13	14	15	16	17	18
B1								SR	SR	
N1						TNA	TNA			
CEIICP	RMD	RMD	TCO	TCO						
CNR						CPA	CPA	SPD	SPD	
Ing1							IT	IT		

Giovedi

	9	10	11	12	13	14	15	16	17	18
C1	SR	SR	SPA	SPA						
CEIICP						TCO	TCO	RMD	RMD	

Venerdi

	9	10	11	12	13	14	15	16	17	18
C1			SR	SR						
N1	TNA	TNA								
CNR	SPD	SPD	CPA	CPA						
Ing1			IT	IT		ACS	ACS			
Ing2							IT	IT		

Contents

1. Objectives, motivations, approach
2. An informal presentation of some concepts and technologies
3. Background and prerequisites
4. Course program
5. Course material/notes
6. Exam modality and working approach

Contents

1. Objectives, motivations, approach
2. An informal presentation of some concepts and technologies
3. Background and prerequisites
4. Course program
5. Course material/notes
6. Exam modality and working approach

Course objectives

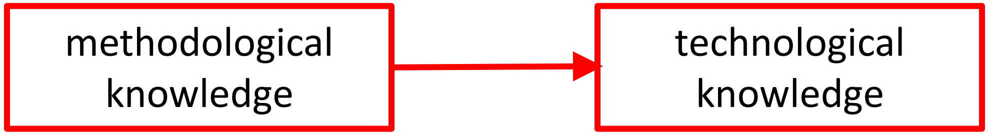
- Provide a solid knowledge framework of concepts and techniques in high-performance computer architecture
  - Organization and structure of enabling platforms based on parallel architectures
  - Support to parallel programming models and software development tools
  - Performance evaluation (cost models)
- Methodology for studying existing and future systems
- Technology: state-of-the-art and trends
  - Parallel processors
  - Multiprocessors
  - Multicore / manycore / … / GPU
  - Shared *vs* distributed memory architectures
  - Programming models and their support
  - General-purpose *vs* dedicated platforms

Motivations

- Basically, the same motivation discussed in “Distributed Systems: Paradigms and Models” (Prof. Marco Danelutto):
  - evolution of computer technology towards parallelism and HPC
    - Multi/many core
    - Large cluster, cloud, …
    - Heterogeneous large-scale enabling ICT platforms
    - Embedding
- Increasing maturity with respect to “hardware-software relationship”
  - Both *Technology Push* and *Technology Pull*
- Language-driven architectural approaches
- Concurrency and parallelism as first-class citizens in application development
- In our Master: HPC is a fundamental methodology and technology for integrated ICT infrastructures and applications

Course approach: a Computer Science approach

- The concept of **Enabling Platform**: strong relationship and integration between architectures and applications
  - Computing architectures are NOT “boxes and wires”



This approach does not imply that *concrete* technologies (of current and future) computing platforms are neglected – *far from it*: our goal is to fully understand and to utilize existing architectures at best, and even to define new ones.

- Computer Science approach
  - ○ Computing architecture has its own concepts, principles, models, and techniques
  - ○ Conceptual framework in common with the other disciplines of Computer Science:
    - Programming languages, algorithms, computability and complexity, …

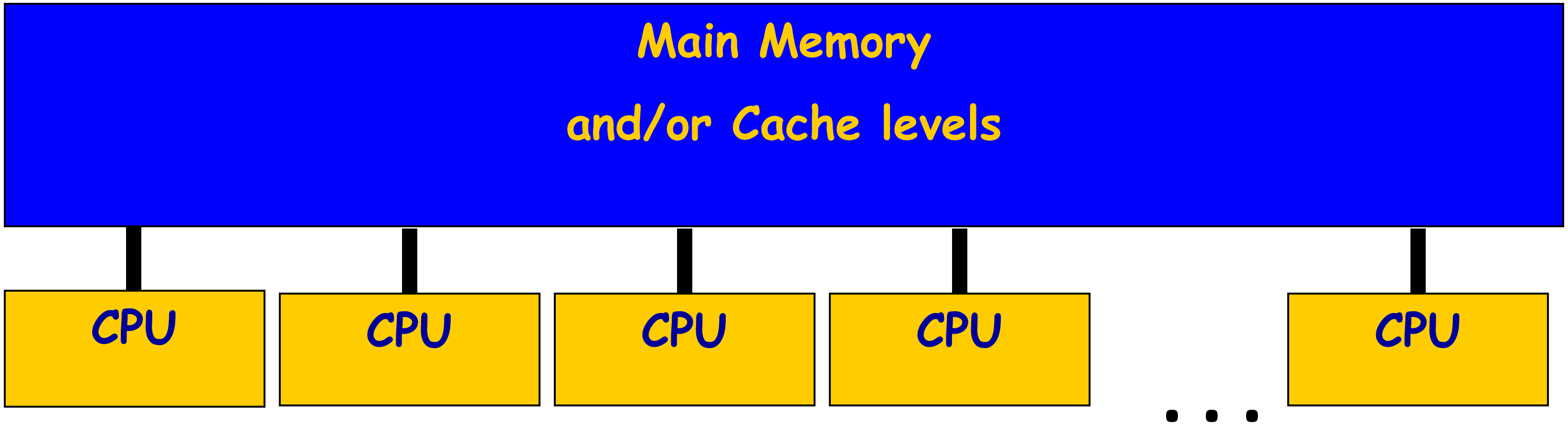
Contents

1. Objectives, motivations, approach
2. An informal presentation of some concepts and technologies
3. Background and prerequisites
4. Course program
5. Course material/notes
6. Exam modality and working approach

HPC enabling platforms

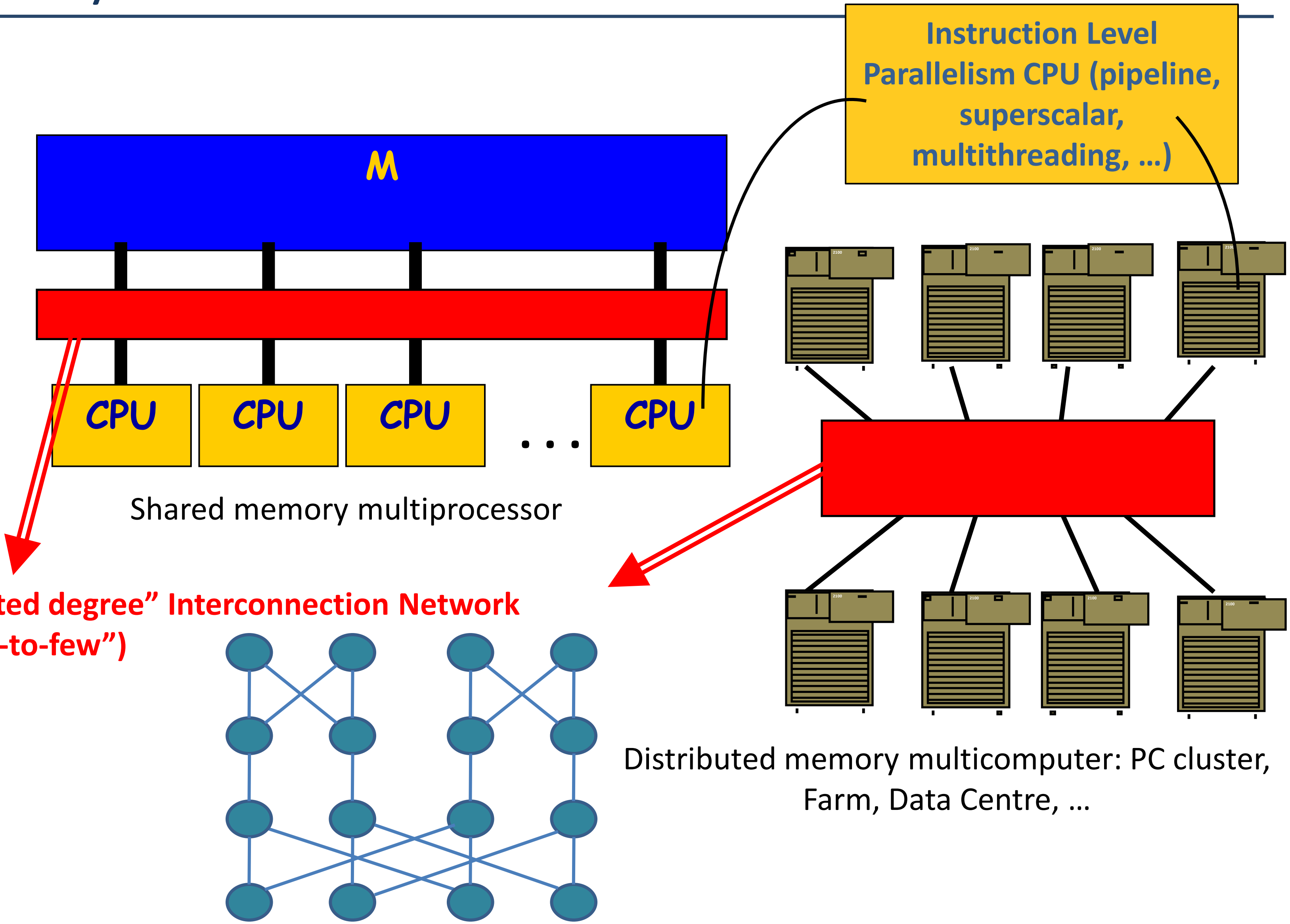
Shared memory multiprocessors

- Various types (SMP, NUMA, …)



- From simple to very sophisticated memory organizations
- Impact on the programming model and/or process/threads run-time support

HPC enabling platforms: shared and distributed memory architectures



Instruction Level Parallelism CPU (pipeline, superscalar, multithreading, …)

Shared memory multiprocessor

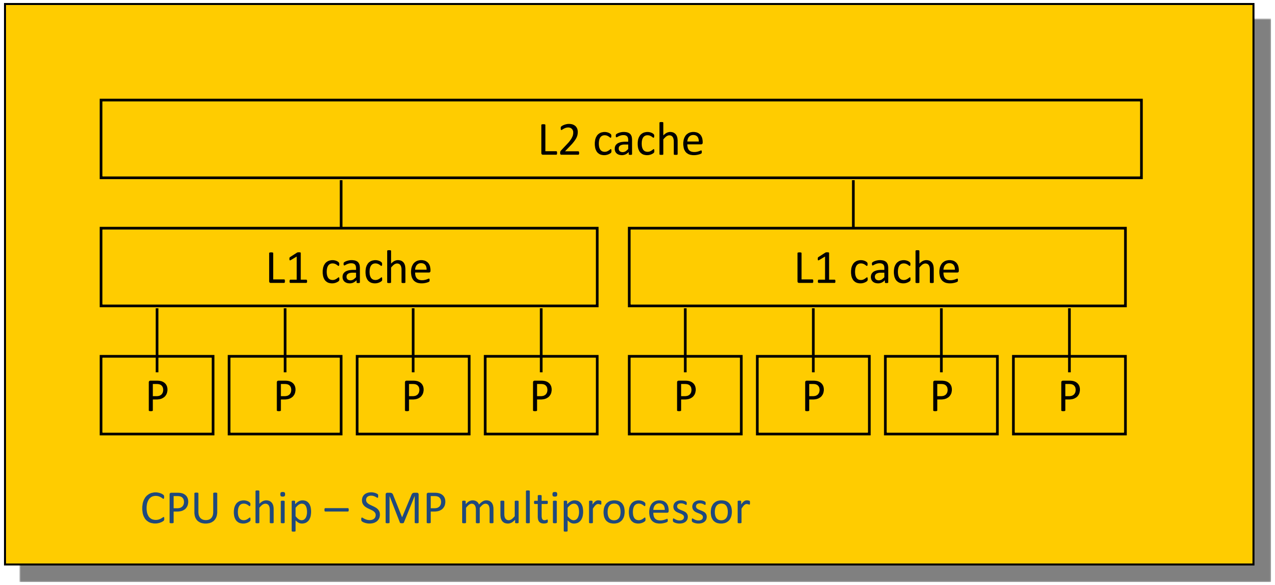
“Limited degree” Interconnection Network (“one-to-few”)

Distributed memory multicomputer: PC cluster, Farm, Data Centre, …

CPU technology evolution and multicore

Multiprocessor on single chip

- “Dramatic revolution” for ITC industry: programmability issues
- Computer providers support : from sequential programming to parallel programming
- Also: NETWORK PROCESSORS with multicore technology

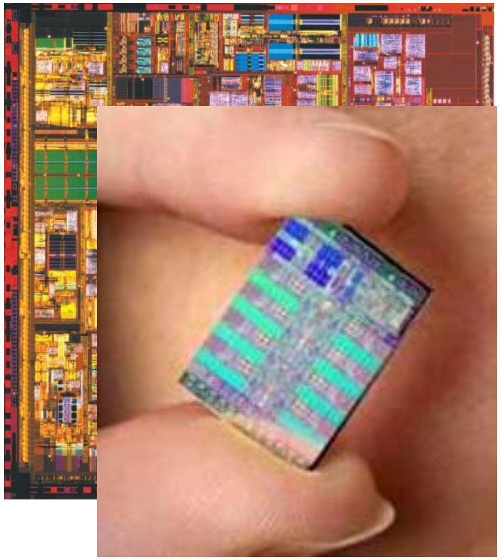

L2 cache

L1 cache

L1 cache

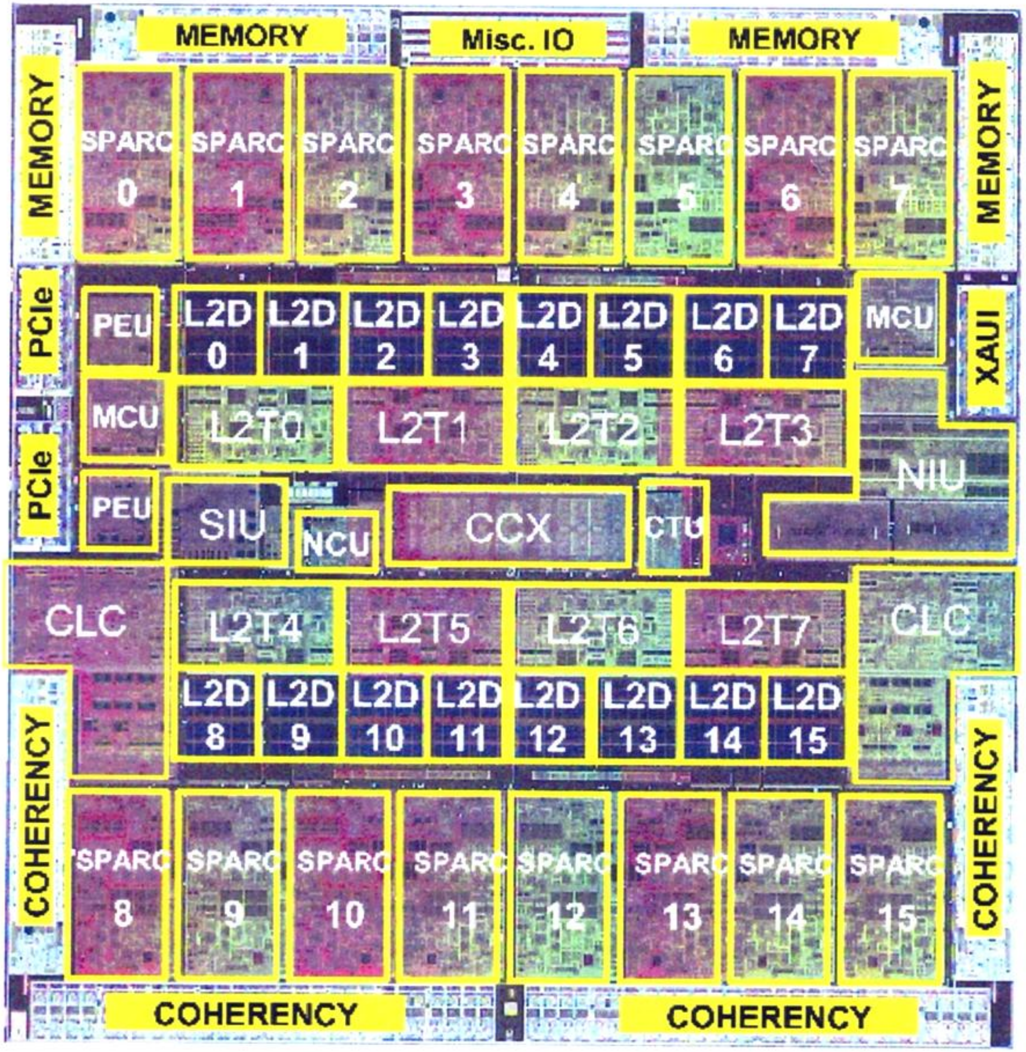
P P P P P P P P

CPU chip – SMP multiprocessor



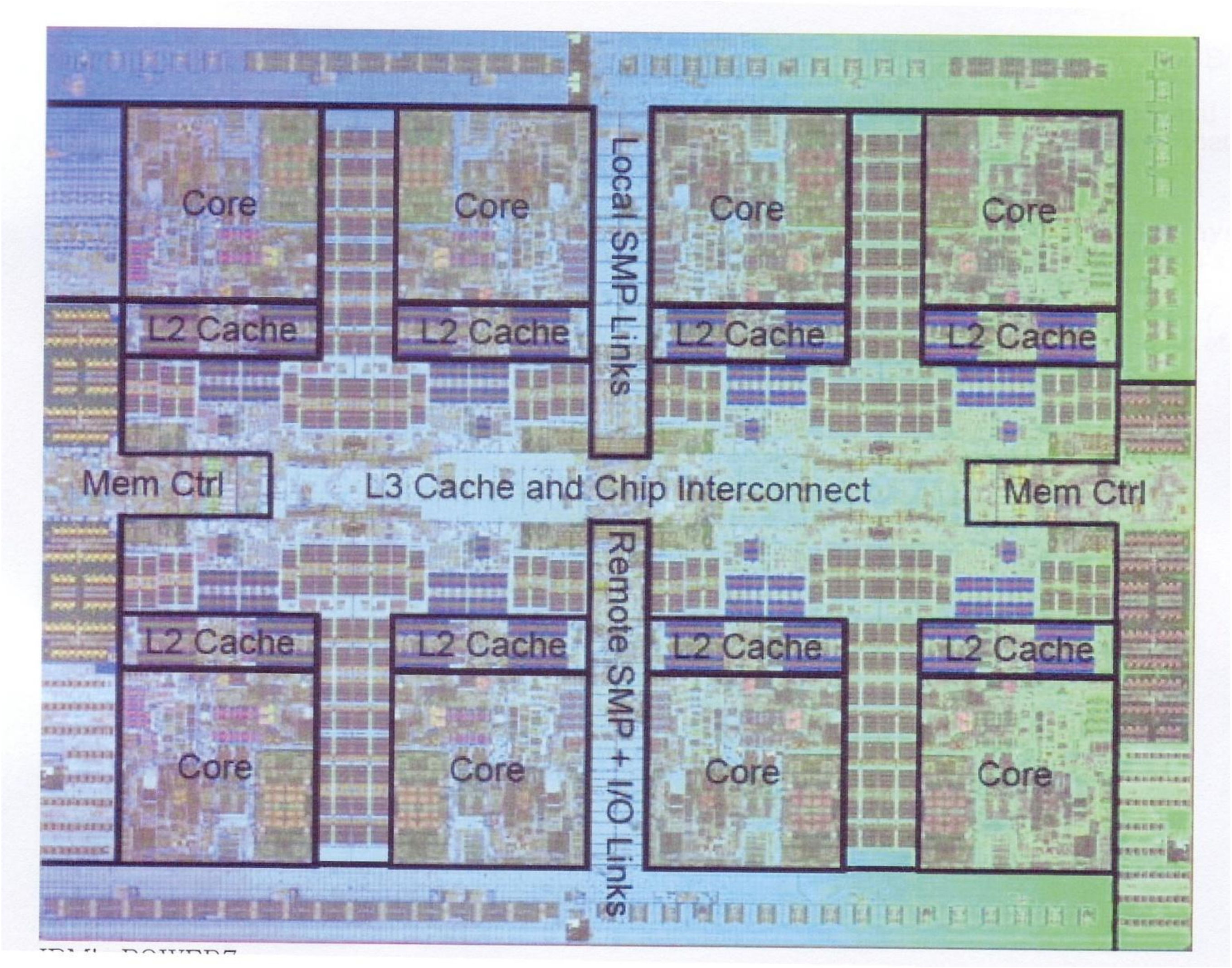
Multicore technology examples

SUN
Niagara 3



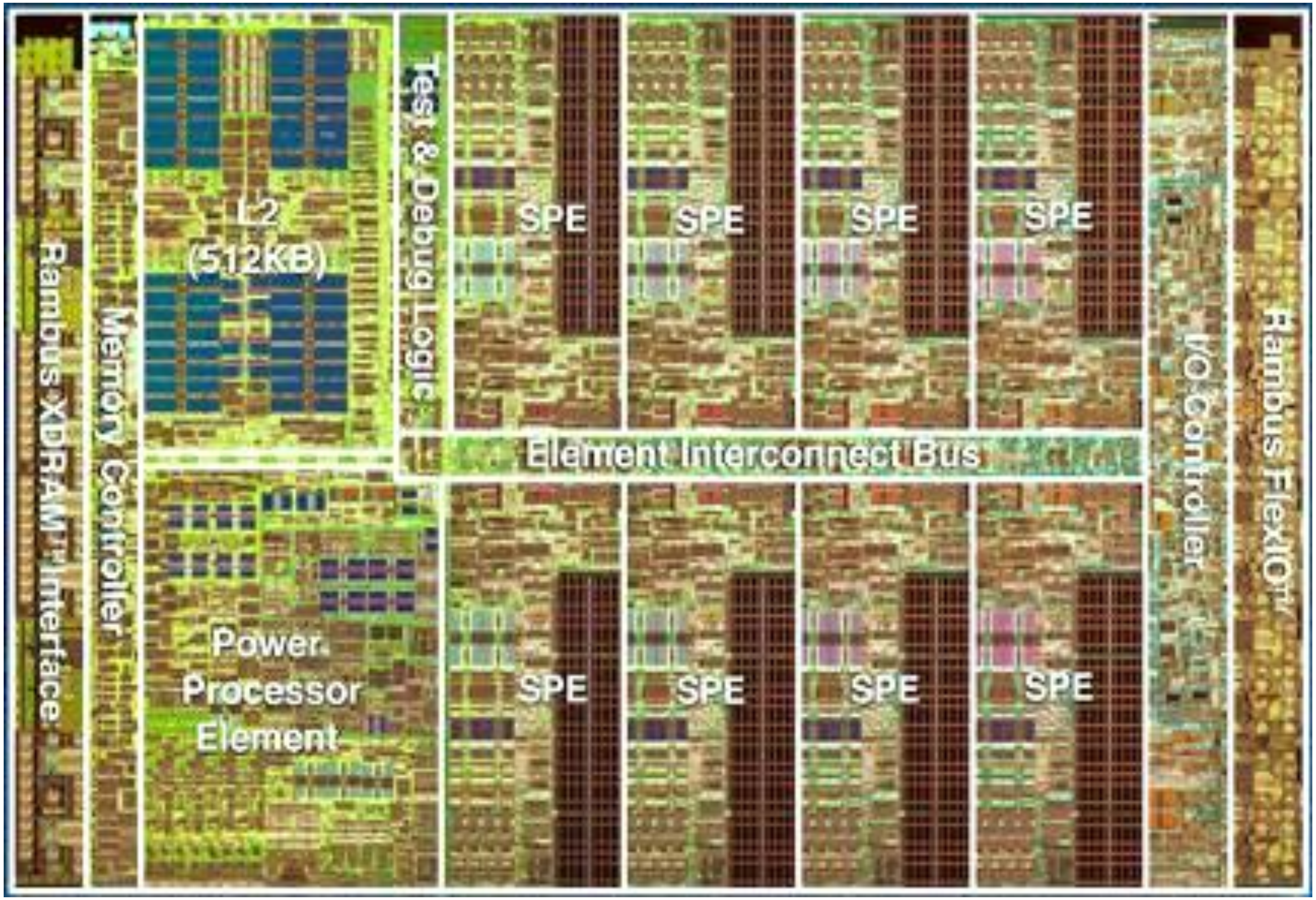
Multicore technology examples

IBM
Power 7



Multicore technology examples

IBM CELL BE (out of production …)



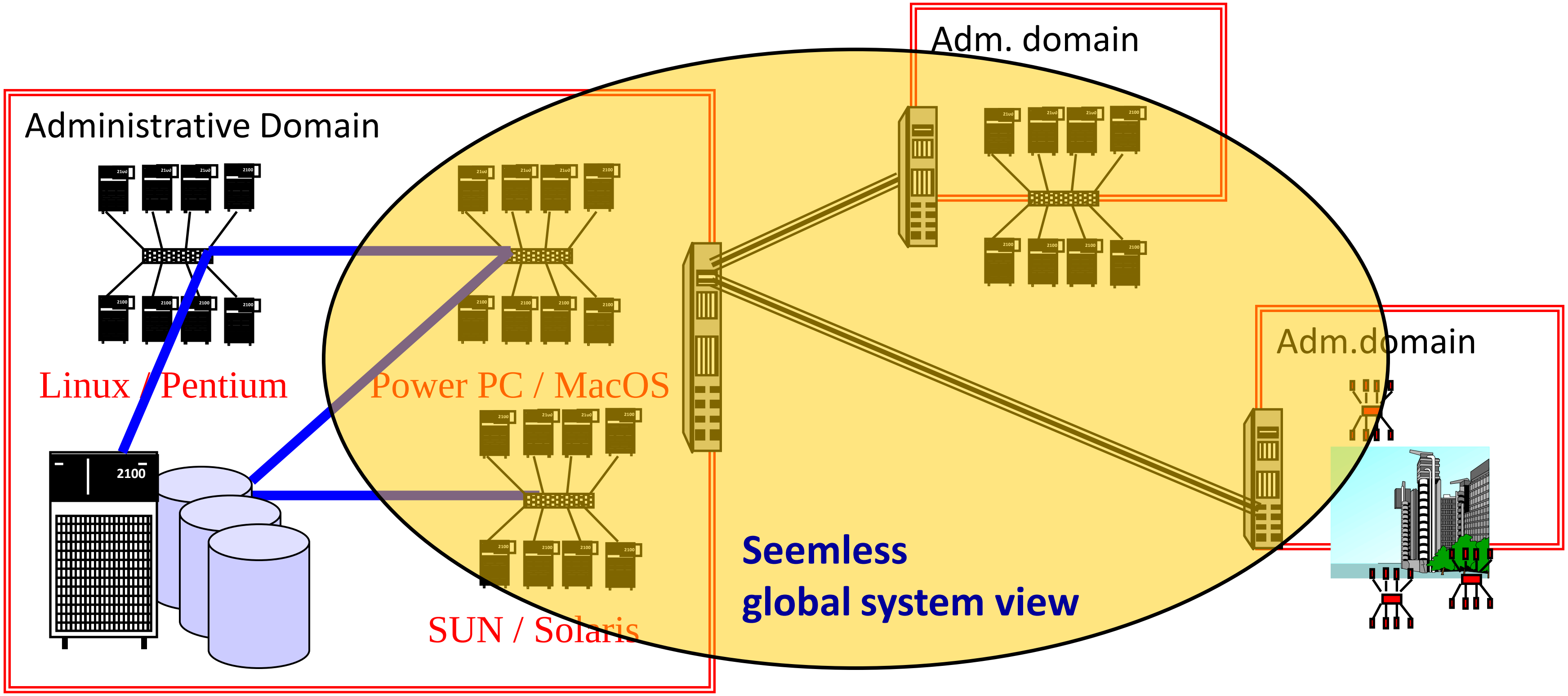
HPC enabling platforms

Homogeneous Clustes, in general with multiprocessor/multicore nodes (SMP, NUMA, …)

Heterogeneous Clusters

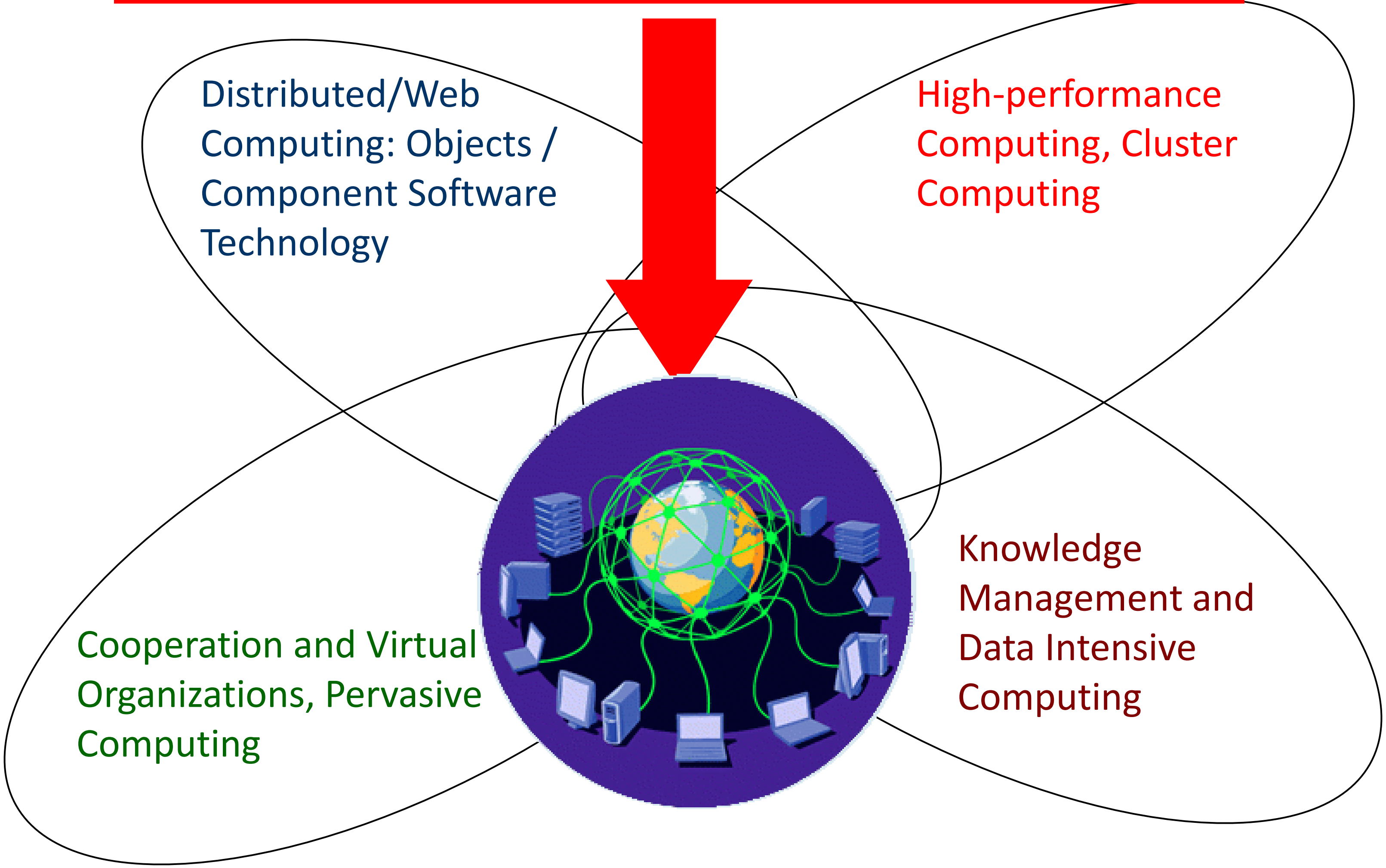
Virtual Private Networks: Farms, Data Centres

Large Scale Platforms (LAN, MAN, WAN): Grids, Clouds, …



Large scale platforms (Grids and much more)

Added value: Quality of Service (QoS)



Distributed/Web Computing: Objects / Component Software Technology

High-performance Computing, Cluster Computing

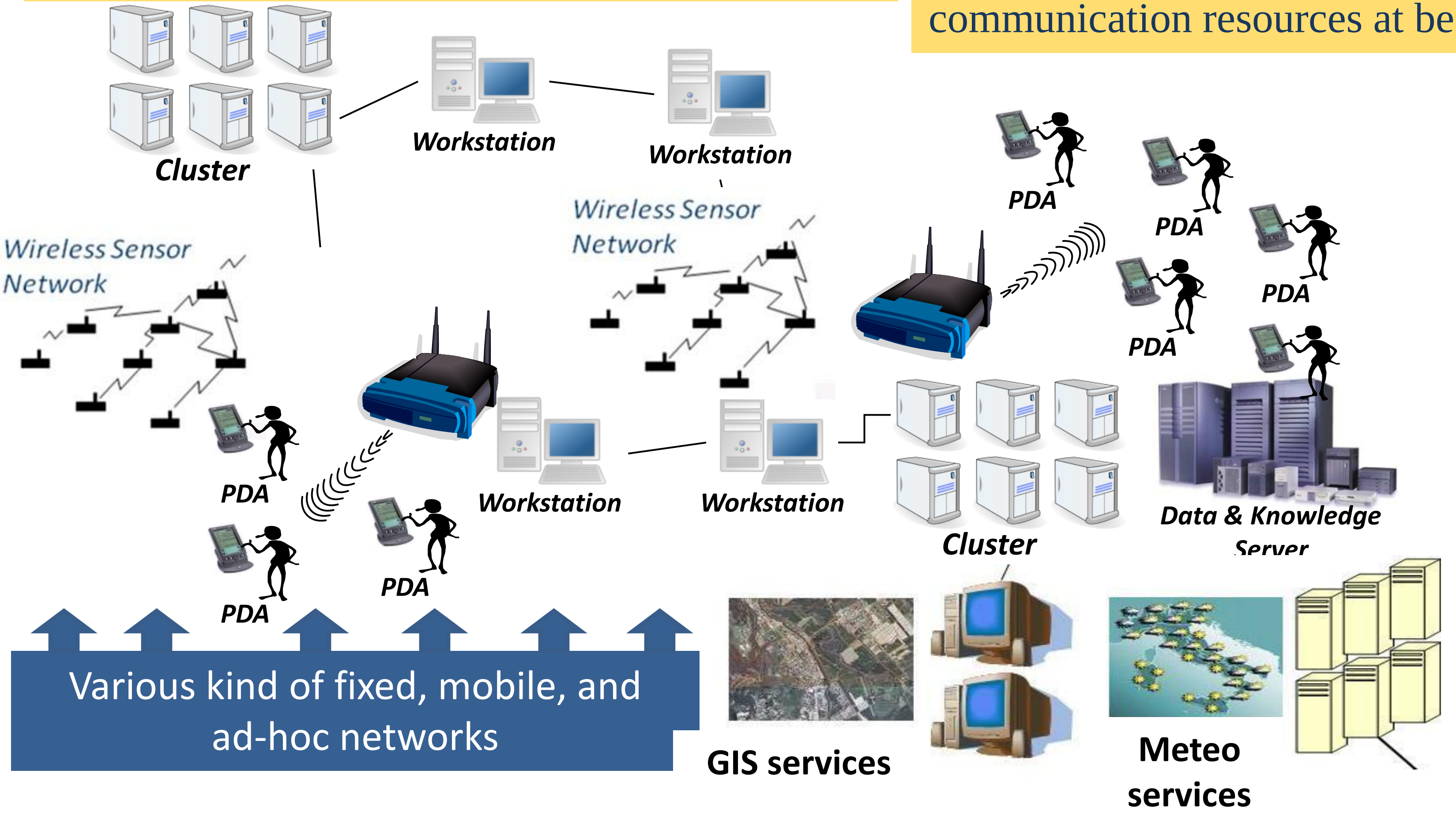
Cooperation and Virtual Organizations, Pervasive Computing

Knowledge Management and Data Intensive Computing

Example of heterogeneous distributed HPC platform: Pervasive Grid

An **integrated** system composed of central servers and services, fixed and mobile decentralized nodes, various kind of networks.

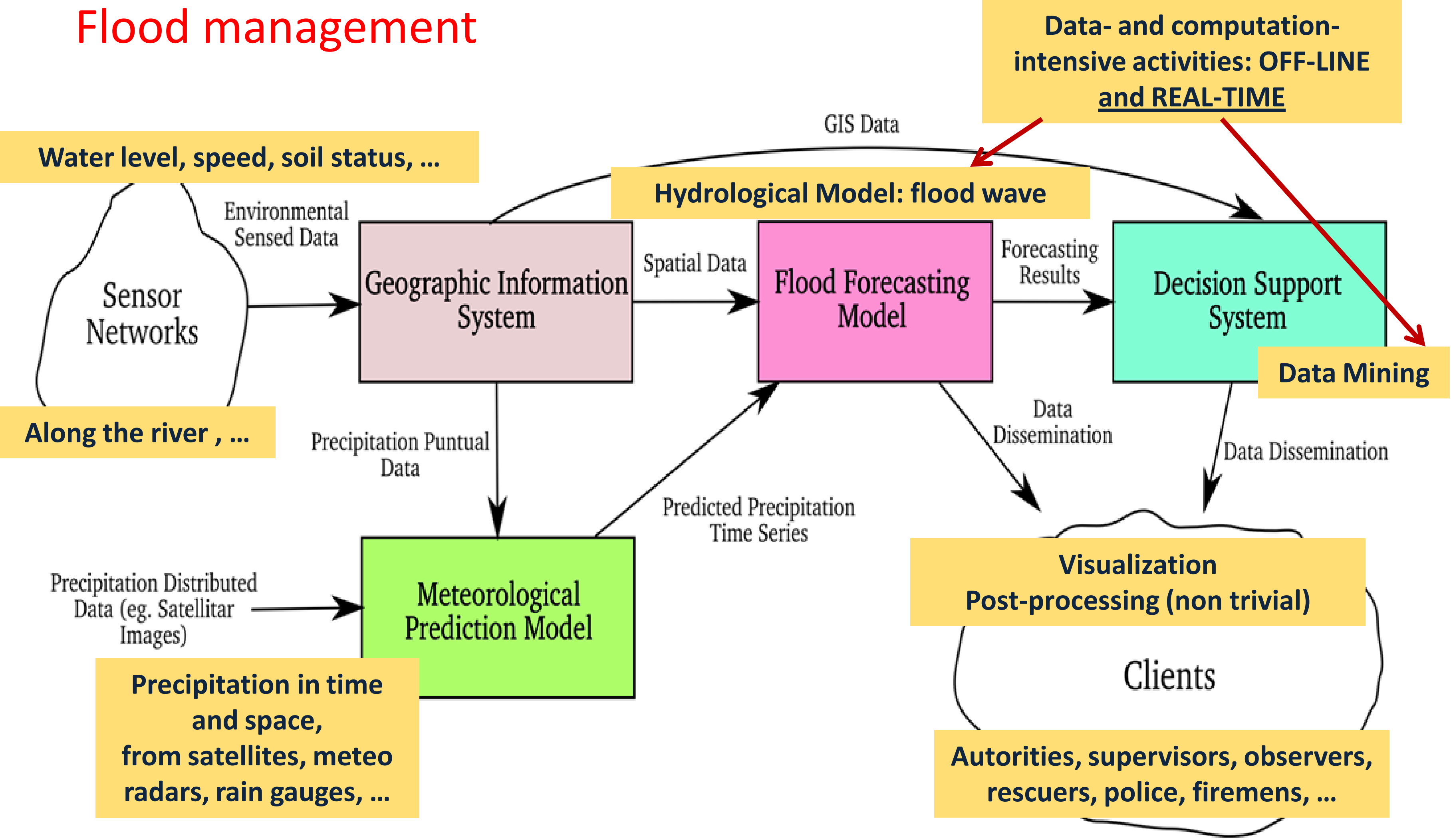
A **distributed application** , *e.g. emergency management*, must be able to exploit **all** the processing **and** communication resources at best.



Various kind of fixed, mobile, and ad-hoc networks

Example of Pervasive Grid application

Flood management



Sensor Networks
Environmental Sensed Data
Geographic Information System
GIS Data
Spatial Data
Flood Forecasting Model
Forecasting Results
Decision Support System
Precipitation Puntual Data
Predicted Precipitation Time Series
Data Dissemination
Data Dissemination
Precipitation Distributed Data (eg. Satellitar Images)
Meteorological Prediction Model
Clients

Data- and computation-intensive activities: OFF-LINE and REAL-TIME

Water level, speed, soil status, …

Hydrological Model: flood wave

Data Mining

Along the river , …

Visualization
Post-processing (non trivial)

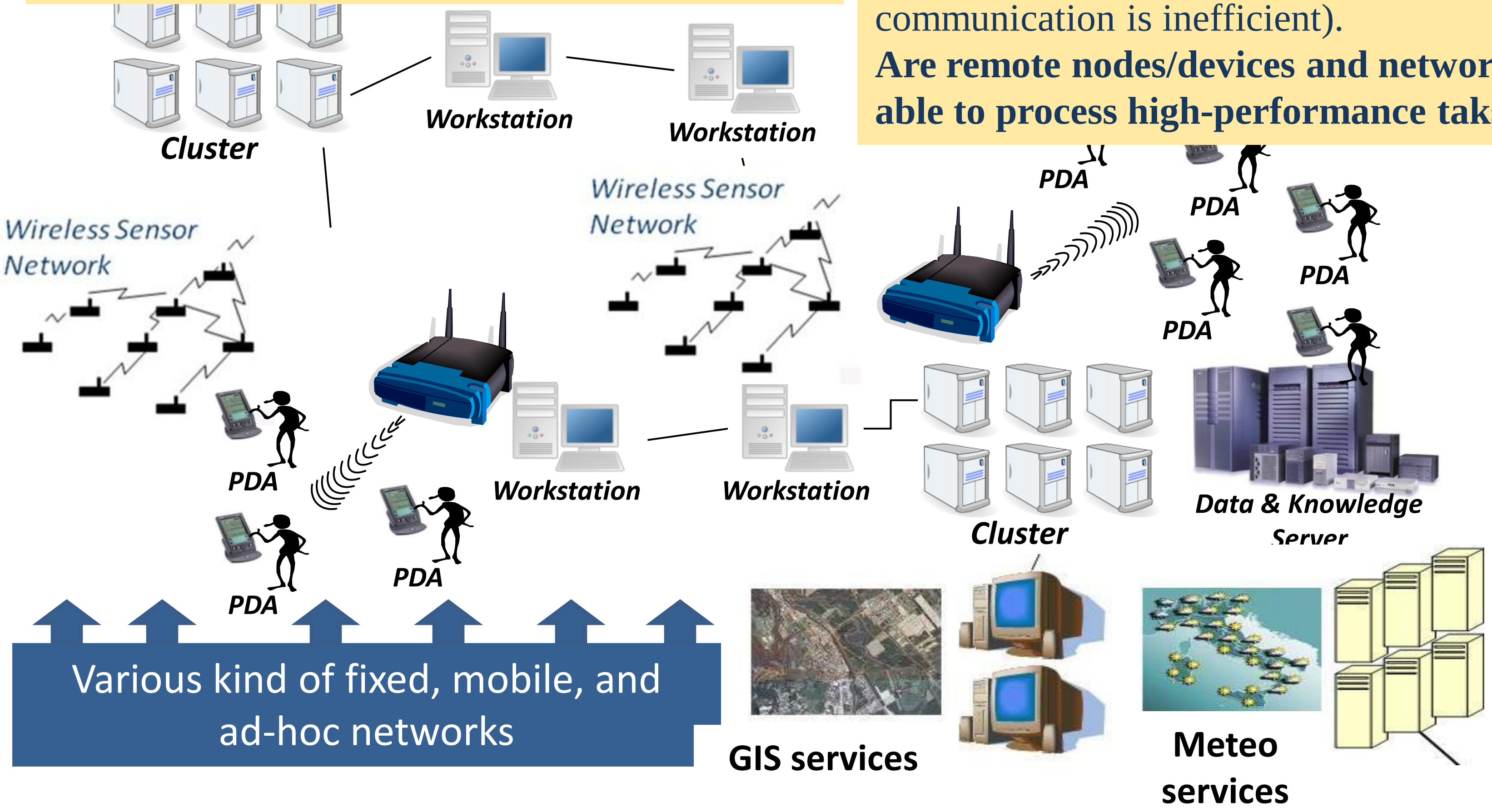
Precipitation in time and space,
from satellites, meteo radars, rain gauges, …

Autorities, supervisors, observers, rescuers, police, firemens, …

Off-line *vs* real-time *adaptive* processing

The tipical **off-line**, routinely tasks involve central servers and some predefined networks. Mobile remote devices are used mainly for communication and visualization.

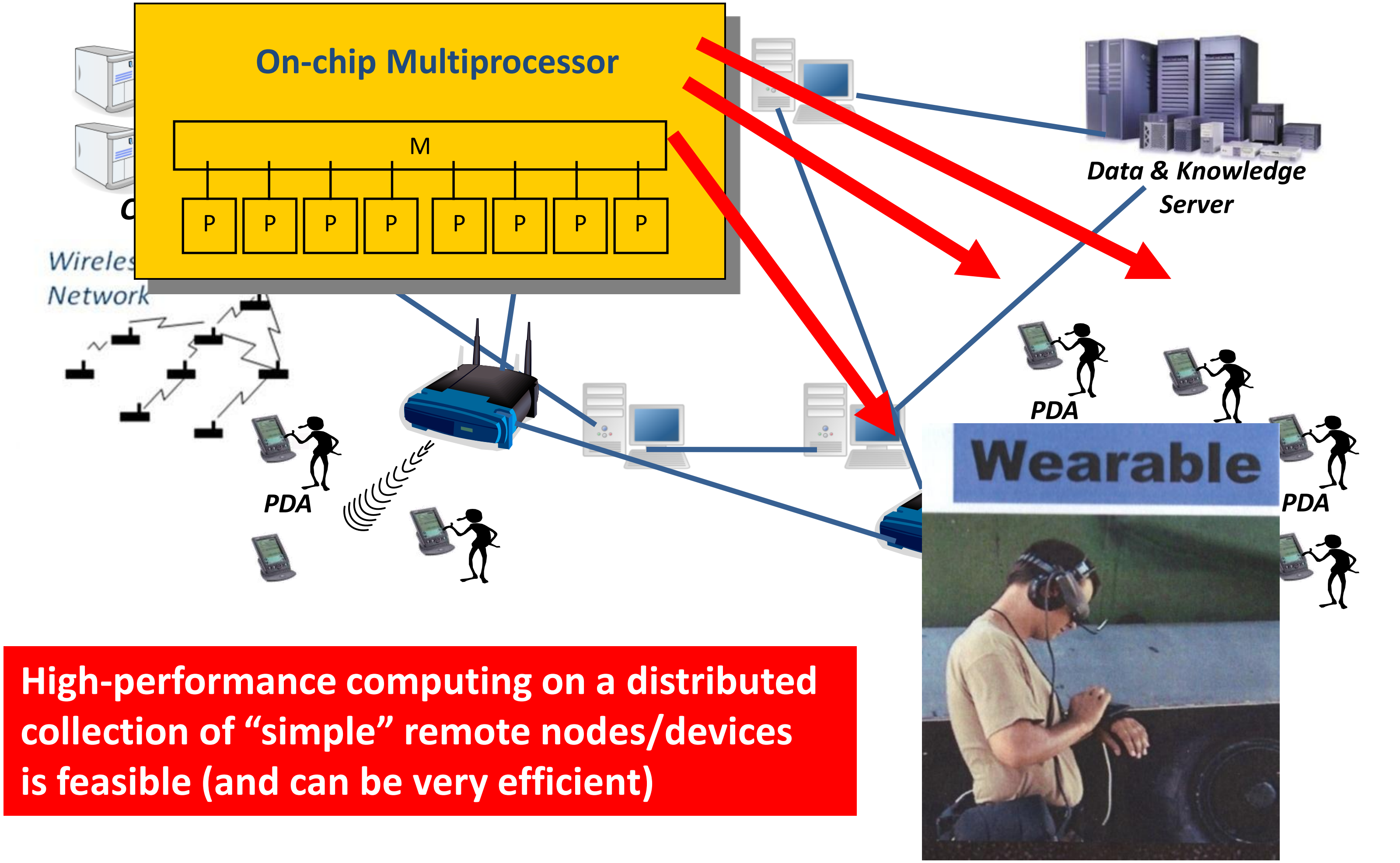
In emergency situations, tasks can be **re-allocated to remote nodes/devices and mobile networks in real-time** (e.g. central resources are disconnected or communication is inefficient).
Are remote nodes/devices and networks able to process high-performance taks?



Various kind of fixed, mobile, and ad-hoc networks

The impact of multicore on next-generation dedicated and mobile technology

Embedding into mobile and /or wearable intelligent devices



High-performance computing on a distributed collection of "simple" remote nodes/devices is feasible (and can be very efficient)

Contents

1. Objectives, motivations, approach
2. An informal presentation of some concepts and technologies
3. Background and prerequisites
4. Course program
5. Course material/notes
6. Exam modality and working approach

Basic background and prerequisites

- An undergraduate-level course on *structured* computer architecture
  - Firmware level structuring
  - Assembler level, CPU architecture, compiling
  - Memory hierarchies and caching
  - Interrupt handling, exception handling
  - Process level, addressing space, low level scheduling, interprocess communication
  - Input/Output processing
- Few books adopt a *structured* approach:
  - Tanenbaum: in principle, some parts only
  - Patterson-Hennessy: mainly description of existing technologies (few concepts)
- In Pisa: course “Computer Architecture”
  - *Book: M. Vanneschi, “Architettura degli Elaboratori”, PLUS (Pisa University Press), 2009 – in Italian !*
- **Some initial lectures will review the basic concepts of the *structured* approach**
  - Students are strongly invited to attend this part in a very critical manner

Background from other courses of MCSN

- Course by Prof. Marco Danelutto (Distributed Systems: Paradigms and Models)
  - Structures of parallel computations
  - Performance measures and cost models
    - Service time / bandwidth, latency, completion time, efficiency, scalability
  - Basic mechanisms for process cooperation (messages, shared variables)
  - Parallelism forms / paradigms, structured parallelism
    - Stream-parallel pipeline
    - Stream-parallel farm
    - Data-flow
    - Data-parallel map, reduce, parallel prefix
    - Data-parallel with stencils
  - Client-server computations
  - Impact of service-time and latency on client service-time

- Course of Advanced Programming
- Basic elements of Queueing Theory
- Basic elements of Networking

Contents

1. Objectives, motivations, approach
2. An informal presentation of some concepts and technologies
3. Background and prerequisites
4. Course program
5. Course material/notes
6. Exam modality and working approach

Course Program

PART 1 (~ 1/4)

1. Prerequisites revisited
   - Firmware structuring; processors, memory hierarchies and caching; assembler level and compiler optimizations, performance parameters; process cooperation and implementation
2. Run-time support to concurrency mechanisms
   - Structured interpretation of process communication and sharing
3. Instruction level parallelism
   - Elements of pipeline and superscalar CPUs, cost models, compiler optimizations

PART 2 (~ 3/4)

4. Shared memory architectures
   - SMP, NUMA, …, interconnection networks, support to concurrency mechanisms, cost models, static and dynamic optimizations, parallel application benchmarks
5. Distributed memory architectures
   - Cluster, MPP, …, interconnection networks, support to concurrency mechanisms, cost models, static and dynamic optimizations, parallel application benchmarks
6. Multicore architectures
   - Current status and trends of single-chip shared/distributed memory architectures

Further MCSN courses on these subjects

- Complements of Distributed Enabling Platforms (CAP)
- Programming Tools for Parallel and Distributed Systems (SPD)
  - **For the current year only (free-choice exam)**: merged into the same course (formally: **SPD**)
    - Grid and Cloud,
    - Distibuted Operating Systems,
    - Tools and Libraries for Parallel and Distributed Machines (MPI and other standards or commercial products; ASSIST–University of Pisa and possibly other research tools)
    - Virtualization,
    - Scheduling.
  - **Next year (study plan)**: 2 distinct courses (CAP: 6 CFU, SPD: 9 CFU)
- Parallel and Distributed Algorithms **Next year**

To increase the knowledge of some notable application paradigms:

- Numerical Techniques and Applications (TNA)
- Network Optimization Methods **Next year**
- Data Mining Techniques **Next year**

Contents

1. Objectives, motivations, approach
2. An informal presentation of some concepts and technologies
3. Background and prerequisites
4. Course program
5. Course material/notes
6. Exam modality and working approach

The lectures

- **Slides and blackboard**
- Slides for
  - (part of) course material
  - lecture outline
- Blackboard
- ~ for each slide, where necessary or convenient: further explanation / discussion using the blackboard

Course Material

- My page:
  - www.di.unipi.it/~vannesch section: Teaching (*)
  - Link in DidaWiki: http://www.cli.di.unipi.it/doku/doku.php/magistraleinformaticanetworking/spa/start
- Lecture Notes
  - Slides (*)
  - Documents (*)
- Papers and selected book chapters
- M. Vanneschi, “Architettura degli Elaboratori”, PLUS, 2009
  - Part IV
  - English version: next year
  - Some parts will by translated in English during the course (*)
- Reference Books
  - *D.A. Patterson, J.H. Hennessy*, “Computer Organization and Design: the Hardware/Software Interface”, Morgan Kaufman Publishers Inc.
  - *D.E. Culler, J.P. Singh, A. Gupta*, “Parallel Computer Architecture: a Hardware/Software Approach”, Morgan Kaufman Publishers Inc.

Contents

1. Objectives, motivations, approach
2. An informal presentation of some concepts and technologies
3. Background and prerequisites
4. Course program
5. Course material/notes
6. Exam modality and working approach

Exam modality

For all students:

Written test + oral test (in English or in Italian)

Written test: explanation/discussion of concepts and techniques of the course, not necessarily focused on small exercizes. Emphasis will be put on the knowledge of methodologies and their application, as well as on the synthesis capability and on the student's ability to establish the proper relationships between the various parts of the course.

Optional: a **report** on a specific topic

- individual written report, maximum 2 persons

- No intermediate tests
- Report to be submitted a certain time in advance wrt the exam date.

- **ASE:** see a subsequent slide.

- *Registration* to the exam on the Official Site of Corso di Laurea: http://compass2.di.unipi.it/didattica, section Laurea Magistrale in Informatica e Netwoking, subsection "orari"

Exam modality

Report

- Some literature material (e.g. one/some papers) is assigned to the student
  - existing parallel machines / multicore, or existing projects,
  - specific techniques and/or technologies on topics of interest .
- The assigned material must be studied and interpreted **according to the course contents, methodology and approach**
- The report must be written in a **didactic style**, as it were a book chapter for students ("student-proof")
  - " if an author is not able to explain a certain thing in an understandable and complete manner, then certainly such thing is not clear to the author himself "
- Literature assigned during the first 2-3 weeks of the course

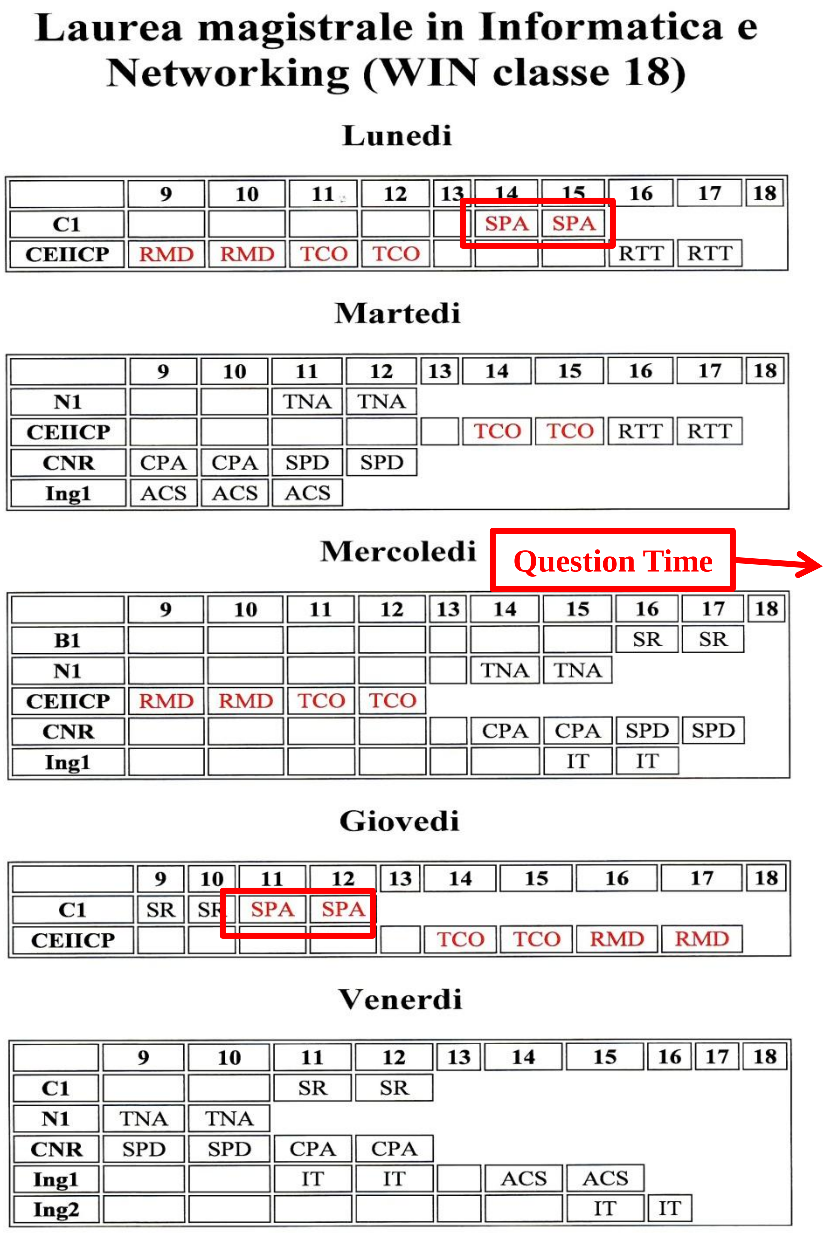
Architetture Parallele e Distribuite (ASE)

Esame di ASE (9 CFU), laurea specialistica vecchio ordinamento:

- **SPA** (6 CFU)
- **+ integrazione 3 CFU** sulla parte delle metodologie di parallelizzazione (Libro Vanneschi, Cap. X)
- **Modalità di esame**: scritto tradizionale e orale
- All’atto dell’iscrizione: indicare ASE

Working approach

- As in any other course, it is fundamental to acquire skills and capabilities in concepts and principles, besides knowing the technologies.
- **Critical aptitude** must be properly developed.
- **Interaction** with the teacher is strongly encouraged
  - Questions during the lectures
  - Question time **(“orario di ricevimento”)** (in Italian for Italians)
    - **Wednesday, 14:30 – 17:30, in my room**
    - *or by appointment in case of collision with other courses.*



Laurea magistrale in Informatica e Networking (WIN classe 18)

Lunedi

	9	10	11	12	13	14	15	16	17	18
C1						SPA	SPA			
CEIICP	RMD	RMD	TCO	TCO				RTT	RTT	

Martedi

	9	10	11	12	13	14	15	16	17	18
N1			TNA	TNA						
CEIICP						TCO	TCO	RTT	RTT	
CNR	CPA	CPA	SPD	SPD						
Ing1	ACS	ACS	ACS							

Mercoledi

Question Time

	9	10	11	12	13	14	15	16	17	18
B1								SR	SR	
N1						TNA	TNA			
CEIICP	RMD	RMD	TCO	TCO						
CNR						CPA	CPA	SPD	SPD	
Ing1							IT	IT		

Giovedi

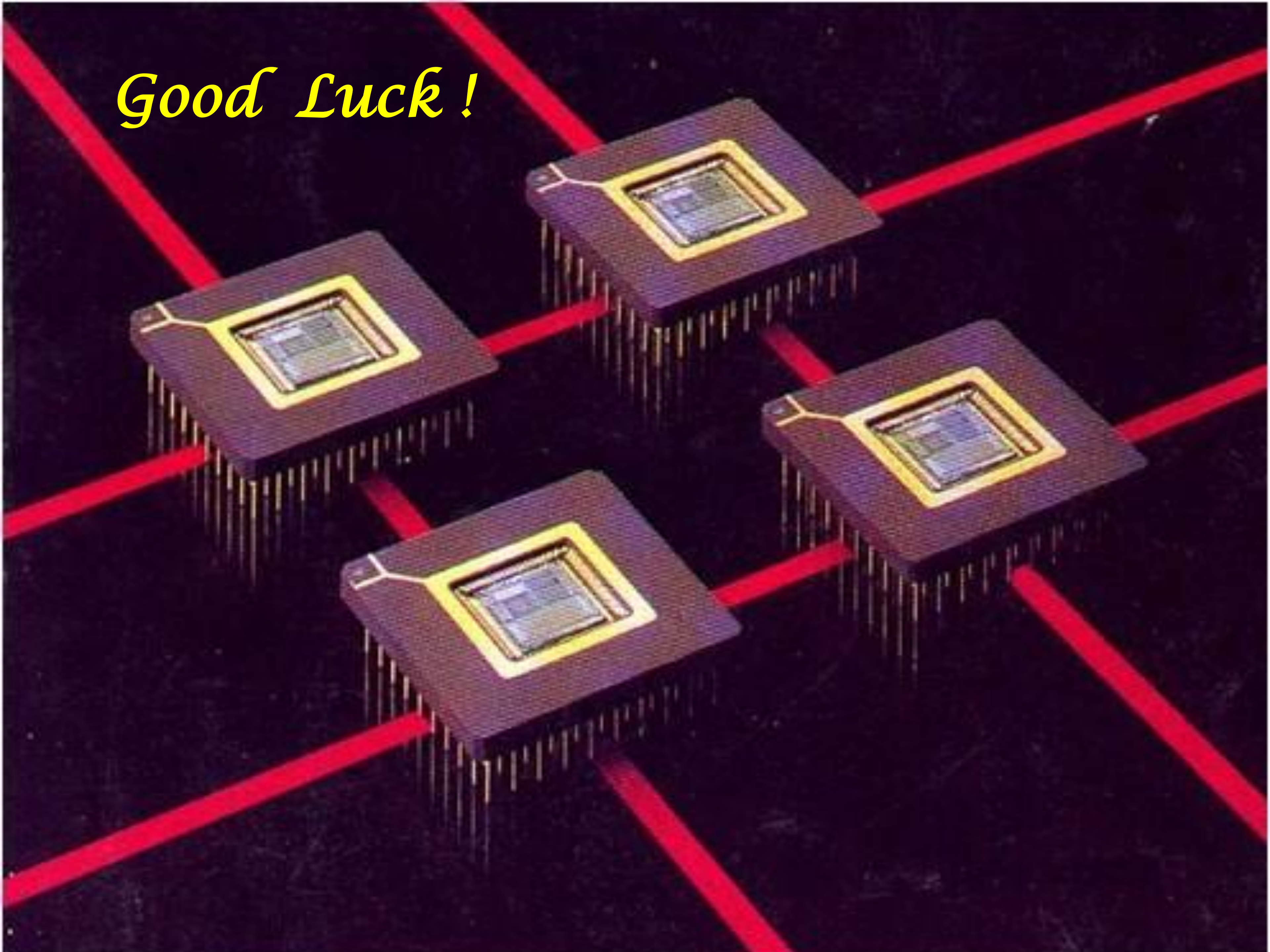
	9	10	11	12	13	14	15	16	17	18
C1	SR	SR	SPA	SPA						
CEIICP						TCO	TCO	RMD	RMD	

Venerdi

	9	10	11	12	13	14	15	16	17	18
C1			SR	SR						
N1	TNA	TNA								
CNR	SPD	SPD	CPA	CPA						
Ing1			IT	IT		ACS	ACS			
Ing2							IT	IT		

14:30 -17:30, my room (Dept)

or by appointment
in case the student attends
the Wednsday afternoon
lectures



Good Luck !