Computação Paralela e Distribuída

CERN School of Computing 2014

25 August to 05 September 2014 in Braga, Portugal

Organized in collaboration with
University of Minho and LIP, Braga, Portugal

Director
Alberto Pace, CERN

Local Organising Committee Chair
Antonio Pina
Vitor Oliveira

Technical Manager
Giuseppe Lo Presti, CERN

School Administrator
Yasemin Hauser, CERN

E-mail: computing.school@cern.ch

Deadline for Application
16th May 2014

Base Technologies
Data Technologies
Physics Computing

Lecturers

François Flückiger, CERN, Geneva, Switzerland
Benedikt Hegner, CERN, Geneva, Switzerland
Robert G. Jacobsen, University of California at Berkeley, USA
Sverre Jarp, CERN, Geneva, Switzerland
Sebastian Lopienski, CERN, Geneva, Switzerland
Pere Mato, CERN, Geneva, Switzerland
Andrzej Nowak, CERN, Geneva, Switzerland
Alberto Pace, CERN, Geneva, Switzerland
Andreas J. Peters, CERN, Geneva, Switzerland
Danilo Piparo, CERN, Geneva, Switzerland
Ivica Puljak, University of Split, Croatia
Arnulf Quadt, Universität Göttingen, Germany
Benjamin Radburn-Smith, Purdue University, West Lafayette, USA

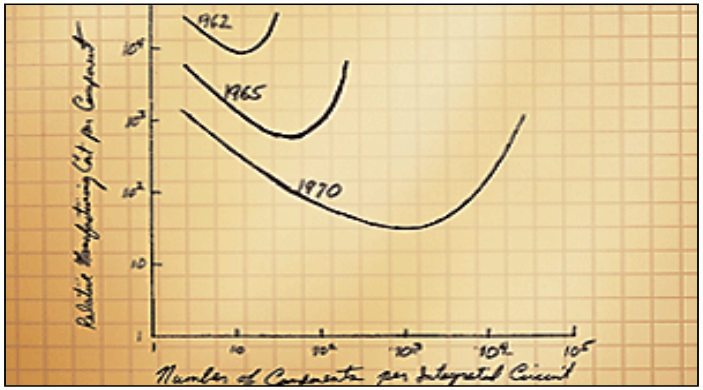
www.cern.ch/CSC

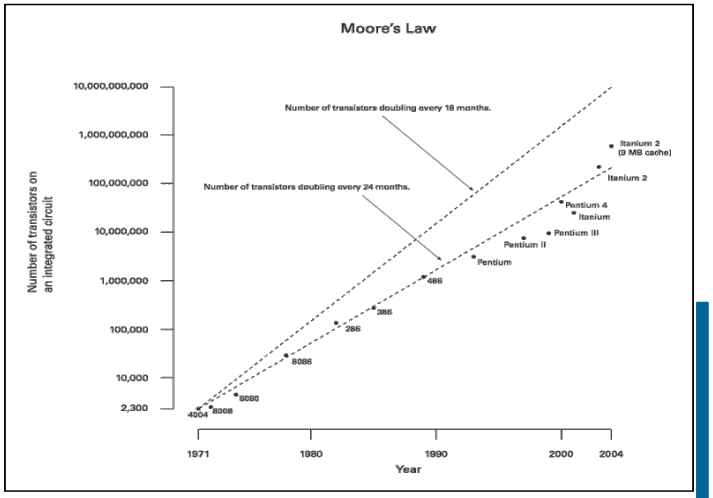
Computação Paralela e Distribuída

CERN School of Computing

Moore's "law"

- **A marching order established ~50 years ago**
  - "Let's continue to double the number of transistors every other year!"
- **First published as:**
  - Moore, G.E.: *Cramming more components onto integrated circuits.* Electronics, 38(8), April 1965.
- **Accepted by all partners:**
  - Semiconductor manufacturers
  - Hardware integrators
  - Software companies
  - Us, the consumers





From Wikipedia

Computação Paralela e Distribuída

CERN School of Computing

Real consequence of Moore's law

- **We are being "snowed under" by transistors:**
  - More (and more complex) execution units
    - Hundreds of new instructions
  - Longer SIMD/SSE hardware vectors
  - More and more cores
  - More hardware threading
- **In order to profit we need to "think parallel"**
  - Data parallelism
  - Task parallelism

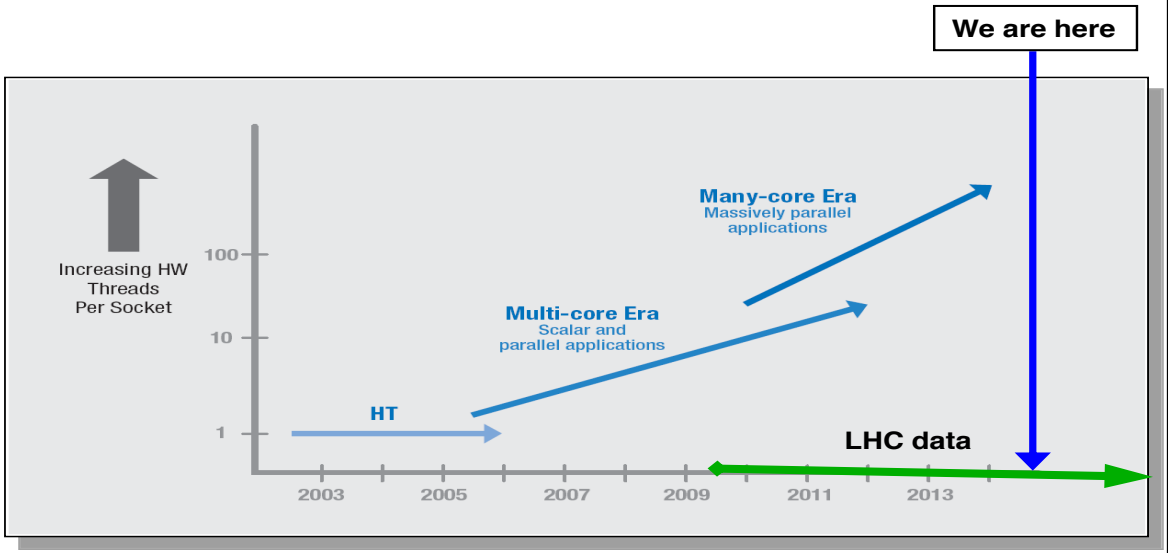
"Data Oriented Design"

Computação Paralela e Distribuída

CERN School of Computing

Semiconductor evolution

- **Today's silicon processes:**
  - 28, 22 nm
- **Being introduced:**
  - 14 nm (2013/14)
- **In research:**
  - 10 nm (2015/16)
  - 7 nm (2017/18)
  - 5 nm (2019/20)
    - Source: Intel



S. Borkar et al. (Intel), "Platform 2015: Intel Platform Evolution for the Next Decade", 2005.

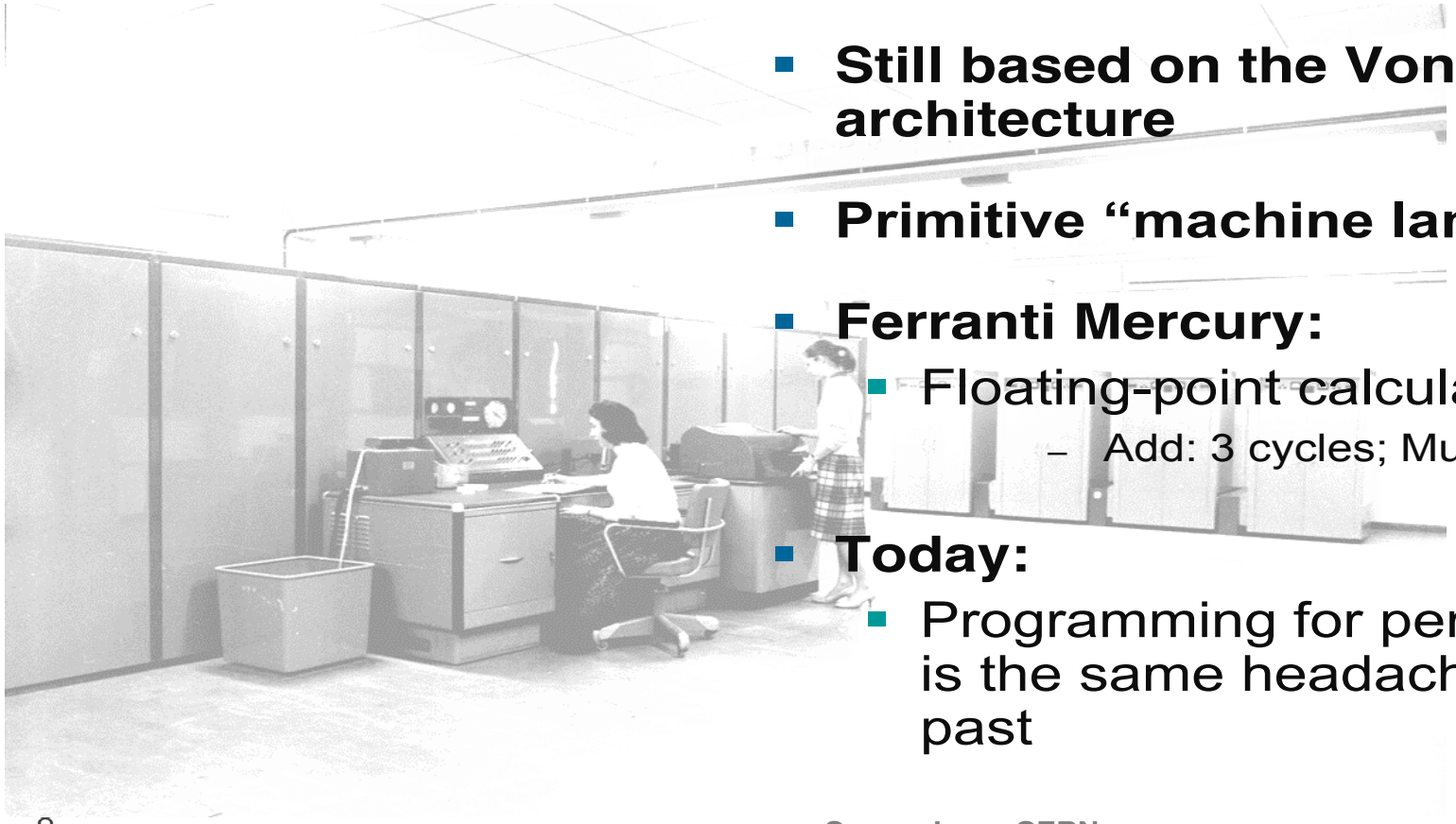
2 nm (2028?) TSMC

- **By the end of this decade we will have chips with ~100'000'000'000 (10^{11}) transistors!**

Computação Paralela e Distribuída

CERN School of Computing

Archaic Computing Units



- **As “stupid” as 50 years ago**
- **Still based on the Von Neumann architecture**
- **Primitive “machine language”**
- **Ferranti Mercury:**
  - Floating-point calculations
    - Add: 3 cycles; Multiply: 5 cycles
- **Today:**
  - Programming for performance is the same headache as in the past

Computação Paralela e Distribuída

CERN School of Computing

Reasons to worry about performance (apart from wanting "more")

- **Moore's law no longer covers inefficient, sloppy or disorganized code**
  - Yes, that means you, C++
- **Different technologies (i.e. storage, memory, processor) progress at different speeds**
- **Still alive from the frequency scaling days: "Performance is not a feature"**
- **The decisions you make today might have a lifespan of decades – thus, smart optimizations will live on… and so will stupid mistakes**
  - In other words, "you can't optimize an elephant into a cheetah"

Quote from http://www.c2.com/cgi/wiki?DesignForPerformance

Computação Paralela e Distribuída

CERN School of Computing

Performance: A complicated story!

- **We start with a concrete, real-life problem to solve**
  - For instance, simulate the passage of elementary particles through matter
- **We write programs in high level languages**
  - C++, JAVA, Python, etc.
- **A compiler (or an interpreter) transforms the high-level code to machine-level code**
- **We link in external libraries**
- **A sophisticated processor with a complex architecture and even more complex micro-architecture executes the code**
- **In most cases, we have little clue as to the efficiency of this transformation process**

Computação Paralela e Distribuída

CERN School of Computing

A Complicated Story (in 9 layers!)

- **Computing problems are solved by getting electrons to "dance"**

Problem
Design, Algorithms, Data
Language, Source program
Compilers, Libraries
System architecture
Instruction set architecture
μ-architecture
Circuits
Electrons

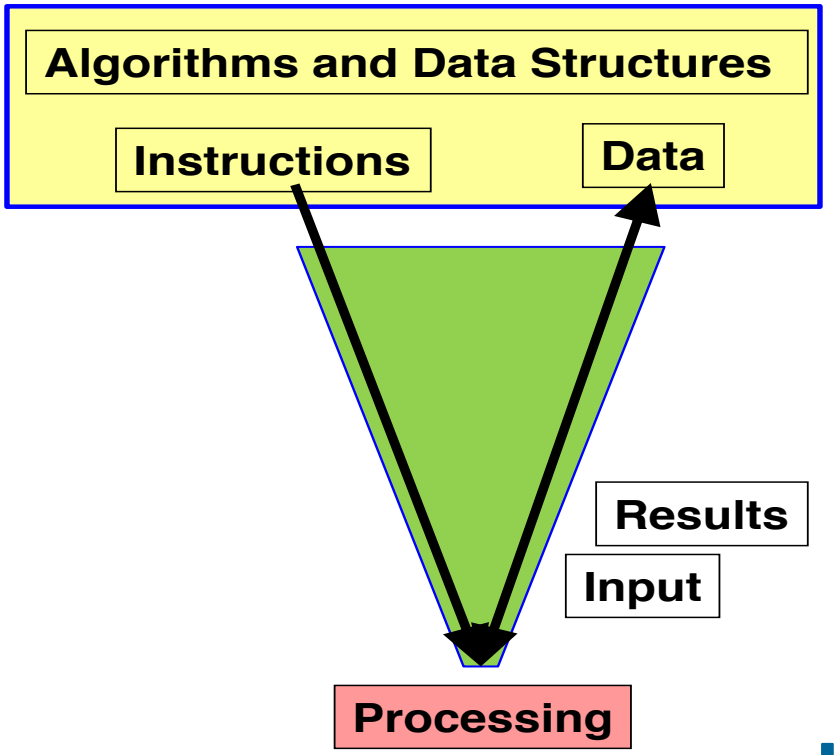
Adapted from Y.Patt, U-Austin

Computação Paralela e Distribuída

CERN School of Computing

Von Neumann architecture

- **From Wikipedia:**
  - The von Neumann architecture is a computer design model that uses a processing unit and a single separate storage structure to hold both instructions and data.
- **It can be viewed as an entity into which one streams instructions and data in order to produce results**



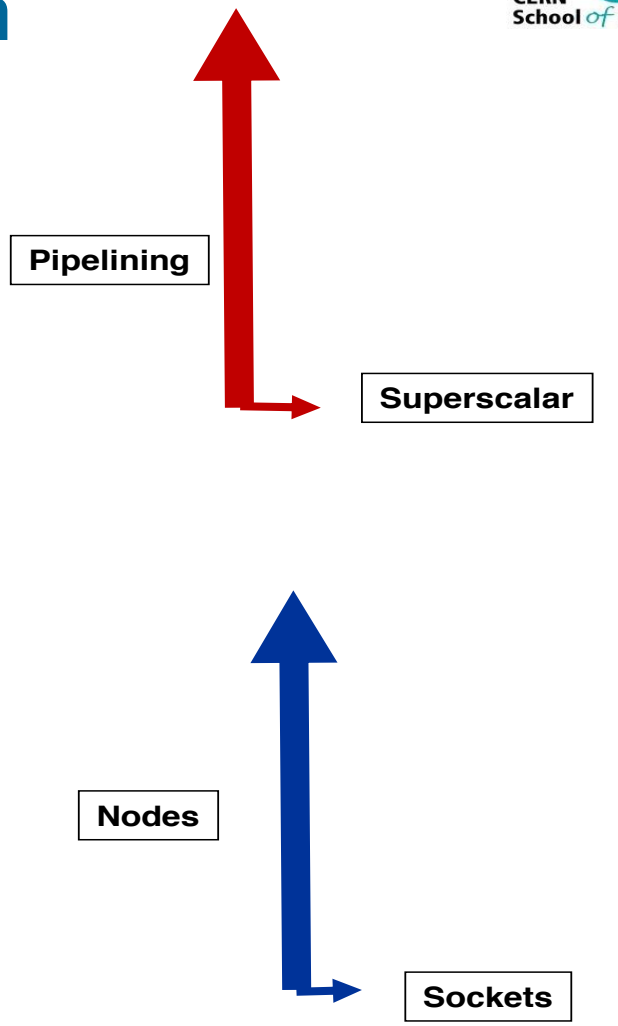
Some people think the architecture is out-dated. But nobody has managed to replace it (yet)

Computação Paralela e Distribuída

CERN School of Computing

In the days of the Pentium

- **Life was really simple:**
  - Basically two dimensions
    - The pipeline and its frequency
    - The number of boxes
  - The semiconductor industry increased the frequency
  - We acquired the right number of (single-socket) boxes

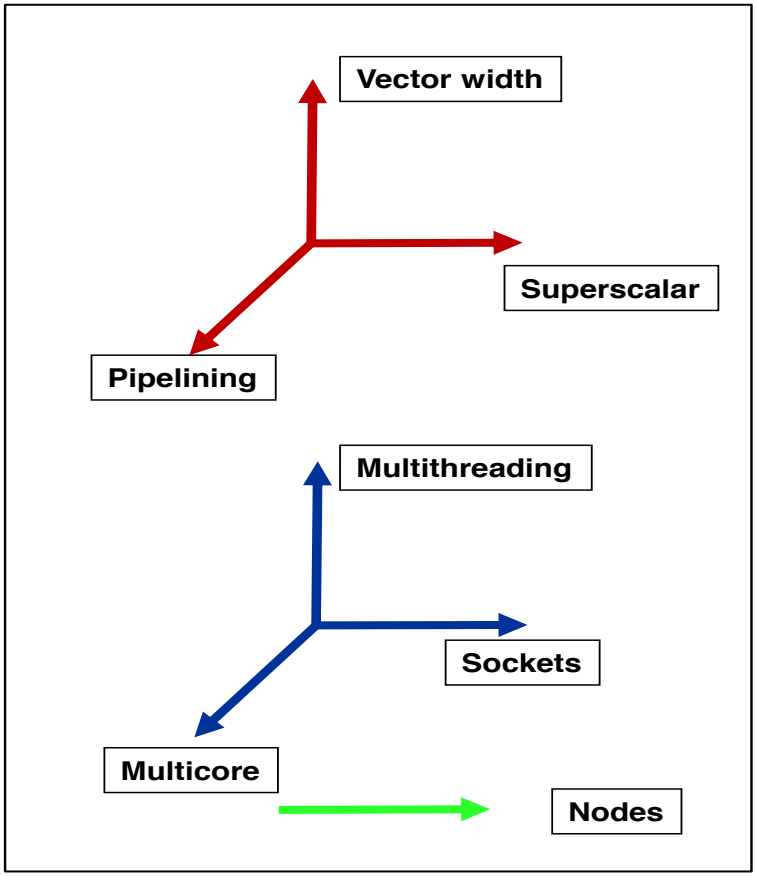


Computação Paralela e Distribuída

CERN School of Computing

Now: Seven dimensions of performance

- **First three dimensions:**
  - Hardware vectors/SIMD
  - Superscalar
  - Pipelining
- **Next dimension is a "pseudo" dimension:**
  - Hardware multithreading
- **Last three dimensions:**
  - Multiple cores
  - Multiple sockets
  - Multiple compute nodes



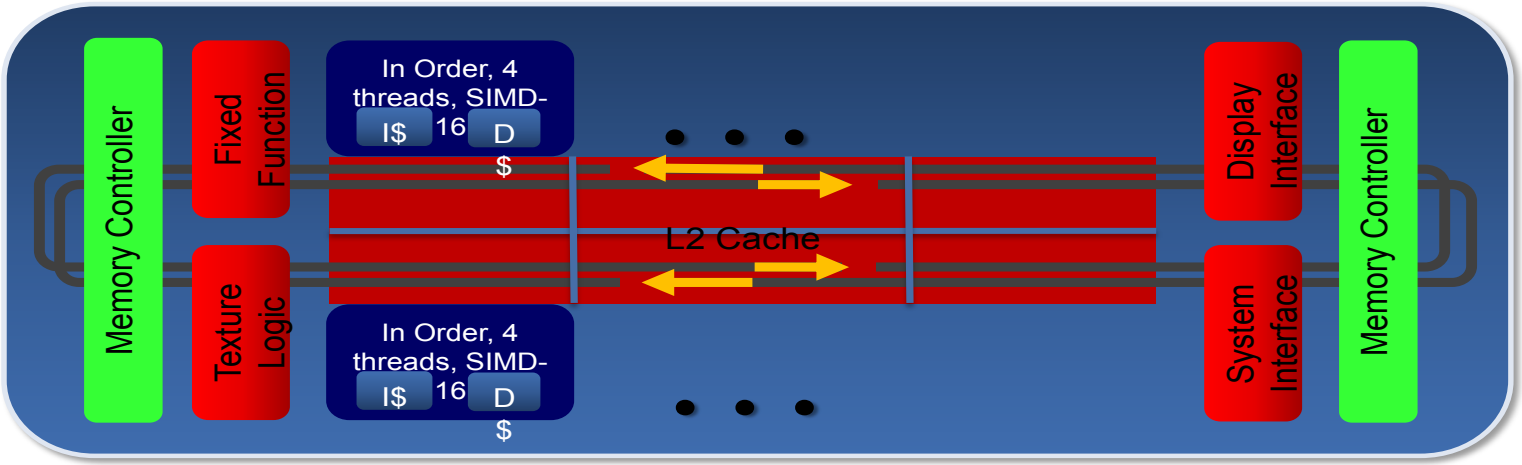
SIMD = Single Instruction Multiple Data

Computação Paralela e Distribuída

CERN School of Computing

Simple, but illustrative example

- **Xeon Phi has ~60 cores, 4-way hardware threading, hardware vectors of size 8 (Double Precision):**
- **Program A: Threaded 60 x 4, vectorised 8x:**
  - Performance potential: 1920
- **Program B: Not threaded: 1x, not vectorised: 1x**
  - Performance potential: 1



Computação Paralela e Distribuída

CERN School of Computing

The move to many-core systems

- **Examples of "CPU slots": Sockets * Cores * HW-threads**
  - Basically what you observe in "cat /proc/cpuinfo"
  - Conservative:
    - Dual-socket AMD six-core (Istanbul): 2 * 6 * 1 = 12
    - Dual-socket Intel six-core (Westmere-EP): 2 * 6 * 2 = 24
  - More aggressive:
    - Quad-socket AMD Interlagos (16-core) 4 * 16 * 1 = 64
    - Quad-socket Westmere-EX "octo-core": 4 * 10 * 2 = 80
- **In the near future: Hundreds of CPU slots !**
  - Octo-socket Oracle/Sun Niagara (T5) processors w/16 cores and 8 threads (each): 8 * 16 * 8 = 1024
- **And, by the time new software is ready: Thousands !!**

Mestrado em Engenharia Informática
Computação Paralela e Distribuída
(2014/2015)

Computação Paralela e Distribuída

- **Docentes**
  - Alberto Proença e João L.Sobral - CCTC, António Pina – LIP
    http://www3.di.uminho.pt/~rcm/mestrados/mei/PT/CPD.html

- **Módulos**
  - Arquiteturas Avançadas
  - Algoritmos Paralelos
  - Engenharia dos Sistemas de Computação
  - Paradigmas de Computação Paralela

- **Recursos**
  - Palestras nas aulas por investigadores convidados em Ciência e Eng, com pb reais, com possibilidade de dissertação http://gec.di.uminho.pt/Discip/MInf/cpd1314/
  - UT Austin|Portugal Program (CoLab)
    - Workshop Internacional UMinho/Portugal (gratuito) http://www.di.uminho.pt/parua14/
    - 5 semanas de estágio de investigação, em Austin, Texas (~ 8 lugares gratuitos) http://utaustinportugal.org/areas/advanced-computing
  - **Search**: Services and Advanced Research Computing with HTC/HPC cluster http://search.di.uminho.pt/wordpress/?page_id=19

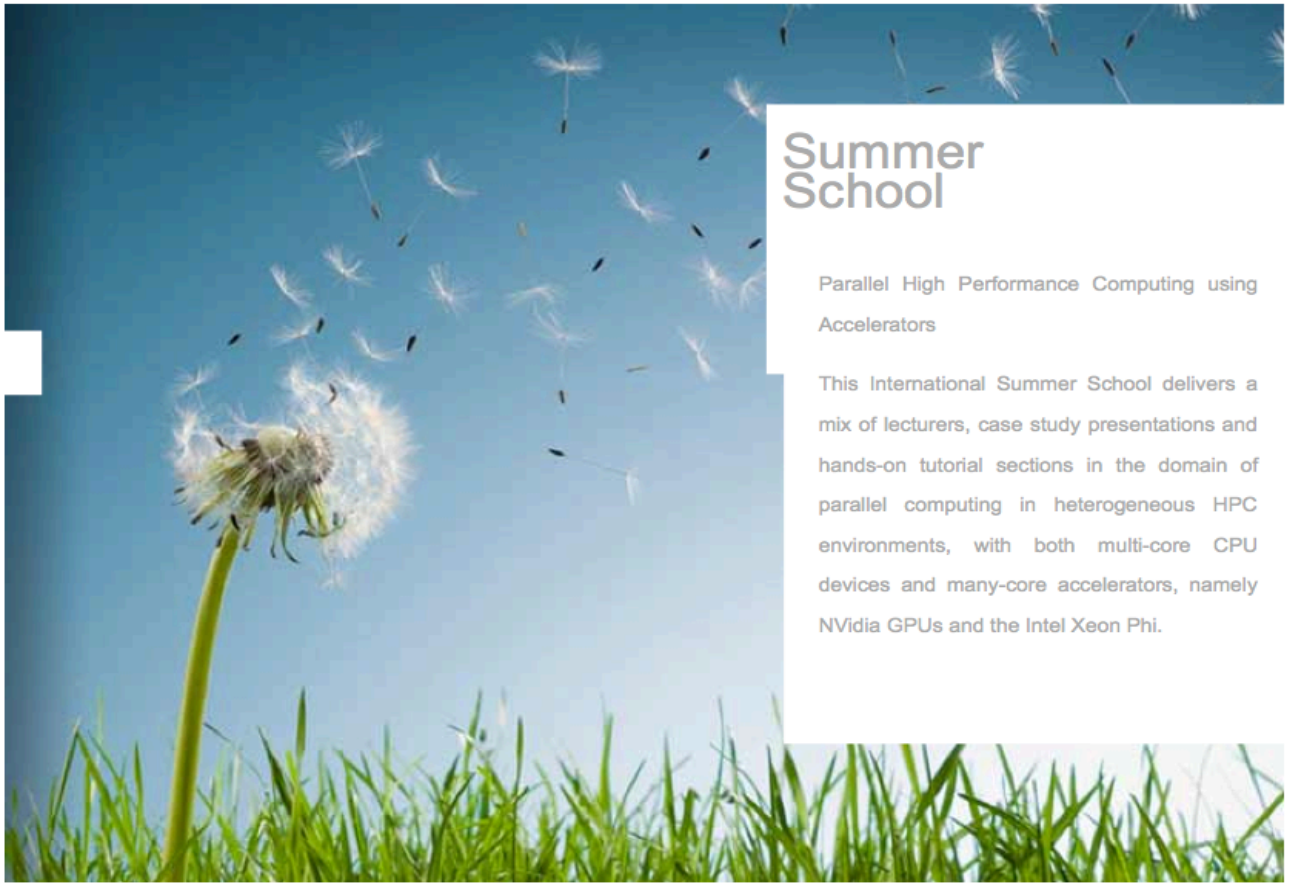
Computação Paralela e Distribuída

Palestras temáticas

- Artur Cavaco-Paulo, Dep. Engª Biológica, *Fine and coarse-grain simulations by molecular dynamics of protein and biological interfaces*
- José Luís Pinho, Dep. Engª Civil, *Delft3D – Software for coastal processes modelling*
- Miguel Nóbrega, Dep. Engª Polímeros, *Computer aided design of extrusion forming tools*
- Stéphane Clain, Dep. Mat & Aplicações, *Modelling and numerical methods in Scientific Computing*
- Luís Alves, Dep. Engª Mecânica, *DD3IMP (Finite Element Solver): the challenge of computational performance improvement through HPC*
- Manuel Melle-Franco, CCTC, *Computer models for (carbon) nanotechnology*
  Miguel Rocha, Dep. Informática, *Graph-based algorithms for metabolic engineering applications*
- Rui Silva, Dep. Informática, *Experience with PAPI performance analysis tool*
- António Onofre / Nuno Castro, Dep. Física, *Code optimization in High Energy Physics - challenges at the LHC*

Computação Paralela e Distribuída

Workshop Internacional



Computação Paralela e Distribuída

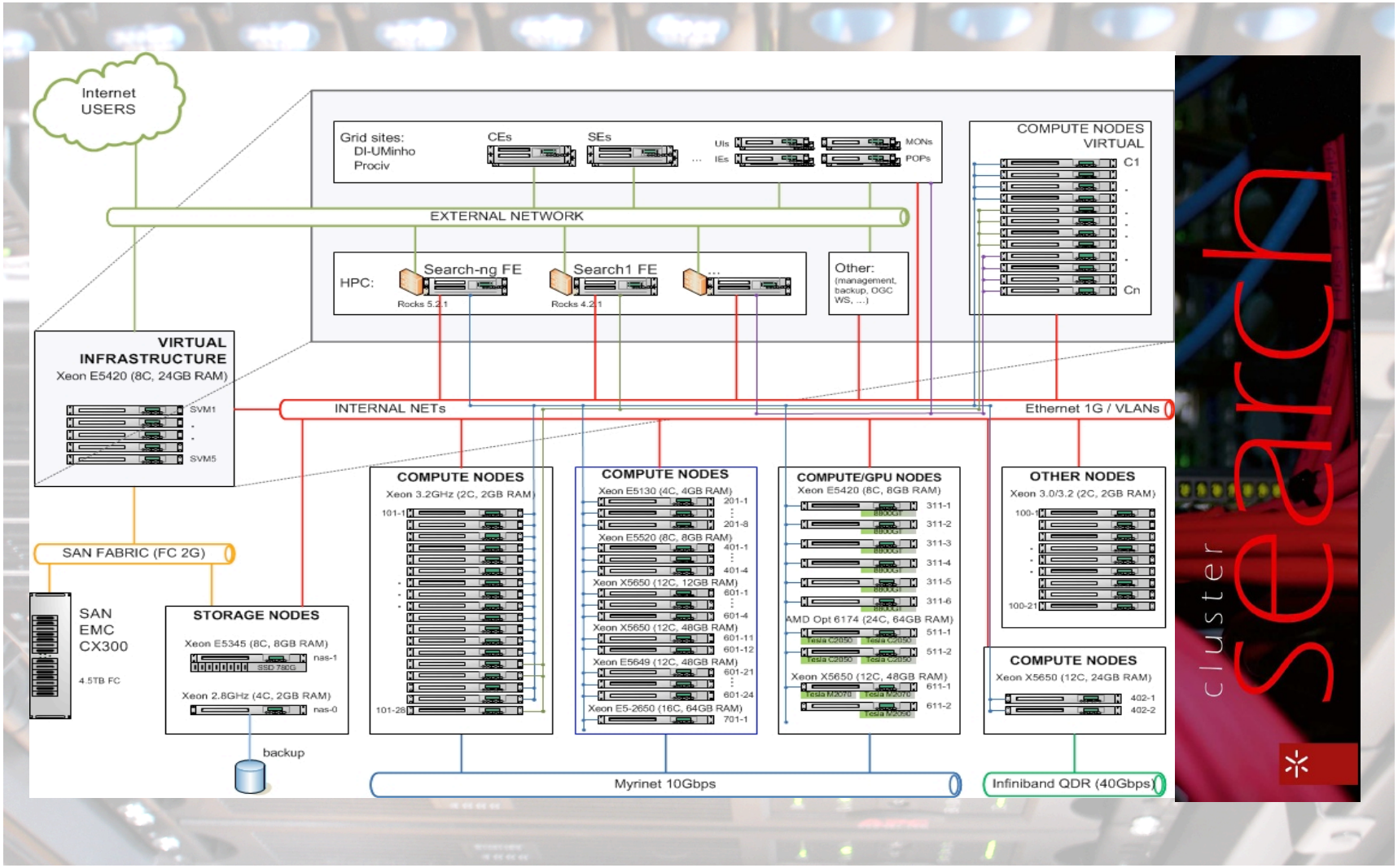
Estágios de investigação em Austin/Texas

Advanced Computing – Portuguese students at the University of Texas at Austin

This summer, eight students from the University of Minho are doing their internships at the University of Texas at Austin. They are working on exciting projects in graphics, high-performance computing, and parallel programming. Daniel Gomes, Jose Alves, and Luis Gomes are working with Professor Don Fussell in the Department of Computer Science at UT Austin on advanced ray-tracing techniques. Fabio Correia and Diogo Lopes are working with Dr. Paul Navratil at TACCC on the Stampede cluster, which is one of the fastest computers in the world. Cristiano Sousa, Rui Brito, David Pereira and Bruno Medeiros are working with Professor Keshav Pingali, co-director of Colab, on graph partitioning problems on multicores and GPUs. The UT Austin research supervisors report that the Portuguese students have an excellent background in parallel programming, and that they are strongly motivated. We look forward to seeing some significant accomplishments by the end of the summer!

Computação Paralela e Distribuída

SeARCH: HTC/HPC clusters



Computação Paralela e Distribuída

- Atualização de nós de computação
  - 20 nós dual processadores E5-Xeon 2650V2 (8 cores, 2,6Ghz), 64GB RAM, SSD de 120GB, c/ gestão remota.
- Tecnologias emergentes
  - 4 nós dual/processador Xeon E5-2695V2, 64GB RAM, SSD de 120GB e dois processadores Xeon Phi 7120X

  - 2 nós dual/processador Xeon E5-2695V2, 64GB RAM, SSD de 120GB e 2 GPUs NVidia Tesla K40M.

- Sistema de armazenamento misto :
  - Uma SAN c/ dois controladores, 4 portas FC de 8Gbps, com 48 discos de 4TB para um total bruto de 48TB (utilizadores)
  - 4 nós com 12TB (scratch e bigdata).

Computação Paralela e Distribuída

- Ajudar a compreender:
  - As mudanças na arquitetura dos computadores.
  - Que nem sempre há um caminho direto para extrair (todo) o desempenho!
- Dar a conhecer as diferentes dimensões do hardware para poder melhorar o projeto de software!
- E pensar, paralelo, paralelo … para poder usar:
  - todos os "sockets e cores"
  - "hardware threads and long vectors"

- E formar profissionais competentes em:
  - Processadores: single-/multi-core, genéricos/especializados
  - Sistemas: homogéneos/heterogéneos, multi-processador/cluster/grid
  - Programação paralela: linguagens, algoritmos, bibliotecas, depuração.

Computação Paralela e Distribuída

Bem Vindos!

http://www3.di.uminho.pt/~rcm/mestrados/mei/PT/CPD.html

António Pina / pina@di.uminho.pt