The Abdus Salam
International Centre for Theoretical Physics

United Nations
Educational, Scientific and
Cultural Organization

IAEA
International Atomic Energy Agency

2068-2

Advanced School in High Performance and GRID Computing - Concepts and Applications

30 November - 11 December, 2009

Introduction to HPC/GRID Linux Cluster and all the rest

S. Cozzini

CNR-INFM Democritos
Trieste
Italy

Advanced School in
High Performance
and GRID Computing: concepts
and applications

Introduction to HPC/GRID
Linux Cluster and all the rest

Stefano Cozzini

Democrito and SISSA/eLAB - Trieste

Agenda

- Introduction:
  - or .. “Why do scientists need HPC and GRIDs?
- Section 1: HPC concepts
- Section 2:
  - Parallel computing
  - Parallel machines
- Section 3: GRID concepts
- Conclusions
  - Computer infrastructure for everybody

Why computer simulations in science ?

- measure theories: solve equations which could not be solved otherwise (i.e. get numbers out of theories, much in the same way as experiments get numbers out of natural processes)
- do virtual experiments where experimental conditions can be controlled in ways that would not be possible in the lab
- benchmark the soundness of ideas and theories

three example of computer simulation we have here at Trieste

- Ab-initio simulations of phonon spectra:
  - measure theories/virtual experiment
- Study/predict the path from straight line of amino-acids to folded state of a protein
  - benchmark the soundness of ideas and theories/virtual experiment
- Simulation of climate change for the next 150 years
  - measure theory/virtual experiment

What do we need to perform them ?

- Example 1: *CPU time & Memory* : ~ hundreds of Gigabyte
  - Ex: A simulation using Ab-initio techniques: 256 GB RAM
    - I need to perform tens of them..
- Example 2 : *CPU time and a lot of PCs*
  - Ex: A single MD Simulation of proteins on my PC ~60 days
  - I need several hundreds...
- Example 3: *CPU time and storage*
  - Ex: A single run producs ~20 Terabyte of data

Ad hoc , fast, powerful,reliable computational platform are needed: in short HPC !

Agenda

- Introduction:
  - or .. “Why do scientists need HPC and GRIDs?
- Section 1: HPC concepts
- Section 2:
  - Parallel computing
  - Parallel machines
- Section 3: GRID concepts
- Conclusions
  - Computer infrastructure for everybody

HPC stands for:

- High Performance Computing
- The term is most commonly associated with computing used for scientific research. [from wikipedia]
- it is not only on hardware but involves software and people as well
- A possible definition:

High Performance Computing encompasses a collection of powerful:

hardware systems

software tools

programming languages

parallel programming paradigms

which make previously unfeasible calculations possible.

Only performance ?

- Performance is not always what is matter..

- From wikipedia:

  To reflect a greater focus on the productivity, rather than just the performance, of large-scale computing systems, many believe that HPC should now stand for High Productivity Computing.

Performance vs Productivity

- A definition:
  - Productivity = (application performance) / (application programming effort)
- scientists in HPC arena have different goals in mind thus different expectations and different definitions of productivity.

Which kind of productivity are you interested in ?

Please describe your productivity concept in your blog on the moodle platform..

My blog entry

Defining Productivity..
by Stefano Cozzini - Sunday, 29 November 2009, 11:17 am

Yourself (draft)

I try to give here a short view of my productivity concept I keep in mind when I perform my duties:

as maintainer of HPC facility:

productivity means to keep as much as possible computational nodes up and running and busy.
We are not concerned about the speed of computational nodes but about reliability: we recently switch off 2.4 Ghz cpus opteron machines and replaced by of 2.2Ghz dual core processors much more stable.
This way we decrease overall performance but we increase overall productivity.

as scientific support:

productivity means scientific productivity.
As an example: it is better to help users to move their calculation from a quite powerful saml HPC platform to a less powerful architecture but larger enough to deal with a larger amount of calculation.

Edit | Delete | Permalink

Why is HPC relevant ?

- We continually demand greater computational power
- We want to reduce the execution time of our important applications
- We want to overcome the limitations of desktop computing architectures
- HPC-capable architectures are becoming more ubiquitous, user-friendly and affordable

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
  - Multiple computers to solve a particular task

NOTE: this three ways can be played simultaneously !

performance in computers and associates units

- How fast can I crunch numbers on my CPUs ?
- How fast can I move data around ?
  - from CPUs to memory
  - from CPUs to disk
  - from CPUs on different machines
- How much data can I store ?

CPU crunching: what do we count ?

- Rate of million of floating point operations per second (Mflops)
  - Compare to theoretical or sustained Mflops peak performance
- Theoretical peak performance:

  determined by counting the number of floating-point additions and multiplications that can be completed during a period of time, usually the cycle time of the machine

  IPC=Instruction per Cycle

Peak performance of modern systems

- my laptop: ~ 4 Gigaflops
- Addishpc.aau.edu.et cluster
  - 4 IPC x2.27 Ghz x 80 cores (20 Intel 5520) ~ 720Gigaflops
- Hg1.hpc.sissa.it cluster (intel partition)
  - 4 IPC x 2.5Ghz x 200 cores (50 Intel 5420) ~ 2 Teraflops
- Sp6.cineca.it:
    - ~ 5300 power6 processors 4.7Ghz = ~100 Teraflops
- ORNL jaguar
    - 224162 cores (Opteron 6 core) ~2331 Teraflops

sustained performance

- What your application is actually able to do on the system..
- Last century figures:

Table 1: Benchmark Results for NERSC's 644-PE Cray T3E

Benchmark	System Performance	Single Processor Performance	% of Peak
Theoretical peak	580.0 Gflop/s	900 Mflop/s	100.0%
Linpack	444.2 Gflop/s	690 Mflop/s	76.6%
LSMS code (Locally Self-consistent Multiple Scattering), 1998 Gordon Bell Prize-winning application	256.0 Gflop/s	398 Mflop/s	44.1%
Average of seven major NERSC applications	67.0 Gflop/s	~104 Mflop/s	11.6%
NAS Parallel Benchmarks	29.6 Gflop/s	~46 Mflop/s	5.1%

Currenent sustained performance (2009)

Taken from: www.nccs.gov/wp-content/training/2009_crayxt.../JohnLevesque1.pdf

CRAY

Two Months and potential break-through Science for 8 research groups

All of these runs set new World Records in Performance

Science Area	Code	Contact	Cores	% of Peak	Total Perf	Notes	Scaling
Materials	DCA++	Schulthess	150144	97%	1.35 PF	2008 Gordon Bell Finalist	Weak
Materials	LSMS/WL	ORNL	149580	76.40%	1.05 PF	64 bit	Weak
Seismology	SPECFEM3D	UCSD	149784	12.60%	164 TF	2008 Gordon Bell Finalist	Weak
Weather	WRF	Michalakes	150000	3.60%	50 TF	2007 Gordon Bell Finalist	Strong
Climate	POP	Jones	21000		20 sim yrs/ CPU dau	Size of Data	Strong
Combustion	S3D	Chen	144000	6.00%	83 TF		Weak
Fusion	GTC	PPPL	102000		20 billion particles pushed	Code Limit	Weak
Materials	LS3DF	Lin-Wang Wang	147456	32%	425 TF	2008 Gordon Bell Finalist	Weak

* Mixed Precision – 626 TF at 128K cores in 64 bit only

Moving data around: bits and/or Mb/sec

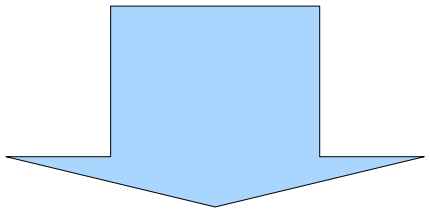
- bit/second transmitted
- among computers: networks
  - default (commodity)
    - 1000Mbit=1Gbit
  - custom(high speed)
    - 10Gb and now 40Gb
- within the computer:
  - CPU-Memory: thousand of Mb/sec (GByte/sec)
    - 10 - 100 Gbit
  - CPU- Disks : MByte/sec
    - 50 ~ 100 MB up 1000MB/sec

Storage size: byte

- size of storage devices:
  - kbyte/Mbyte ----> caches/RAM
  - Gigabyte -----> RAM/hard disks
  - Terabyte -----> Disks/SAN
  - Petabyte ------> SAN / Tapes devices

HPC architecture

HPC architectures try to maximize performance simultaneously on all the three aspects (number crunching/ data access /data storage) by using many Processing Elements (CPUs) together to solve a given task.



PARALLEL COMPUTING !

Agenda

- Introduction:
  - or .. “Why do scientists need HPC and GRIDs?
- Section 1: HPC concepts
- Section 2:
  - Parallel computing
  - Parallel machines
- Section 3: GRID concepts
- Conclusions
  - Computer infrastructure for everybody

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]

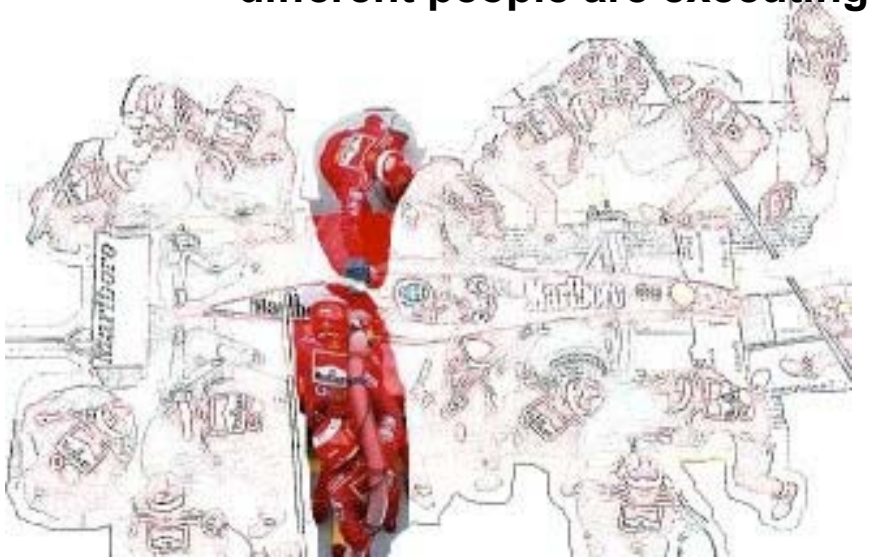
high performance problem example:

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

HPC parallel computers

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

- How CPUs view the available memory ?

  - SHARED MEMORY
  - DISTRIBUTED MEMORY

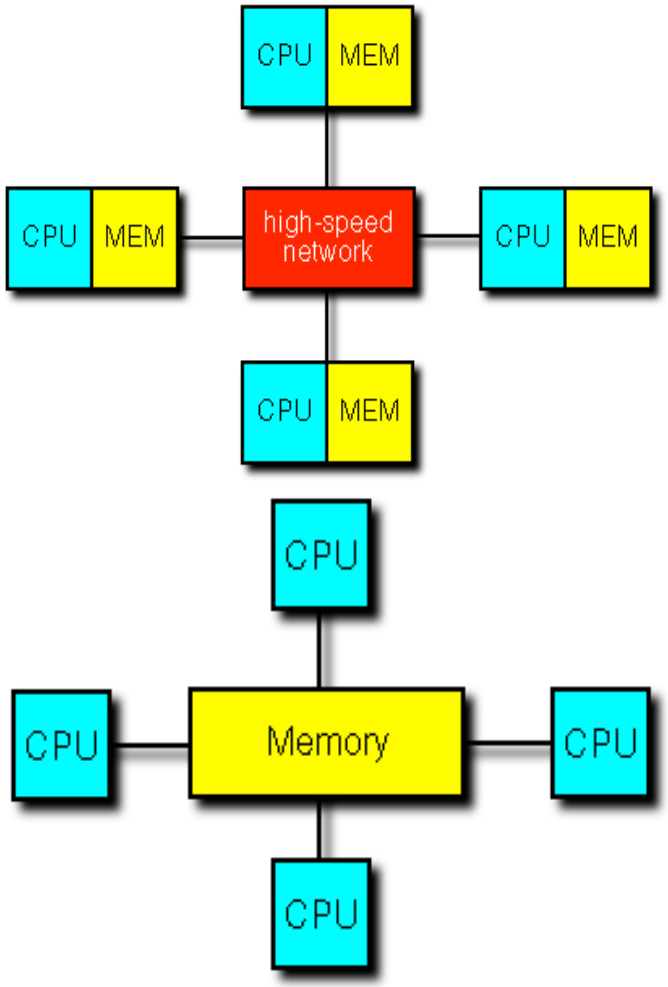
Shared vs Distributed ?

- Distributed Memory each processor has its own local memory. Must do message passing to exchange data between processors

- **multicomputers**

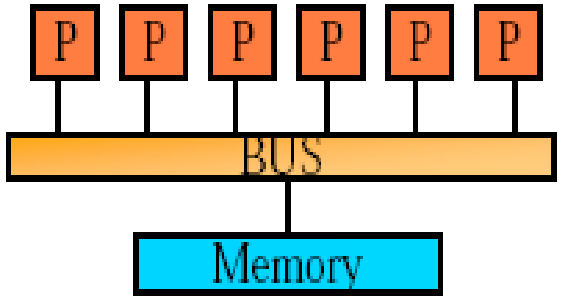
- Shared Memory
  - single address space. All processors have access to a pool of shared memory.

- **Multiprocessors (MPs)**

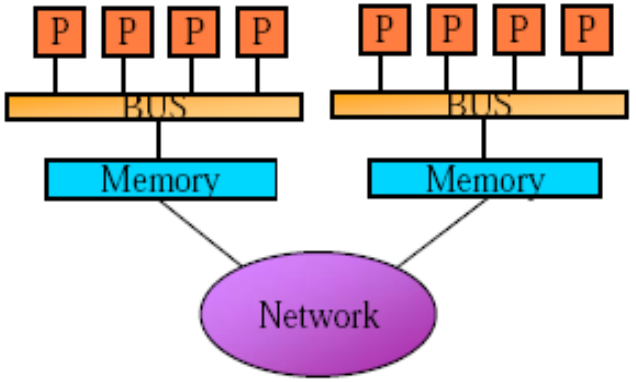


Shared Memory: UMA vs NUMA

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

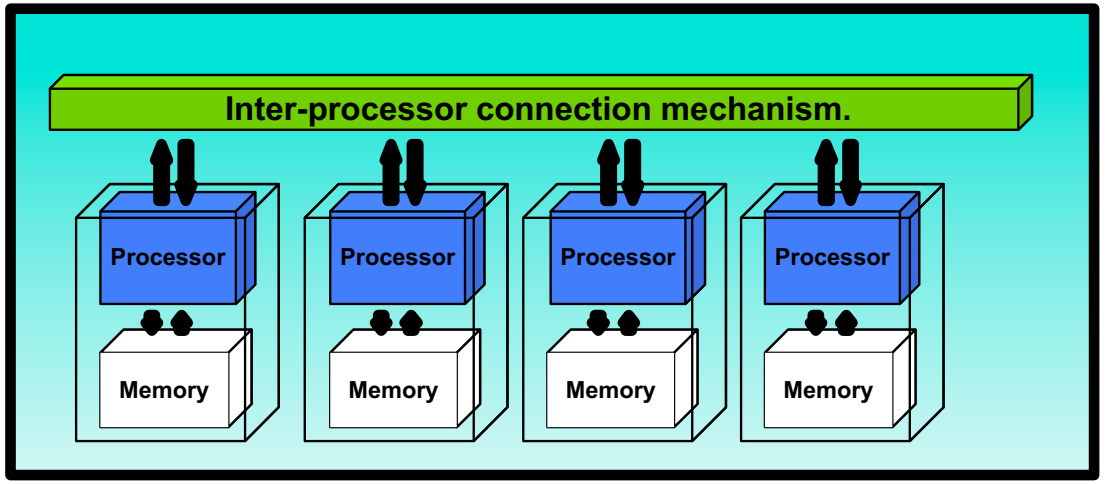


- *Non-uniform memory access (NUMA):* Time for memory access depends on location of data. Local access is faster than non-local access.

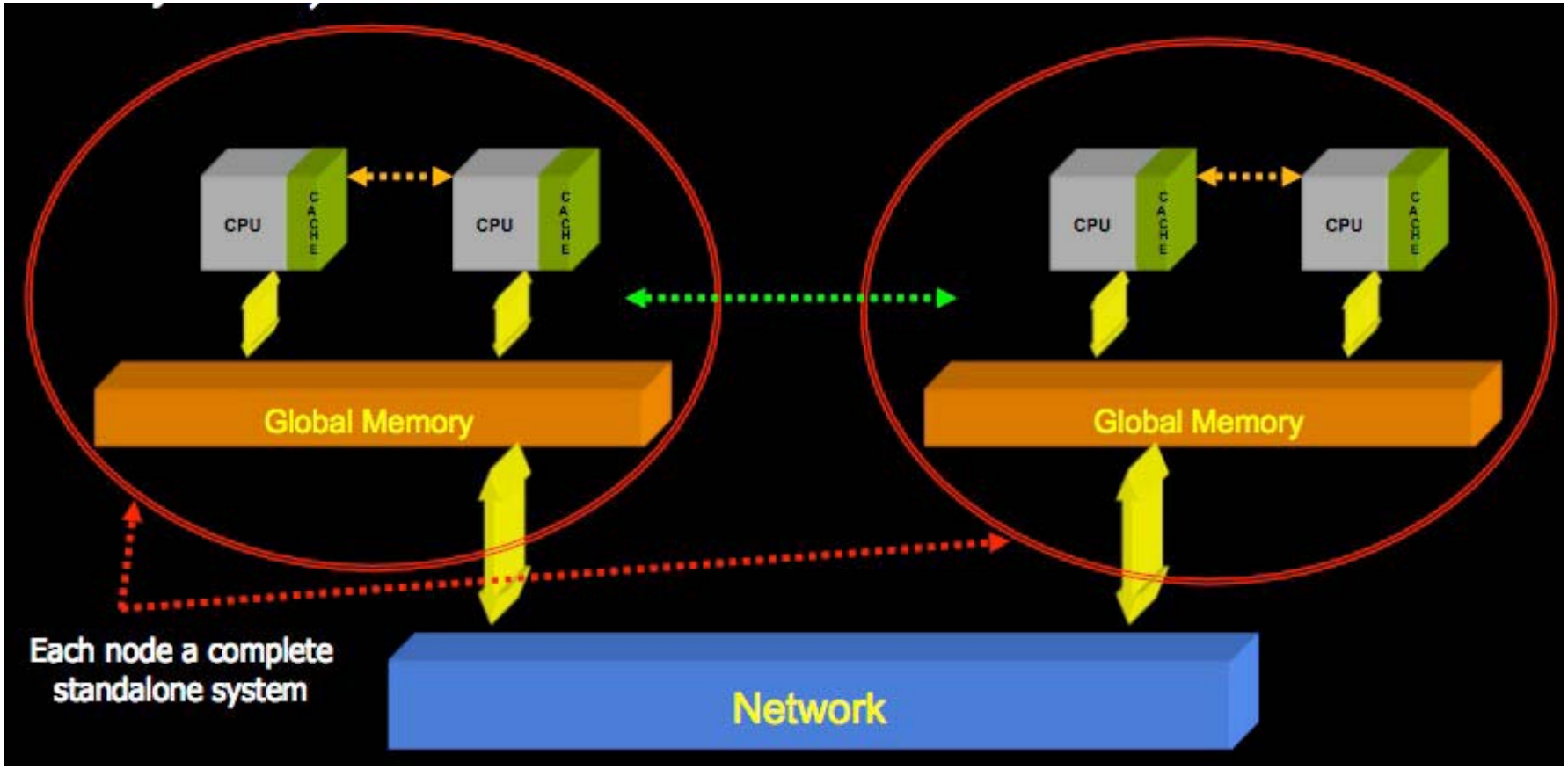


Distributed memory architecture: Clusters !

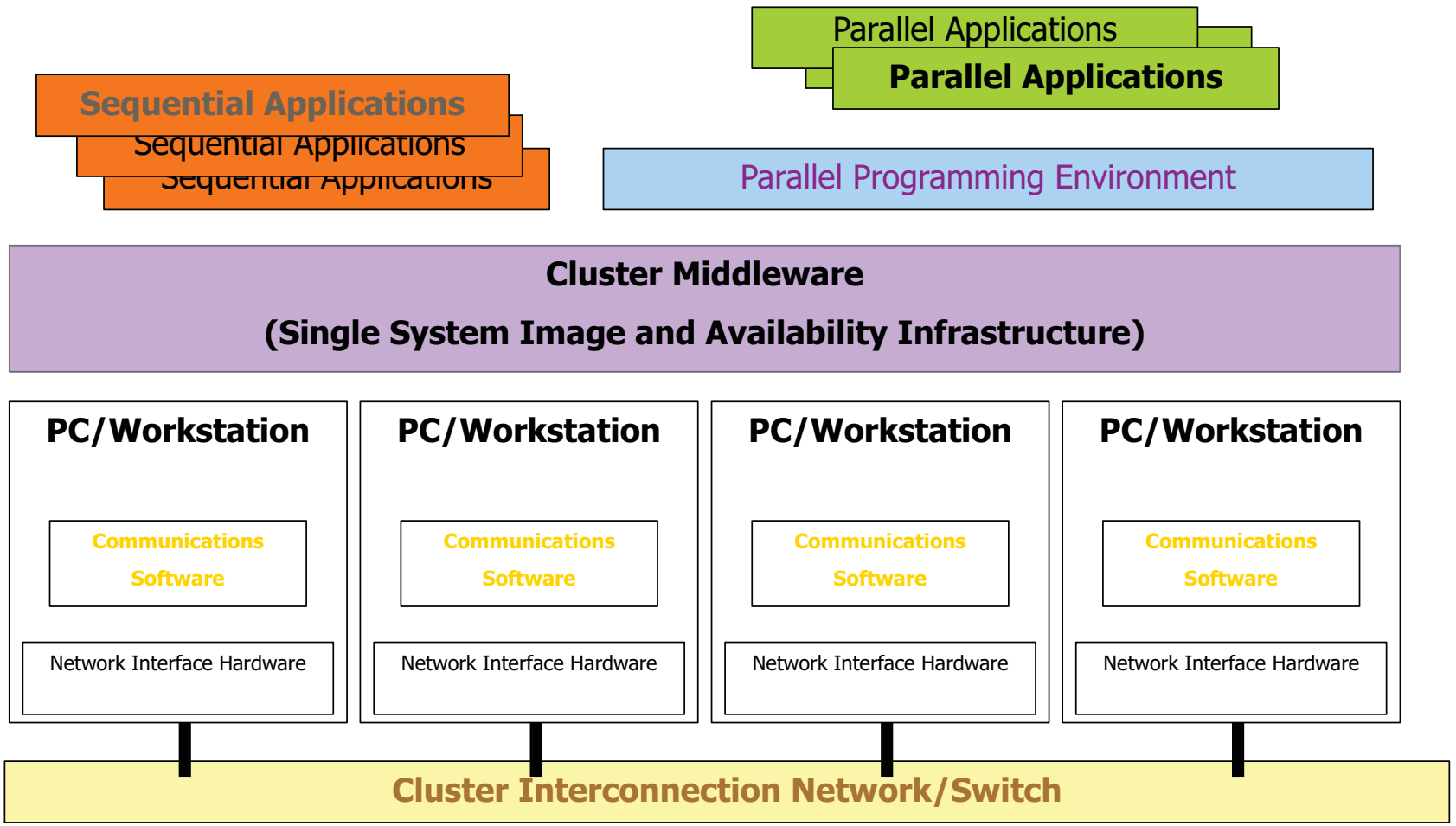
CLUSTER: independent machines combined into a unified system through software and networking



Modern clusters are hybrid architectures !



Cluster Computer Architecture

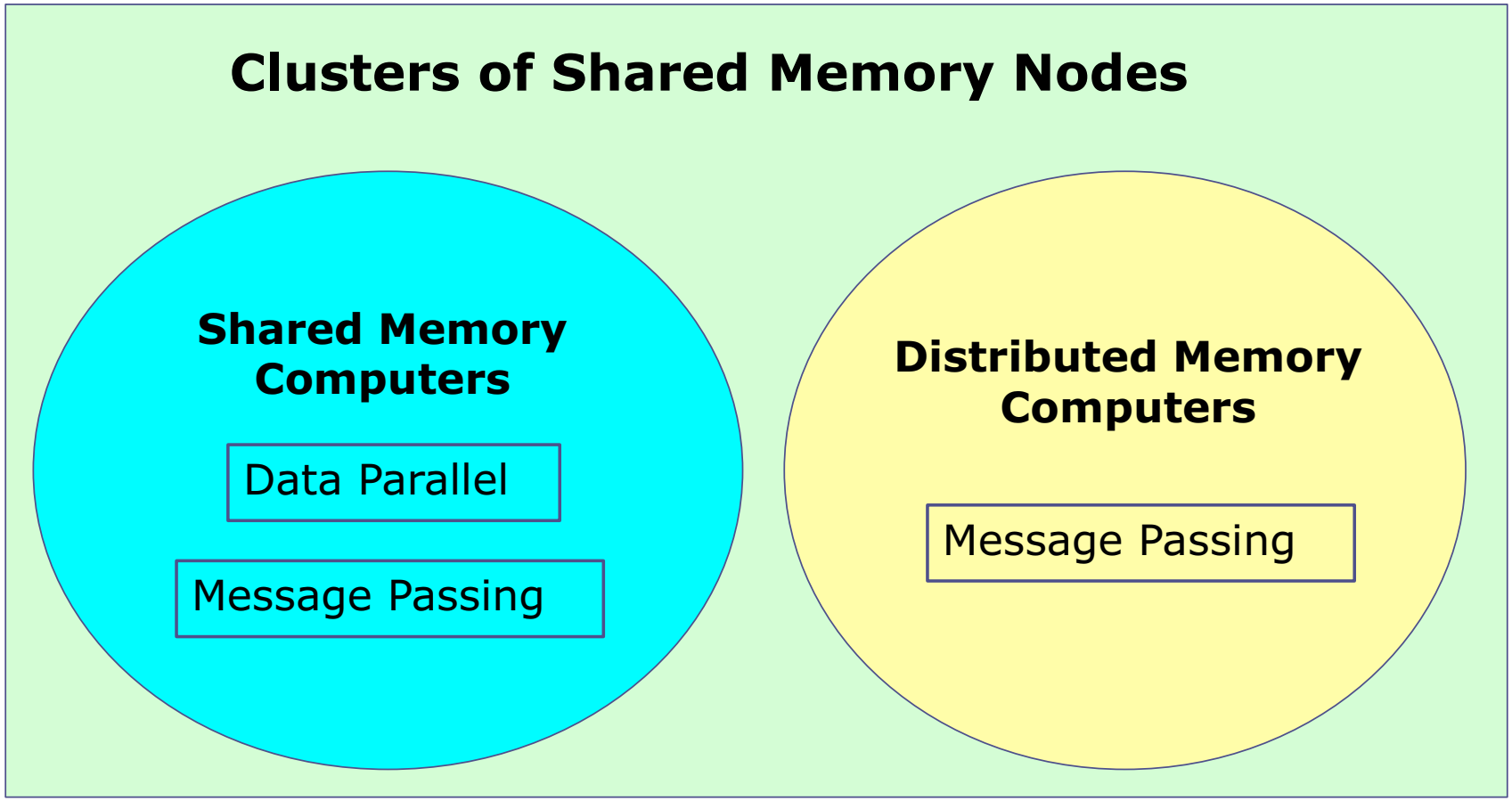


Parallel Programming Paradigms

The two architectures determine two basic schemes for parallel programming

Parallel machines	
Distributed memory	**Shared Memory**
Parallel paradigms	
Message Passing	Data parallel
All processes could **directly** access only their local memory,. Explicit messages are requested to access remote memory of different processors	Single memory view, all processes (usually threads) could **directly** access the whole memory

Architectures vs. Programming Paradigms



Agenda

- Introduction:
  - or .. “Why do scientists need HPC and GRIDs?
- Section 1: HPC concepts
- Section 2:
  - Parallel computing
  - Parallel machines
- Section 3: GRID concepts
- Conclusions
  - Computer infrastructure for everybody

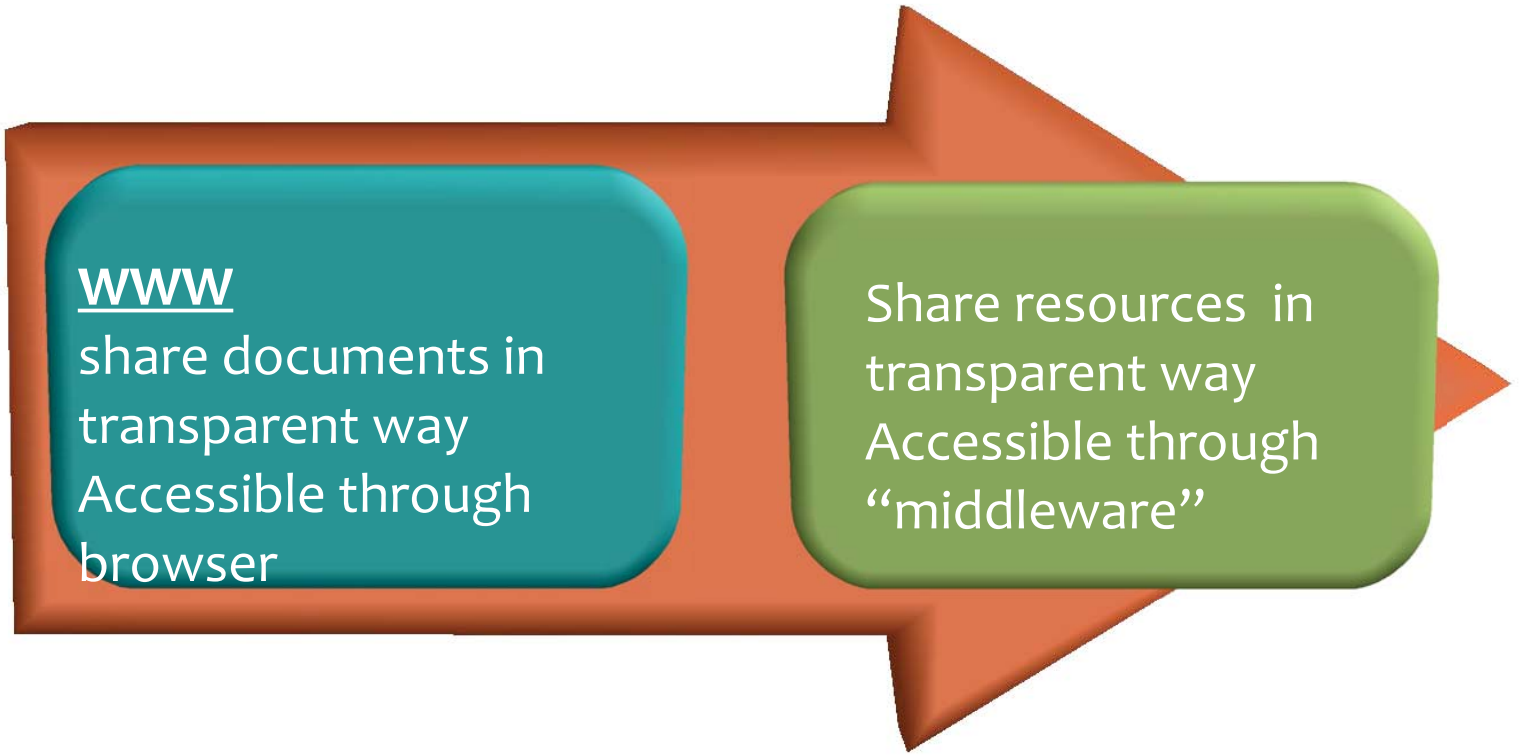
Why the GRID?

- Motivation: When communication is close to free we should not be restricted to local resources when solving problems.
- A Grid Infrastructure built on the Internet and the Web to enable and exploit large scale sharing of resources
- It should provides Scalable Secure Reliable mechanisms for discovery and for remote access of resources.

Using internet to make science...

- On-line publication paper/pre-prints (eg. babbage.sissa.it)
- CPU cycle scavenging (eg. Seti@home, Condor)
- Sloan Digital Sky Survey: online database of astronomical data http://www.sdss.org/
- and many others...

A new paradigm



Resource sharing..

- Applications: web services technology
- CPU and Storage: Grid computing, Cloud Computing, etc.
- Data: Data Grid, Virtual Observatory, Google Filesystem, etc.
- Instruments: Virtual Labs, collaboration tools, etc.

a few definitions for grid computing

- “a single seamless computational environment in which cycles, communication, and data are shared, and in which the workstation across the continent is no less than one down the hall"
- “wide-area environment that transparently consists of workstations, personal computers, graphic rendering engines, supercomputers and non-traditional devices: e.g., TVs, toasters, etc."
- “[**framework** for] flexible, secure, coordinated resource sharing among dynamic collections of individuals, institutions, and resources"
- “collection of geographically separated resources (people, computers, instruments, databases) connected by a high speed network [...distinguished by...] a software layer, often called middleware, which transforms a collection of independent resources into a single, coherent, virtual machine”

Agenda

- Introduction:
  - or .. “Why do scientists need HPC and GRIDs?
- Section 1: HPC concepts
- Section 2:
  - Parallel computing
  - Parallel machines
- Section 3: GRID concepts
- Conclusions
  - Computer infrastructure for everybody

Building your own computational infrastructure

- Open source software + commodity off the shelf hardware provides now tools to build low cost HPC infrastructure
  - based on clusters
- GRID infrastructures are just two clicks away
  - they can provide a looot of resources

Which computational infrastructure do you want ?

Elements of a computational 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 paradigm/infrastructure for your problem ?

- HPC infrastructure:
  - Hpc systems + high performance network to link them together
- Grid Computing infrastructures :
  - many systems tightly coupled by software, perhaps geographically distributed, to work together on single problems or on related problems

Not an “either/or” question
- Each addresses different needs
- Each are part of an integrated solution

Which HPC/GRID infrastructure do I need ?

- Which applications ?
  - Parallel
    - Tightly coupled
    - Loosely coupled
    - Embarrassingly
  - Serial
    - Memory / I/O requirements
- Which user's community ?
  - Large /Small
  - Distributed or not ?
  - Homogeneous /heterogeneous
- Budget considerations

Wrap-up

- Modern scientific research need lots of computational resources provide by HPC/GRID infrastructures
- HPC means parallel computing
- GRID means pooling of geographically distributed resources
- HPC and GRID computing are not mutually exclusive but can be both used to address computational resources in a transparent way.
- The challenge is now to build your own computational infrastructure driven by real needs.

Conclusions

With access to commodity HPC hardware and a free OS such as Linux, entry-level HPC is within reach of even the most modest budget.

It is relatively easy to join large grid infrastructures that make available large amount of computational power

However...

To fully exploit HPC/GRID one needs to obtain detailed knowledge of HPC/GRID architectures as well as master HPC/GRID development tools and advanced programming techniques

We hope to give some insight about these in this two weeks school .