Latinamerican School for Computational Materials Science.

Introduction to High Performance Computing (and to grid as well)

Stefano Cozzini

CNR-INFM DEMOCRITOS, Trieste

Santiago, Chile - Enero, 2009

Agenda

- Introduction: what is e-science ?
- High Performance Computing:
  - introduction/ concepts /definitions
- Parallel computers
- Clusters:
  - definitions and some other funny things
- Grids
- Wrap-up

in search of E-science

- What is meant by e-Science? In the future, e-Science will refer to the large scale science that will increasingly be carried out through distributed global collaborations enabled by the Internet

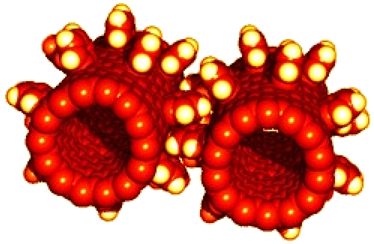
[from http://www.nesc.ac.uk/nesc/define.html]

- The term e-Science (or eScience) is used to describe computationally intensive science that is carried out in highly distributed network environments

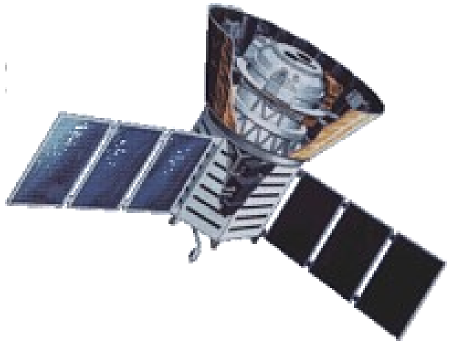
[from wikipedia]

e-science=computationally intensive science

- Science is becoming increasingly digital and needs to deal with increasing amounts of data and computing power
- Simulations get ever more detailed
  - Nanotechnology – design of new materials from the molecular scale
  - Modelling and predicting complex systems (weather forecasting, river floods, earthquake)
  - Decoding the human genome



- Experimental Science uses ever more sophisticated sensors to make precise measurements
  - → Need high statistics
  - → Huge amounts of data
  - → Serves user communities around the world



e-science= new approach to do science

- New tools&methods
  - powerful and modern
    - hardware
    - software

High Performance Computing

  - pooling of resources geographically distributed
  - distribute collaborations

GRID COMPUTING

  - IT- skilled computational scientists

TRAINING

High Performance Computing (HPC)

- performance is everything (well, almost everything):
- I want ...
    - my calculation run faster and faster...
- it ranges from your laptop to the cutting-edge supercomputers
- it is not only on hardware but involves software and people as well

How to run application faster ?

- There are 3 ways to improve performance:
  - Work Harder
  - Work Smarter
  - Get Help
- Computer Analogy
  - Using faster hardware
  - Optimized algorithms and techniques used to solve computational tasks

    Learn how to use tools and optimized your code
  - Multiple computers to solve a particular task

    Parallel computing ()

defining parallel computing

- Parallel computing is the simultaneous execution of the same task (split up and specially adapted) on multiple processors in order to obtain results faster.
- The process of solving a problem usually can be divided into smaller tasks, which may be carried out simultaneously with some coordination.

[from wikipedia]

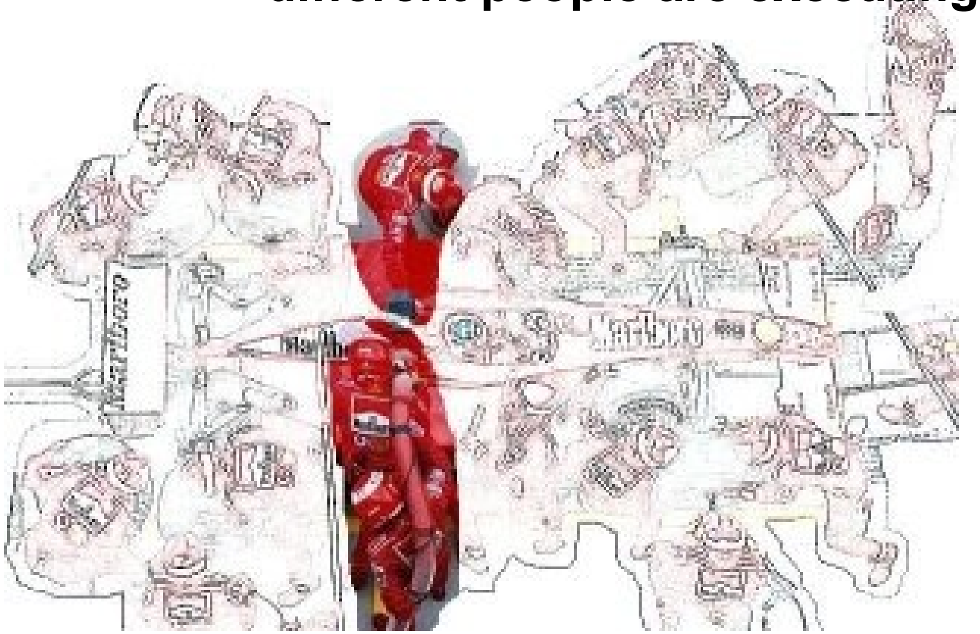
high performance problem example:

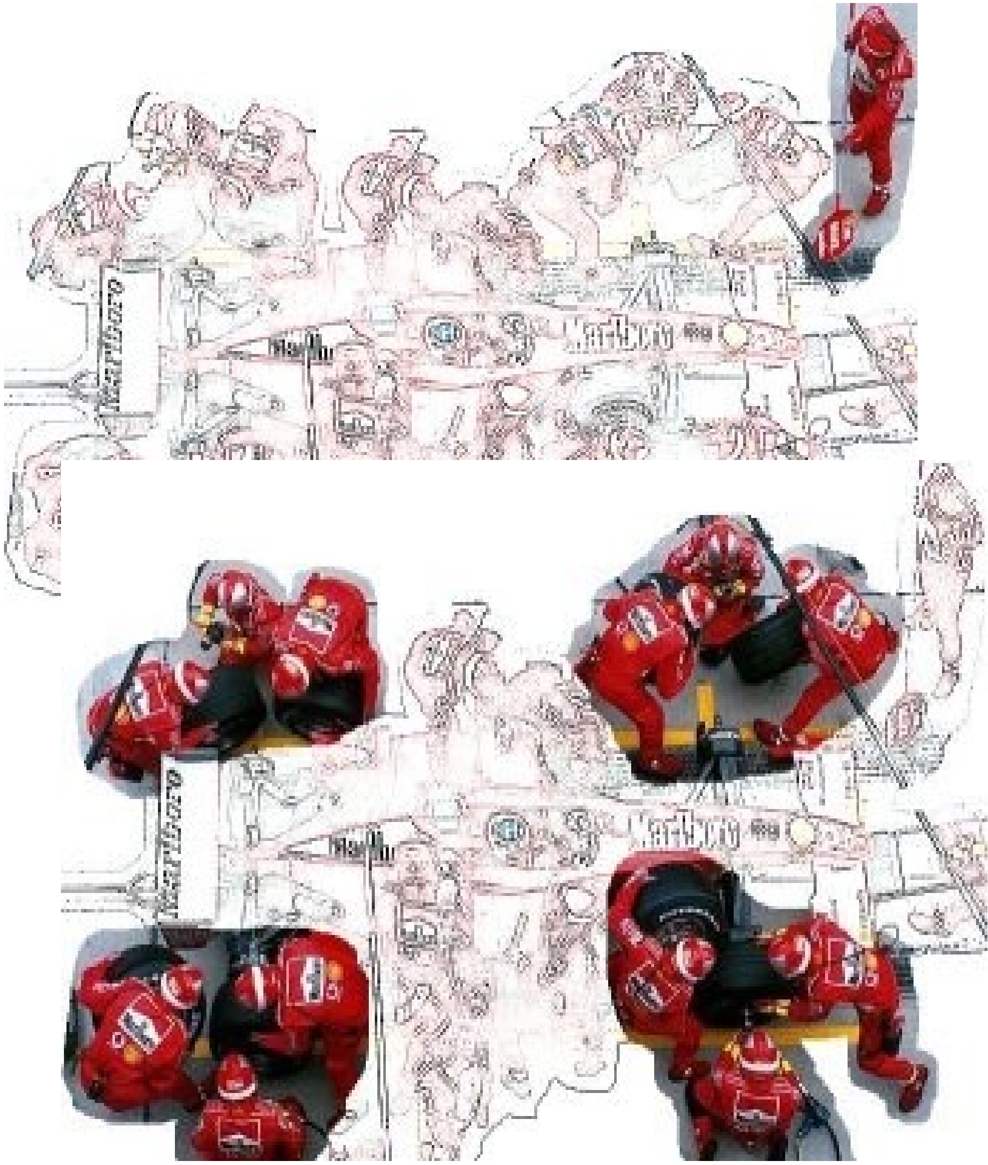
picture from http://www.f1nutter.co.uk/tech/pitstop.php

analysis of the parallel solution:

FUNCTIONAL PARTITIONING

different people are executing different tasks

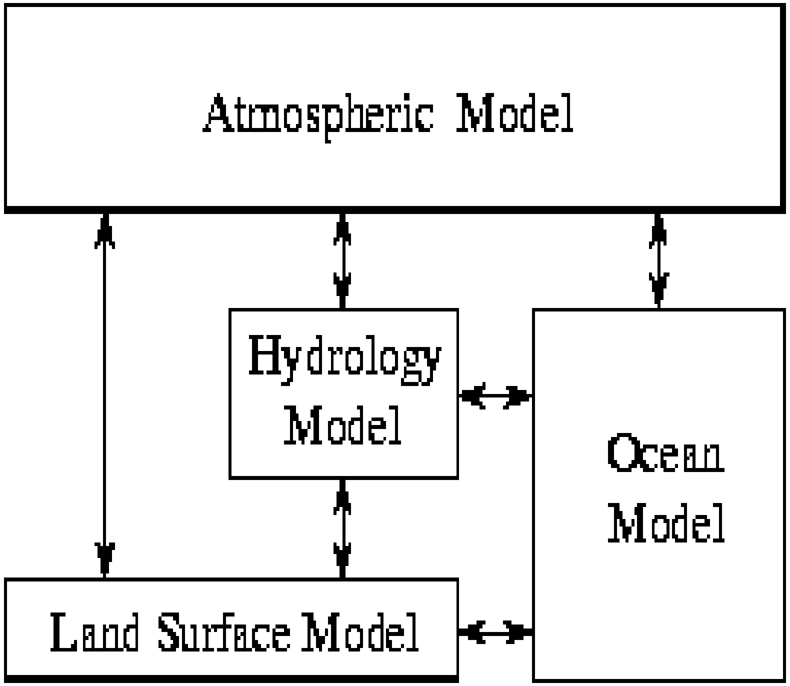


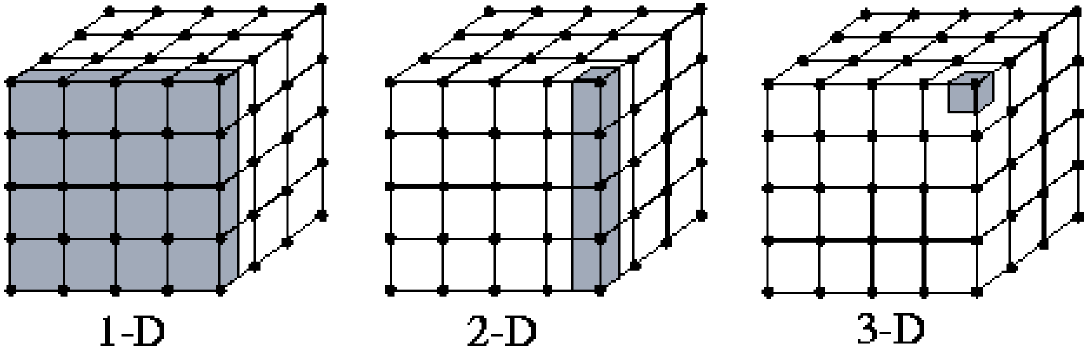


DOMAIN DECOMPOSITION

different people are solving the same global task but on smaller subset

Parallel computing techniques





- DOMAIN DECOMPOSITION

- FUNCTIONAL PARTITIONING

Lab 2 II week activity

EFFICIENT SOLUTION TO THE PROBLEM

picture from the on-line book: http://www-unix.mcs.anl.gov/dbpp/

Units of High Performance Computing:

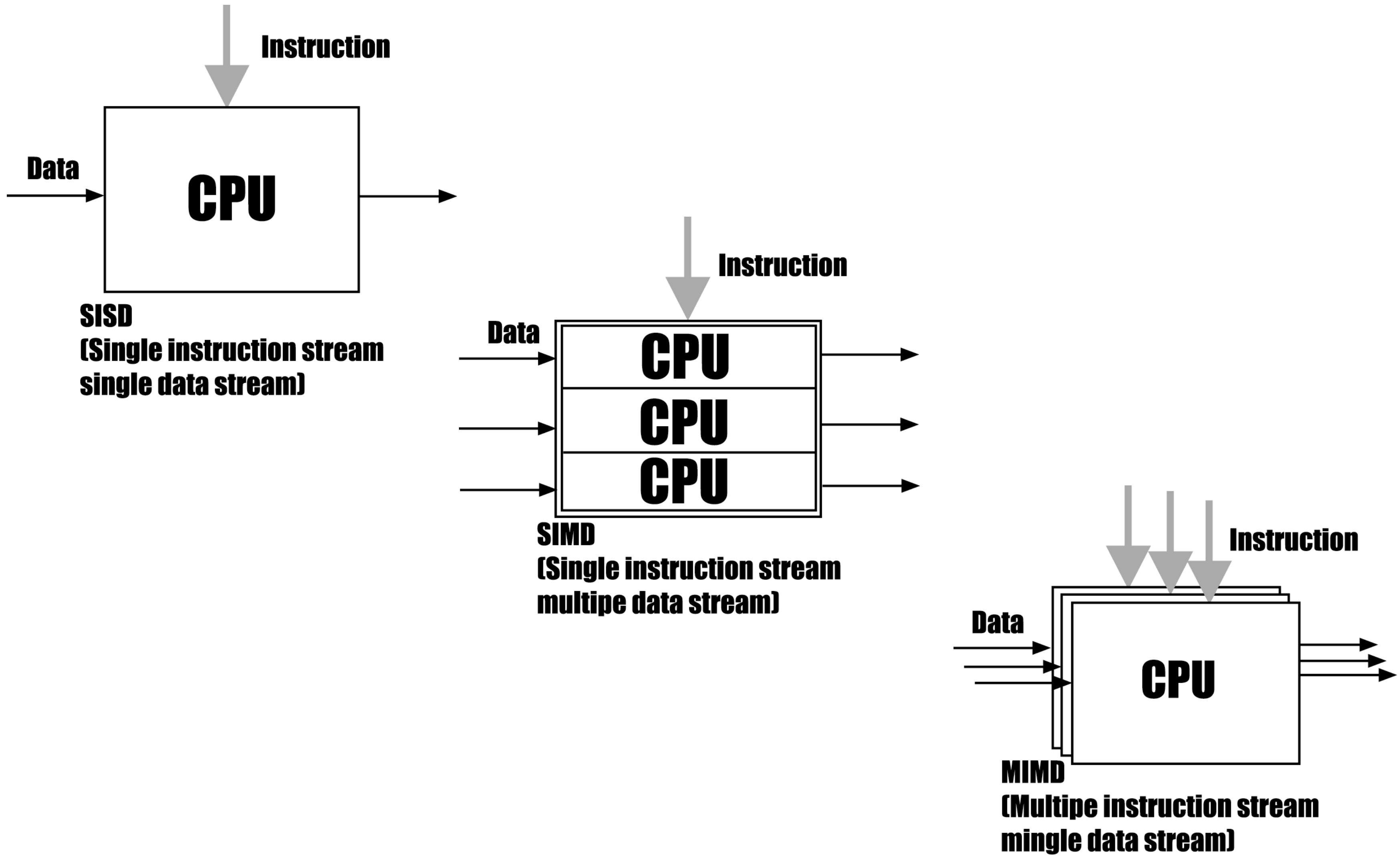
- Computing Data: Floats: floating point operation/ second
  - Mega flops / Gigaflops / Teraflops / Petaflops
- Moving Data: bits : bit /second transmitted
  - among computers: networks
    - 10Mbit/100Mbit/1000Mbit=1Gbit and now also 10Gb
  - within the computer:
    - CPU-Memory: 1 - 10 Gbit
- Storing Data: byte (1byte= 8 bits)
  - kbyte/Mbyte ----> caches/RAM
  - Gigabite -----> RAM/hard disks
  - Terabyte -----> Disks/SAN ...
  - Petabyte ------> SAN

Parallel computers

- Tons of different machines !
- Flynn Taxonomy (1966): helps (?) us in classifying them:
  - Data Stream
  - Instruction Stream

Name	*Instruction stream*	*Data stream*
SISD	Single	Single
SIMD	Single	Multiple
MIMD	Multiple	Multiple
MISD	Multiple	Single

Flynn Taxonomy (graphical view)



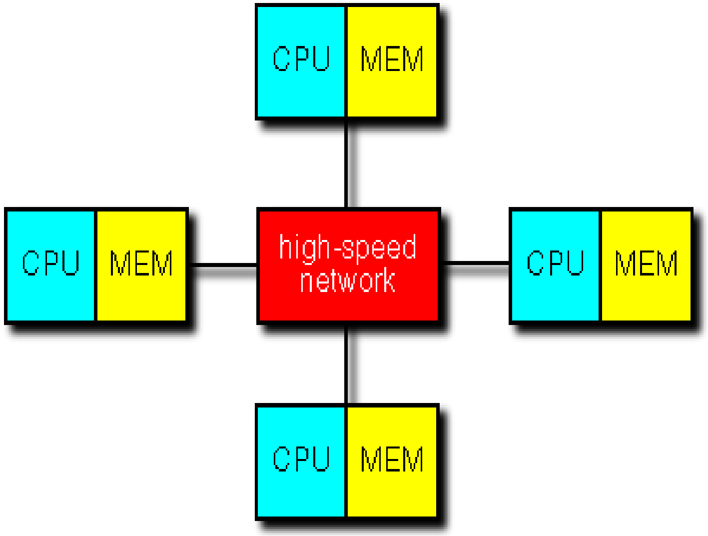
Modern parallel architecture

MEMORY: The simplest and most useful way to classify modern parallel computers is by their memory model:

- SHARED MEMORY
- DISTRIBUTED MEMORY

Shared vs Distributed ?

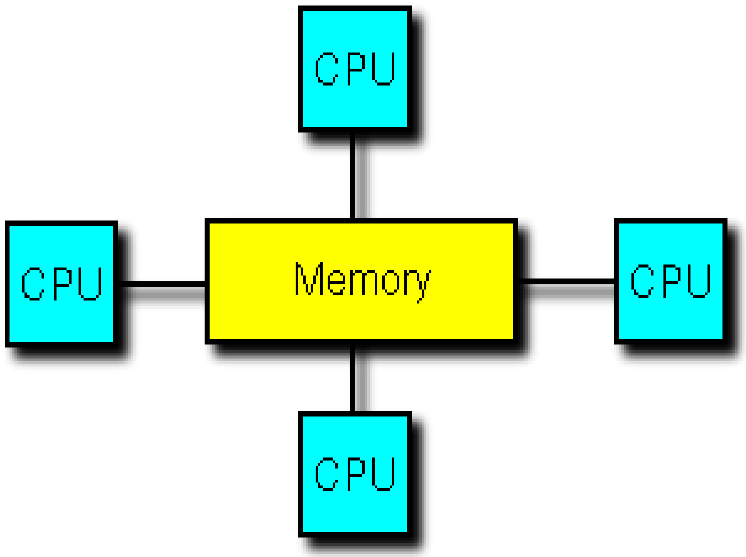
❑Distributed Memory each processor has it's own local memory. Must do message passing to exchange data between processors



❑**multicomputers**

❑Shared Memory

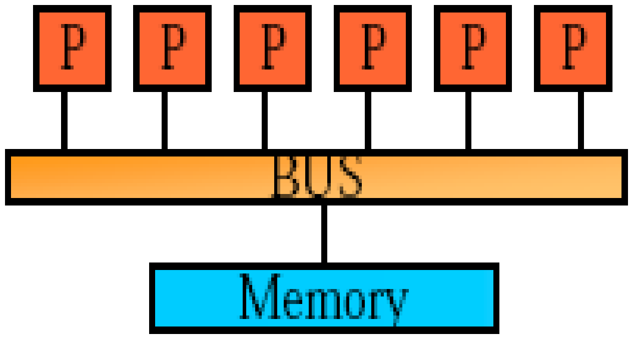
- single address space. All processors have access to a pool of shared memory.



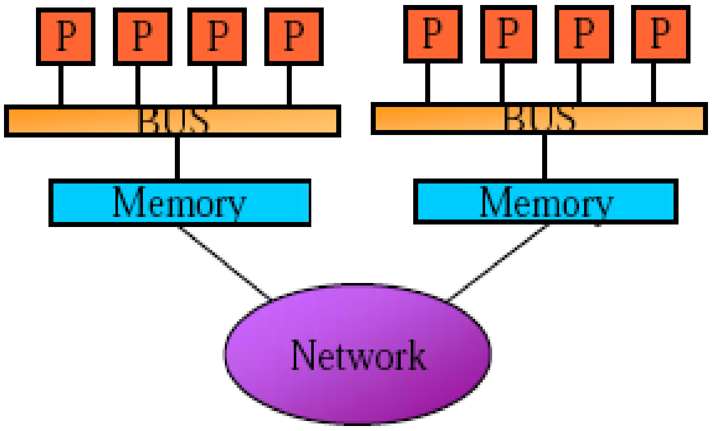
❑**Multiprocessors (MPs)**

Shared Memory: UMA vs NUMA

- *Uniform memory access (UMA):* Each processor has uniform access to memory. Also known as symmetric multiprocessors (SMP)



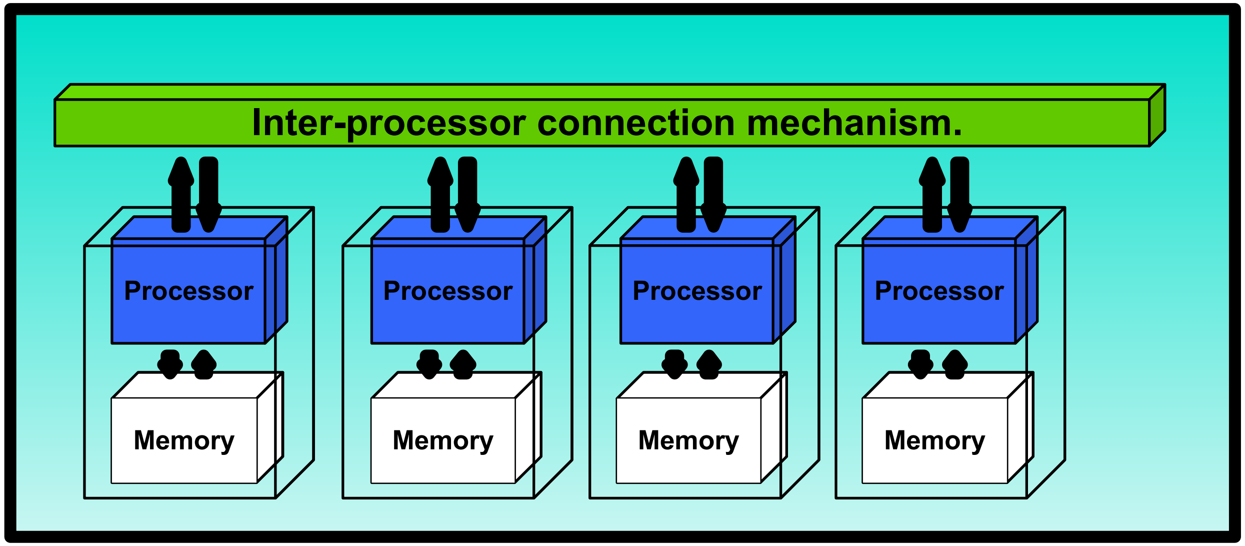
- *Non-uniform memory access (NUMA):* Time for memory access depends on location of data. Local access is faster than non-local access.



Distributed memory architecture: Clusters !

Subject: Re: [Beowulf] about concept of Beowulf clusters
Date: Thu, 24 Feb 2005 19:41:22 -0500 (EST)
From: Donald Becker <becker@scyld.com>

CLUSTER: independent machines combined into a unified system through software and networking



Beowulf Clusters !

Subject: Re: [Beowulf] about concept of beowulf clusters

Date: Thu, 24 Feb 2005 19:41:22 -0500 (EST)

From: Donald Becker <becker@scyld.com>

- The Beowulf definition is commodity machines connected by a private cluster network running an open source software infrastructure for scalable performance computing
- this means:

commodity machines: exclude custom built hardware

connected by a cluster network: These machines are dedicated to being a cluster, at least temporarily.

running an open source infrastructure The core elements of the system are open source and verifiable.

for scalable performance computing The goal is to scale up performance over many dimensions. Ideally a cluster incrementally scales both up and down, rather than being a fixed size.

The Cluster revolution in HPC

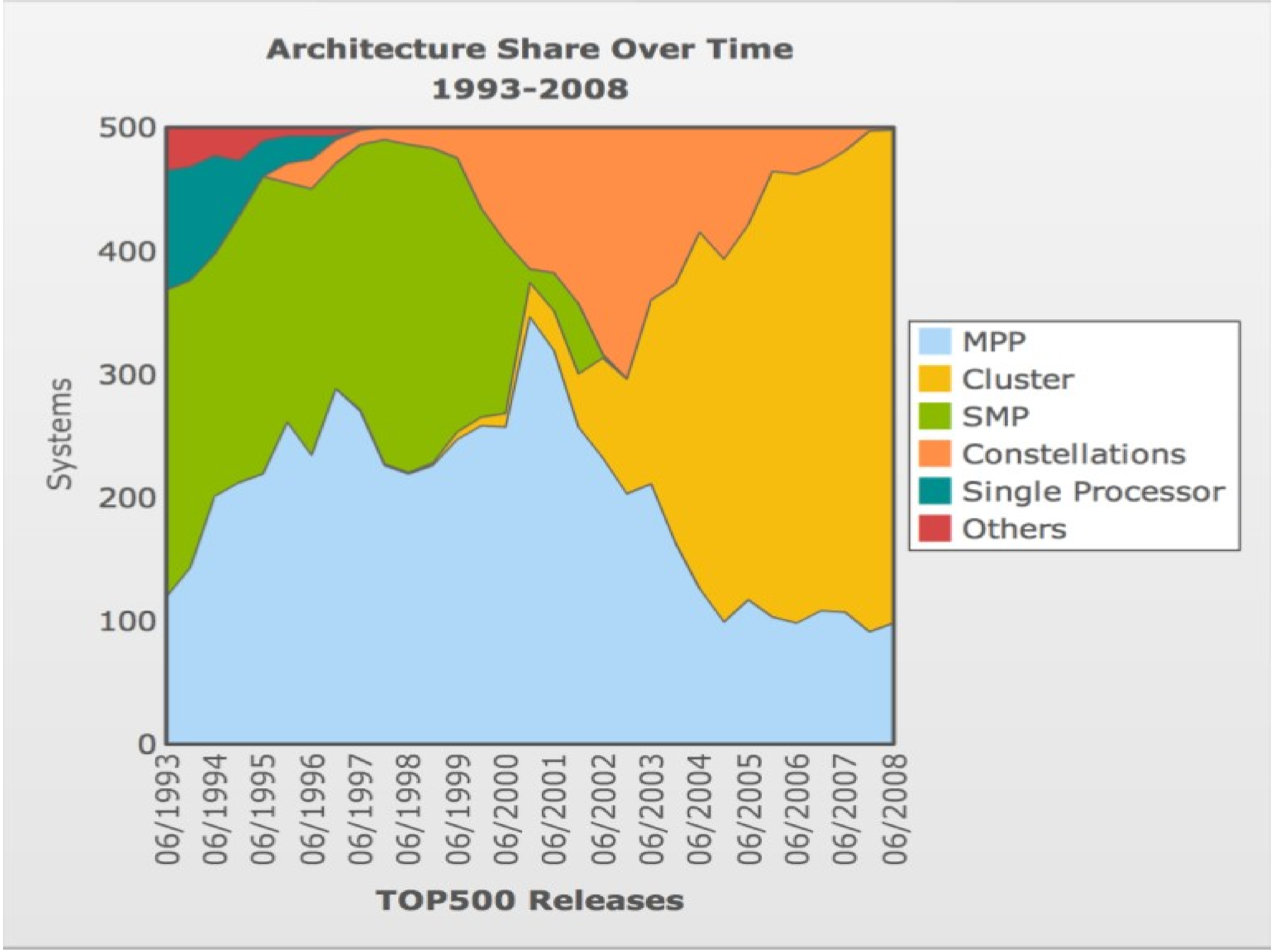
- The adoption of clusters, virtually exploded since the introduction of the first Beowulf cluster in 1994.
- The attraction lies
  - in the (potentially) low cost of both hardware and software
  - the control that builders/users have over their system.
- The problem lies:
  - you should be an expert to build and run efficiently your clusters
  - not always the problem you have fit into a cluster solution (even if this is cheap!)

really a cluster revolution ?

Let us check the Top500 list

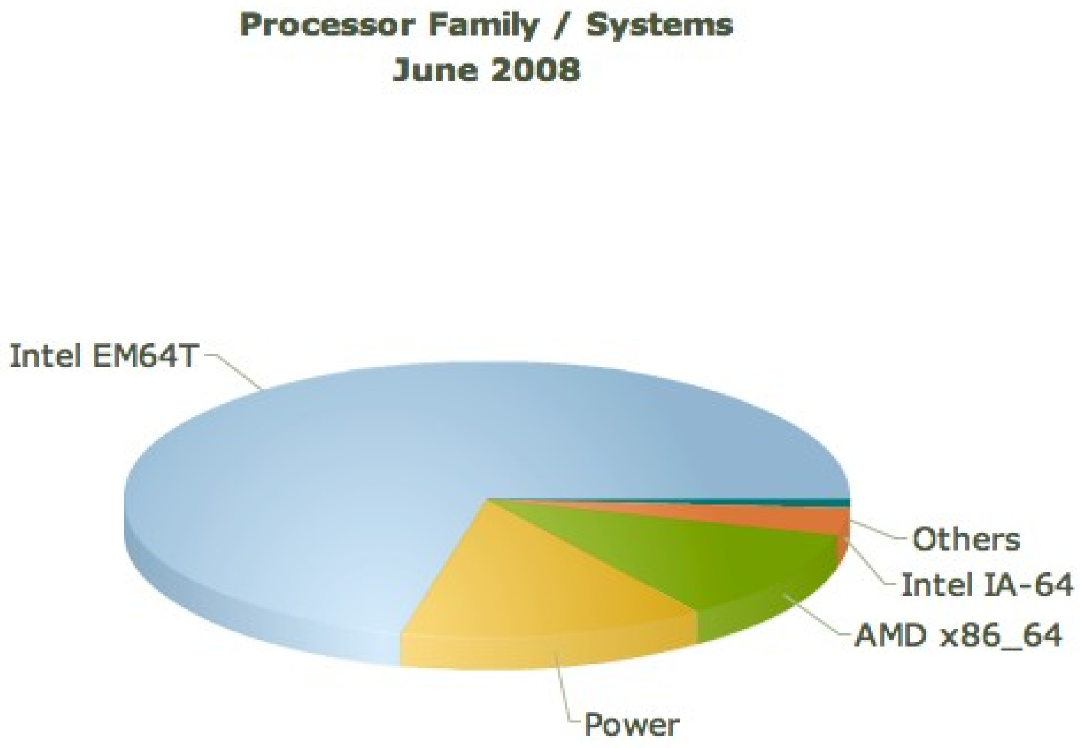
- Listing of the 500 most powerful Computers in the World
- Updated twice a year
  - SC‘xy in the States in November
  - Meeting in Germany in June
- All data available from www.top500.org

architectures over last 15 years



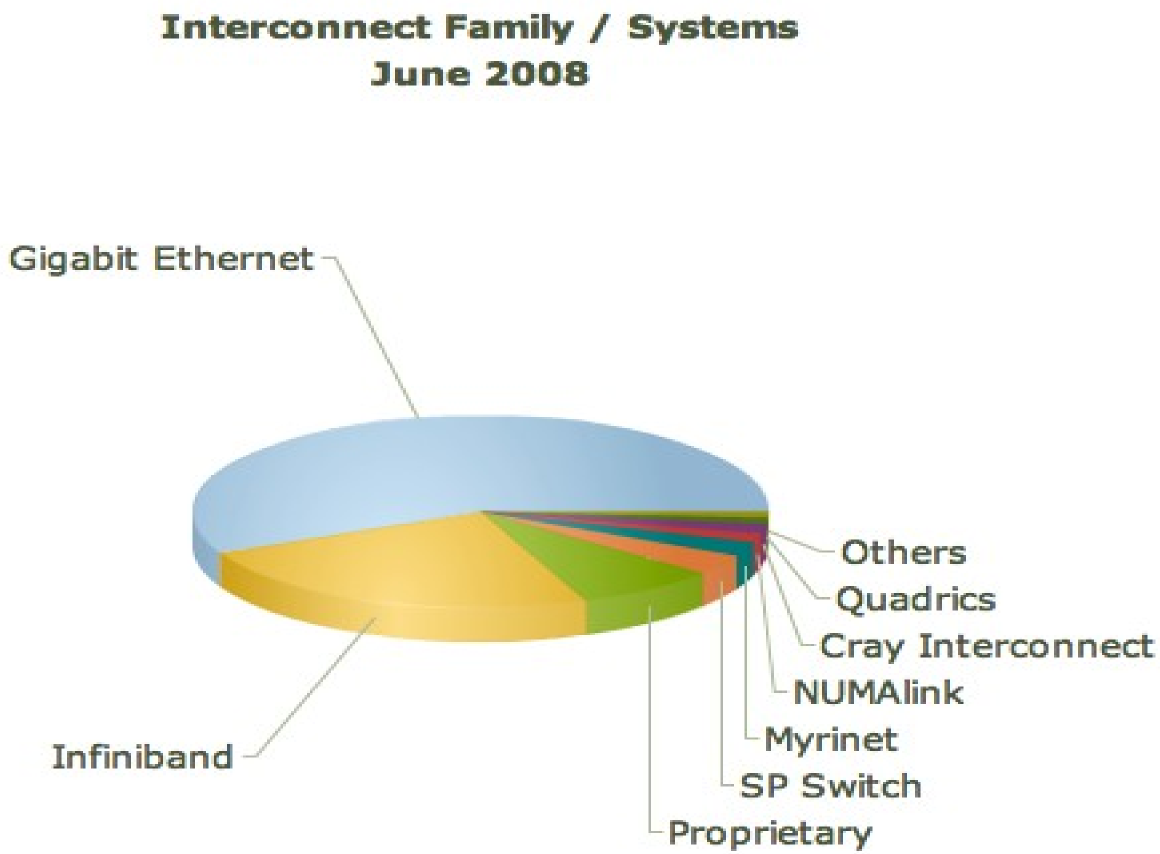
Elements of a Beowulf cluster (1)

The Beowulf definition is commodity machines connected by a private cluster network running an open source software infrastructure for scalable performance computing



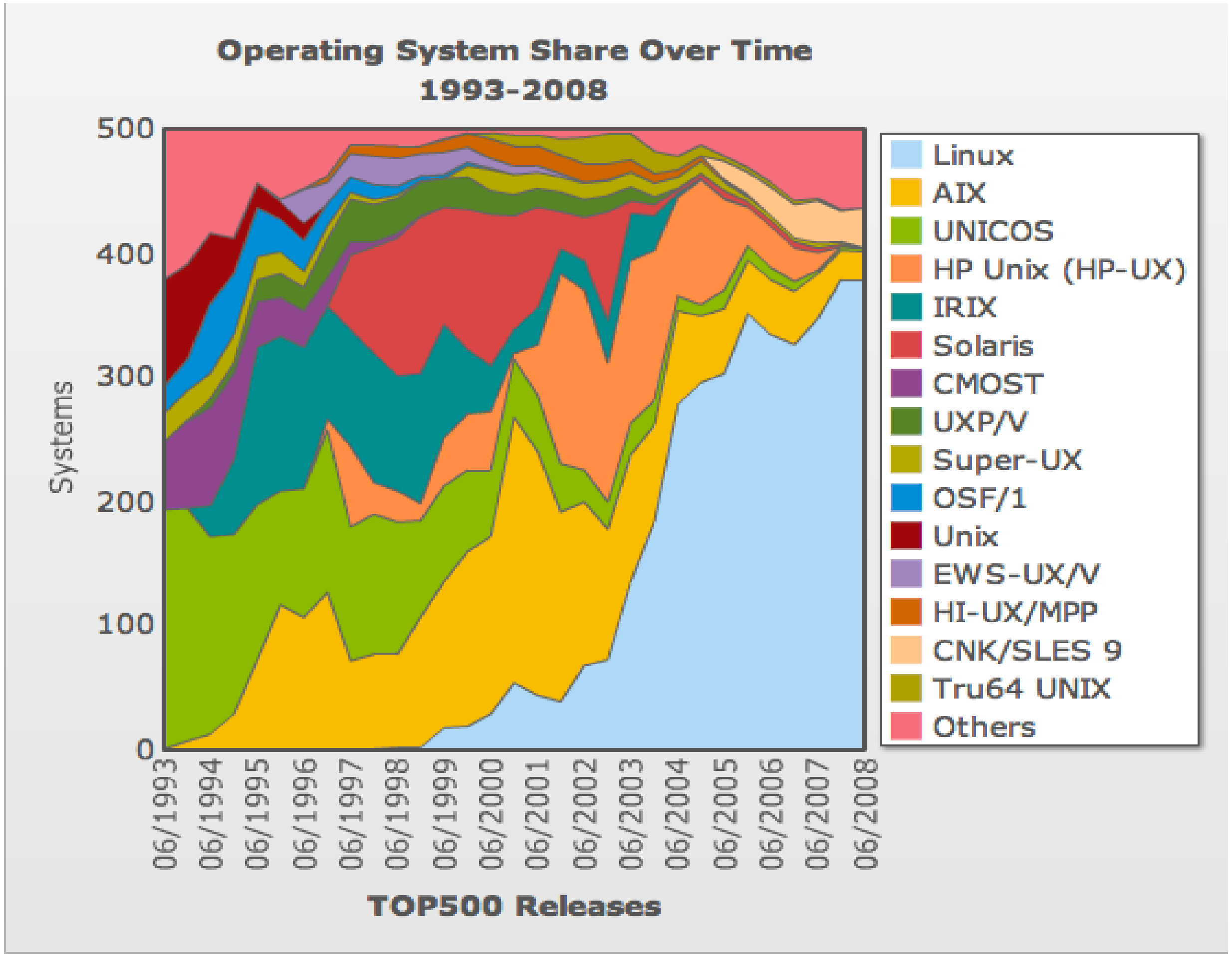
Elements of a Beowulf cluster (2)

The Beowulf definition is commodity machines connected by a private cluster network running an open source software infrastructure for scalable performance computing



Elements of a Beowulf cluster (3)

The Beowulf definition is commodity machines connected by a private cluster network running an open source software infrastructure for scalable performance computing



Why Linux ?

- Access to cheap hardware
- Access to Source code is needed to implement desired features.
- Availability of software
- Access to cheap graduate students
- Access to large community
  - response speed from community sometime much better then vendor/support ones.
- open source/ free software: no license Issues.
- Availability of Scientific Tools/Resources.

Building your own HPC infrastructure

- HPC infrastructure was extremely expensive a few years ago
  - based on supercomputers
- Open source software + commodity off the shelf hardware provides now tools to build low cost HPC infrastructure
  - based on clusters

GREAT CHANCE FOR
LOW BUDGET INSTITUTIONS

Elements of an HPC infrastructure

- Hardware
  - The basic bricks
- Software
  - To make hardware usable
- People
  - installers/sys adm. /planners/ users etc..
- Problems to be solved
  - Any action in building such an infrastructure should be motivated by real needs

Which architectures in your infrastructure ?

- Parallel computing:
  - single systems with many processors working on same problem
- Distributed computing:
  - many systems loosely coupled by a scheduler to work on related problems
- Grid Computing:
  - many systems tightly coupled by software, perhaps geographically distributed, to work together on single problems or on related problems

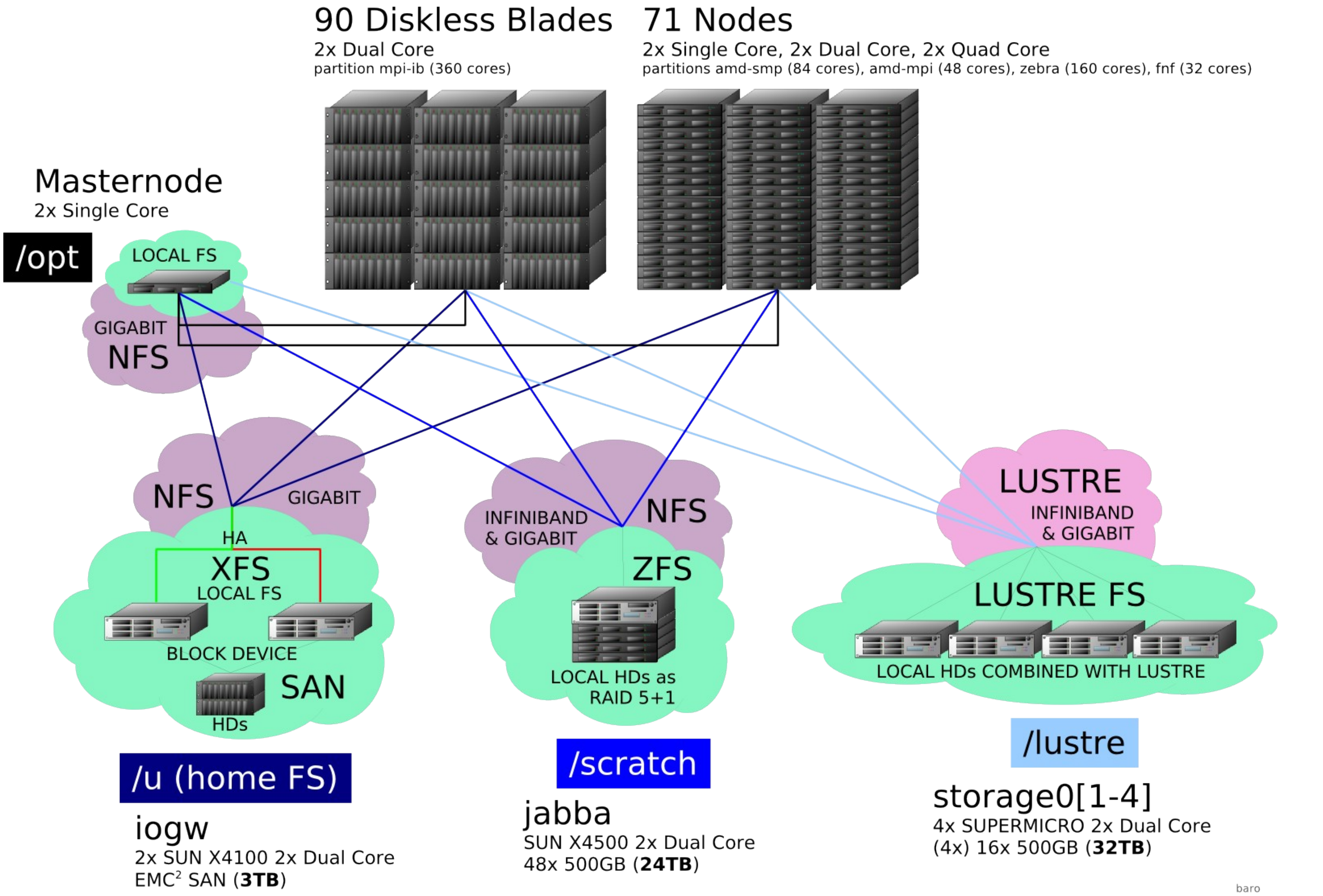
Capability vs Capacity Computing

- Capability computing: the system is employed for one or a few programs for which no alternative is readily available in terms of computational capabilities
  - typical cluster usage
    - small research groups using a few bunch of scientific application

Cluster Installation Lab II week

- Capacity computing: a system is employed to the full by using the most of its available cycles by many, often very demanding, applications and users.
  - typical computer center usage:
    - still clusters can be useful: they required much more work/tuning to fulfill all the requirements

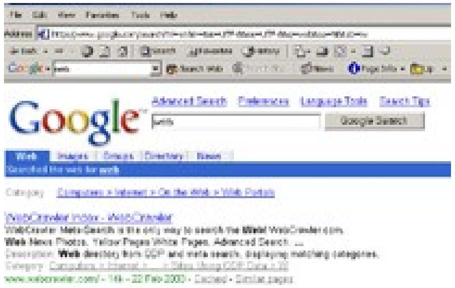
which kind of computing at Sissa/eLab ?



What is Grid ?

• The World Wide Web provides seamless access to information that is stored in many millions of different geographical locations

• In contrast, the Grid is an emerging infrastructure that provides seamless access to computing power and data storage capacity distributed over the globe.



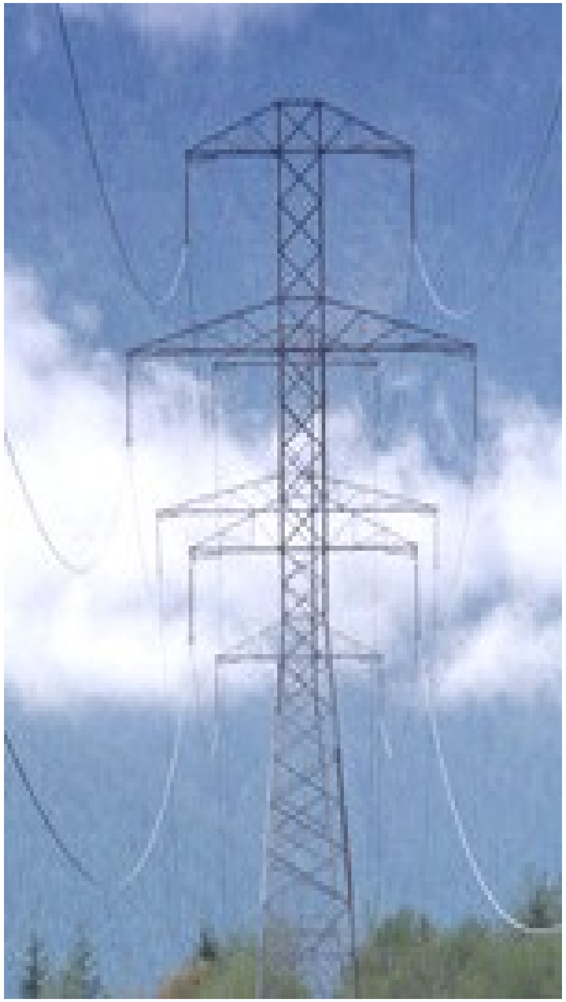
What is Grid ?

• The term Grid was coined by Ian Foster and Carl Kesselman (Grid bible “The Grid: blueprint for a new computing infrastructure”).

• The name Grid is chosen by analogy with the electric power grid: plug-in to computing power without worrying where it comes from, like a toaster.

• The idea has been around under other names for a while (distributed computing, metacomputing, …).

•This time, technology is in place to realise the dream on a global scale.



More on GRID:
Introduction to GRID (day 4)
GRID Lab II week

Grids vs. HPC

- Not an “either/or” question
  - Each addresses different needs
  - Each are part of an integrated solution
- Grid strengths
  - Coupling necessarily distributed resources instruments, software, hardware, archives, and people
  - Eliminating time and space barriers
  - remote resource access and capacity computing
- Grids are not a cheap substitute for capability
- HPC Highest performance computing strengths
  - Supporting foundational computations
  - terascale and petascale “nation scale” problems
  - Engaging tightly coupled computations and teams

- Key is easy access to resources in a transparent way

Wrap-up

- HPC and GRID computing are now fundamental tools for scientific research
- HPC means parallel computing
- HPC experienced a great change in the last ten years: from custom machine to Beowulf clusters
- The challenge is now to build your own HPC infrastructure driven by real needs.
- HPC and GRID computing are not mutually exclusive but can be both used to address computational resources in a transparent way.

Question time