Which infrastructure for my computation ?

Stefano Cozzini

Democrito and SISSA/eLAB - Trieste

Agenda

- Introduction:
  - E-infrastructure and computing infrastructures
  - What is available on the market ?
- Every Thing You Always Wanted to Know About ~~Sex~~ Computing infrastructure But Were Afraid to Ask

What is e-infrastructure ?

- E-infrastructure includes:
  - Networks (internet, light paths…)
  - Computers (workstations, servers, HPC…)
  - Access controls (security, AAA…)
  - Middleware (metadata…)
  - Finding tools (portals, search engines…)
  - Digital libraries (bibliographic, text, images, sound…)
  - Research data (national and scientific databases, individual data…)

What is a computing infrastructure?

infrastructure for computational science

- powerful and modern clusters of multicores.
  - hardware
  - software

High Performance Computing

- pooling of resources geographically distributed
- distribute collaborations

GRID COMPUTING

- Infrastructure as a Service

CLOUD COMPUTING

HPC stands for:

- High Performance Computing
- The term is most commonly associated with computing used for scientific research.

[wikipedia]

- it is not only on hardware but involves software and people as well
- Performance is not always what is matter..

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.

[wikipedia]

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.

Question: Which kind of productivity are you interested in ?

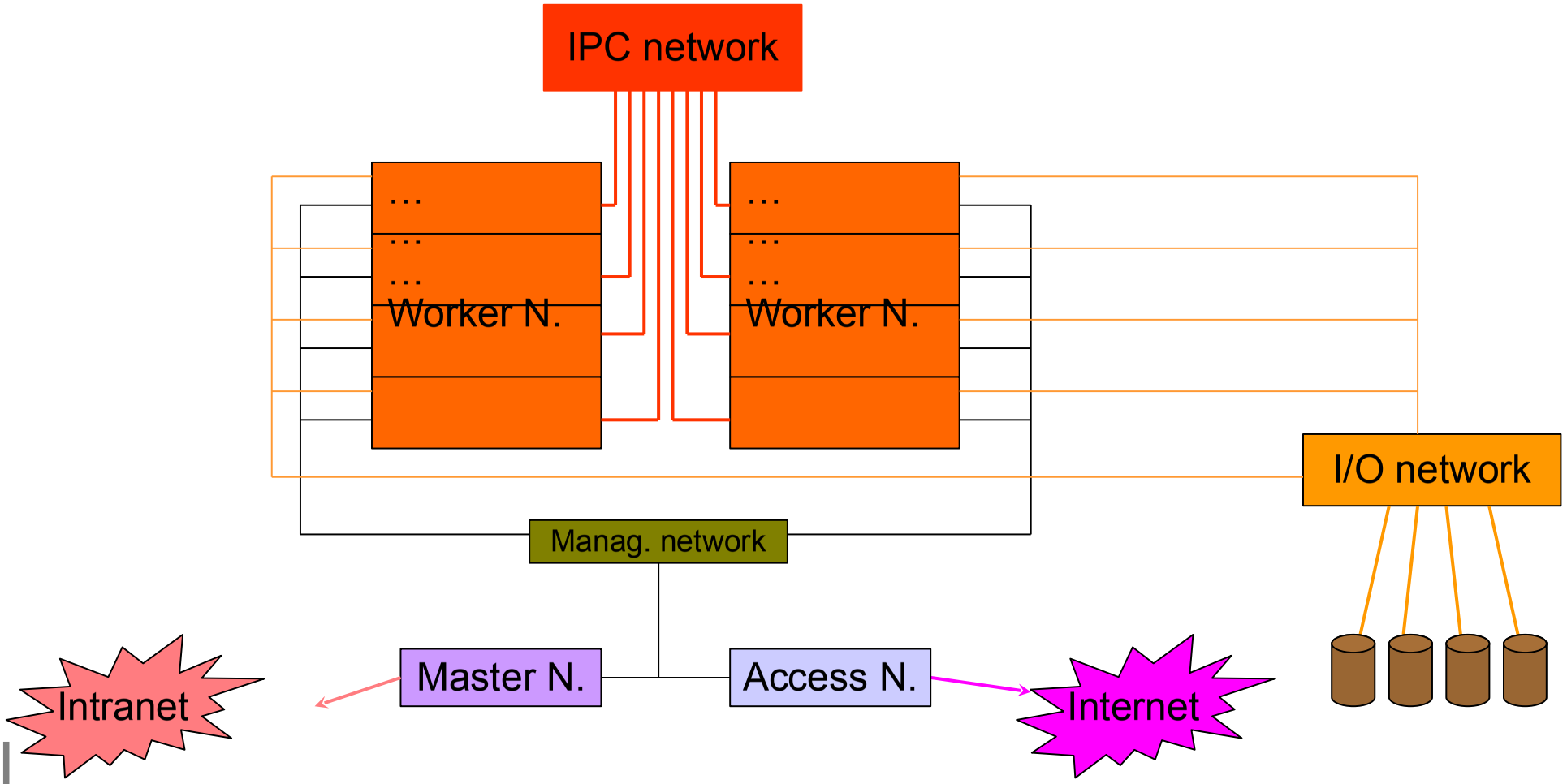
Please describe your productivity concept in your blog on the moodle platform..

HPC for masses: the cluster

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!)

HPC cluster logical structure

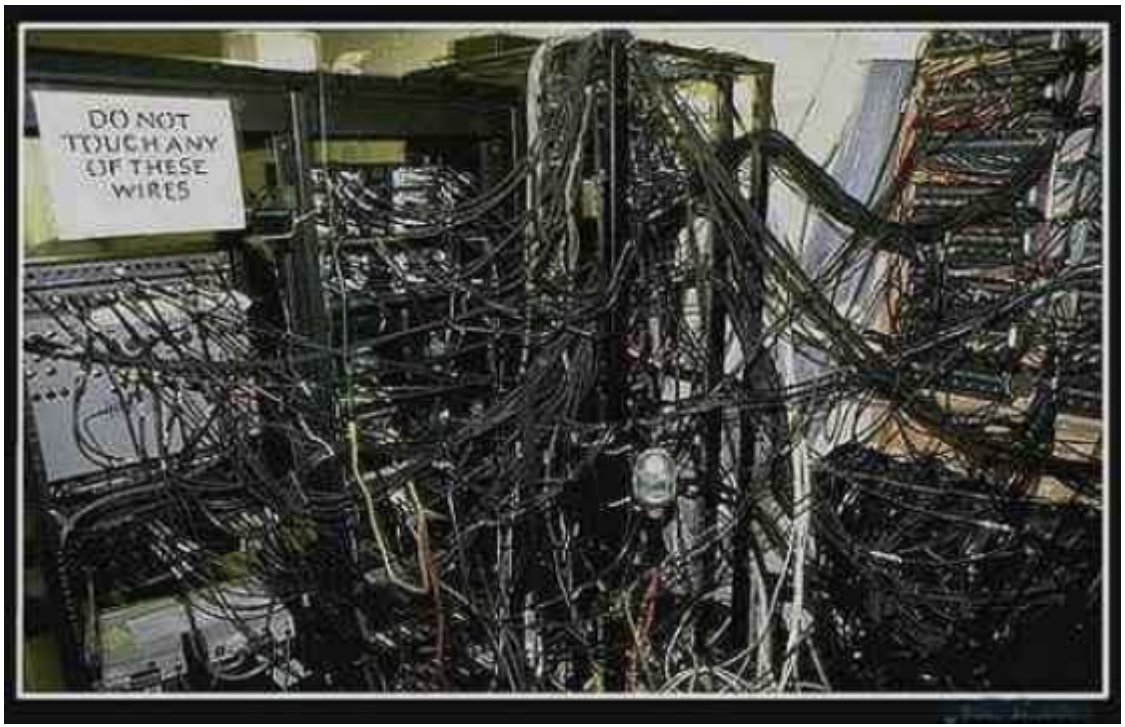


HPC clusters: 3 networks

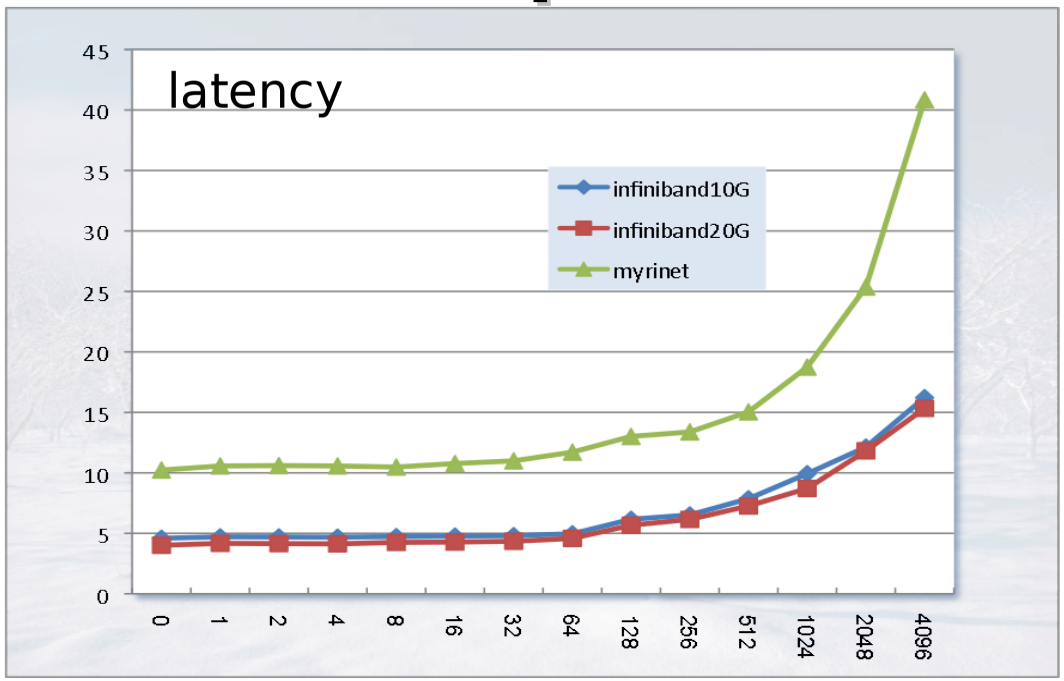
- HIGH SPEED NETWORK
  - parallel computation
    - low latency /high bandwidth
    - One choice now : Infiniband...
- I/O NETWORK
  - I/O requests (NFS and/or parallel FS)
    - latency not fundamental/ good bandwidth
    - infiniband/gigabit
- Management network
  - management traffic
    - any standard network (fast ethernet OK)

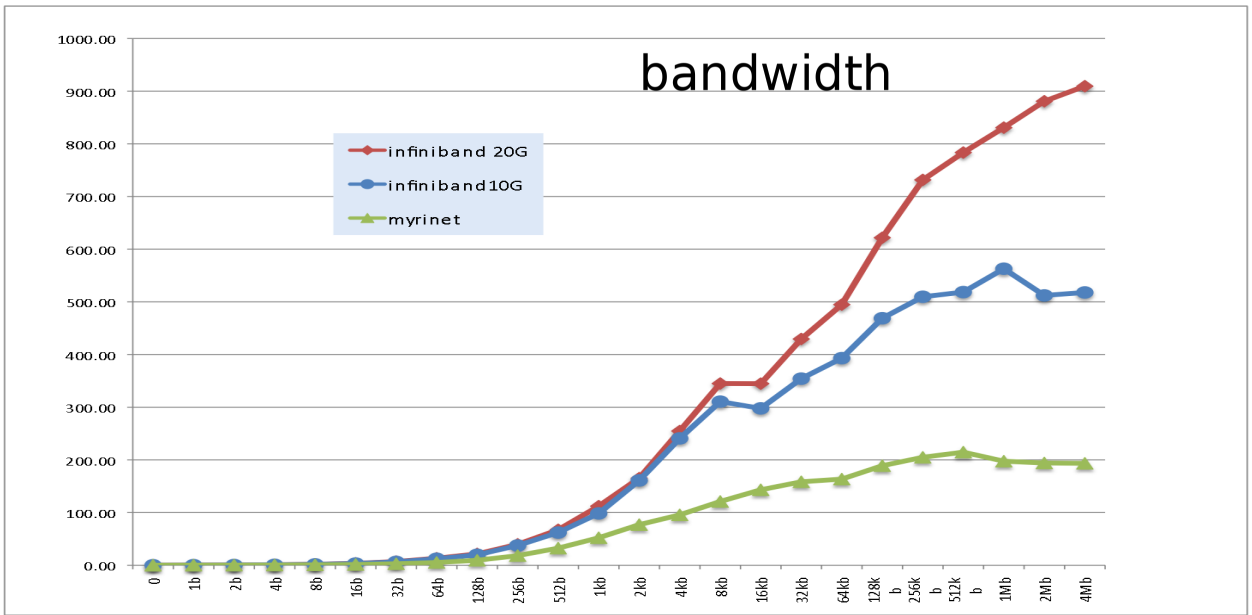
About network for clusters

- The performance of the network cannot be ignored
  - ***Latency:*** Initialization time before data can be sent
  - ***Per-link Peak Bandwidth:*** Maximum data transmission rate (varies with packet size)



HPC Network performance (SISSA example)

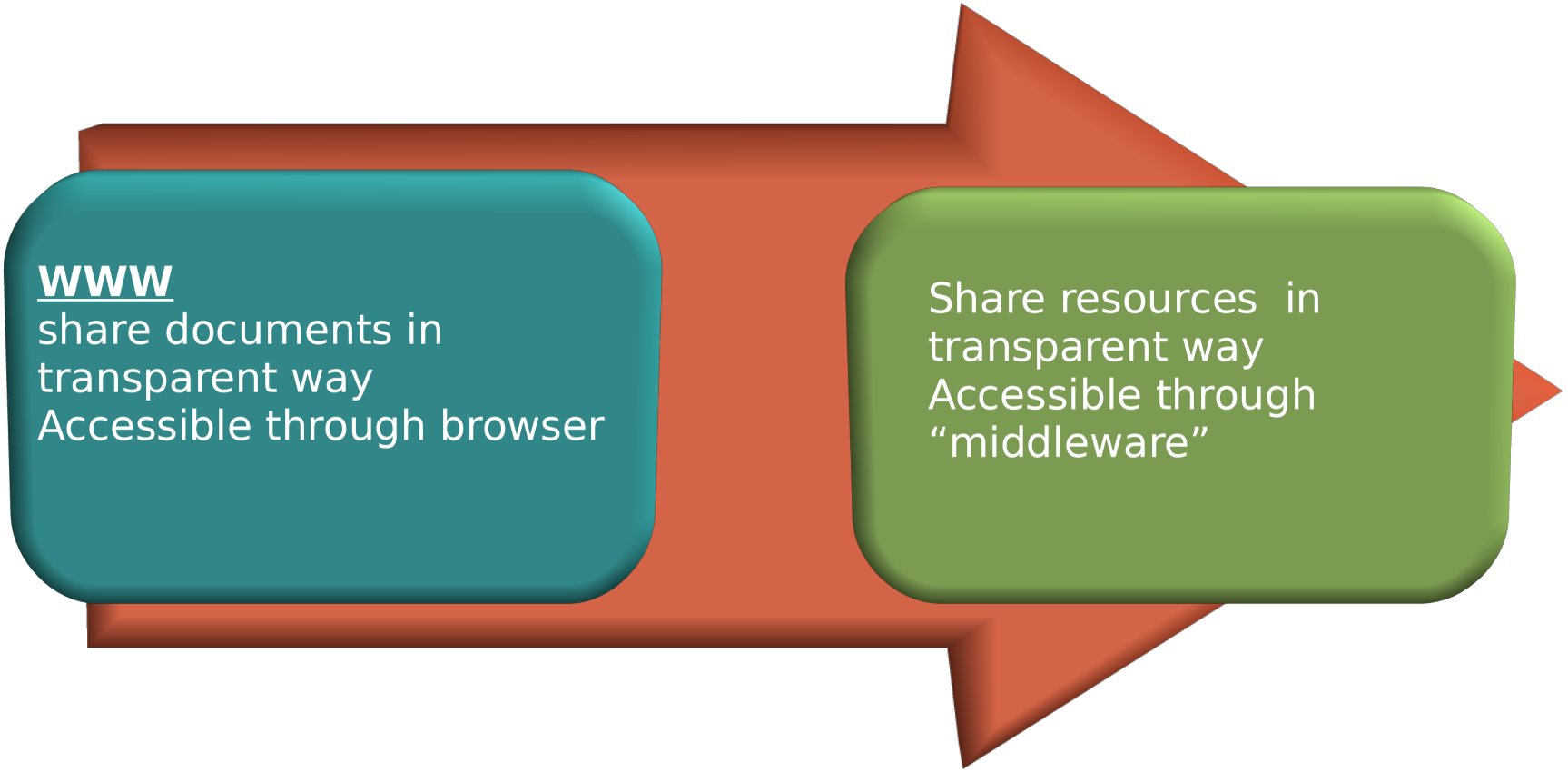




Why the GRID?

- 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.

A new paradigm



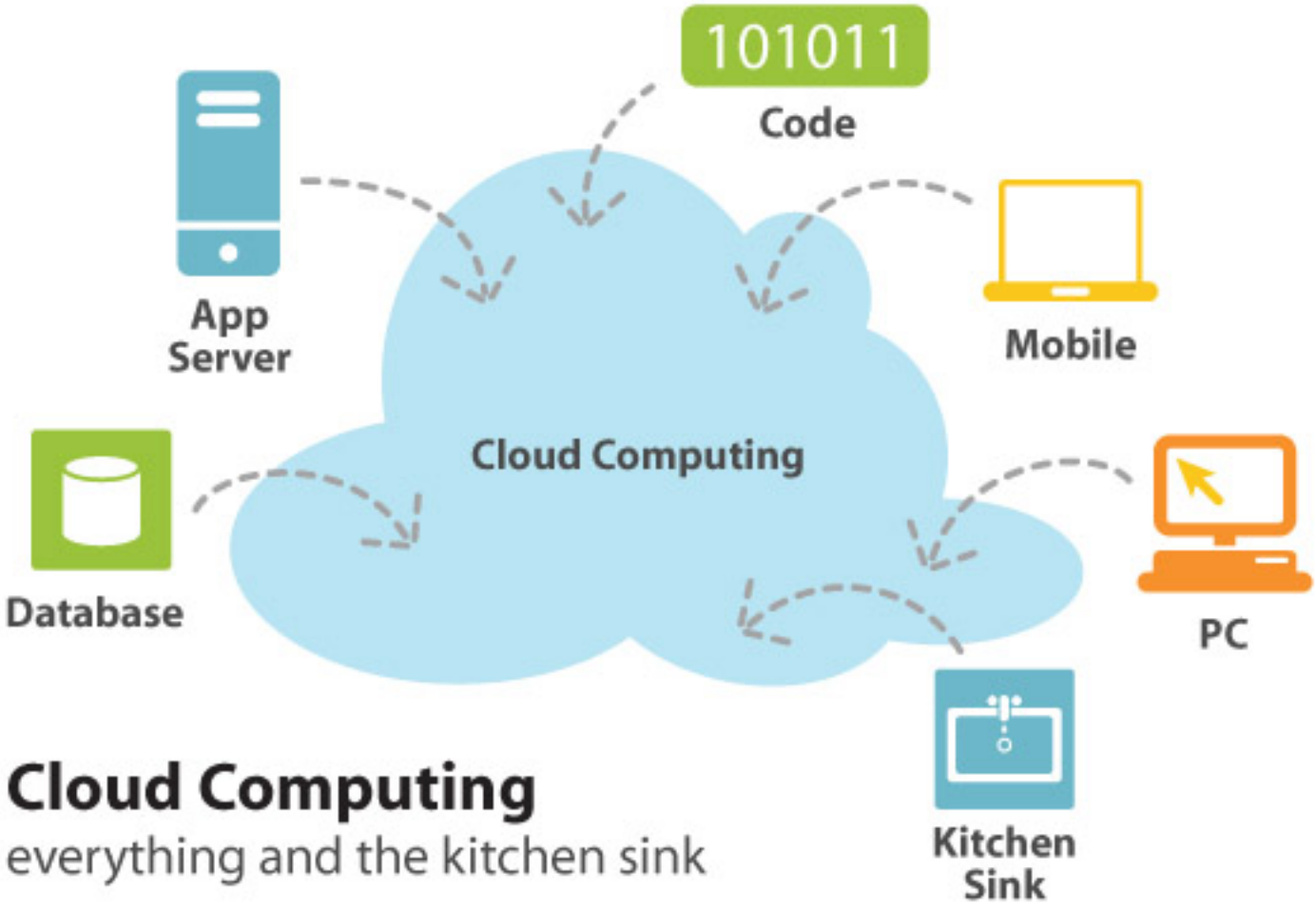
Grid Resource

- Storage systems
- Computer clusters
- HPC clusters
- Supercomputers (IBM SP, blue jean, etc)
- Databases
- Keyword: heterogeneous as regards hardware and software

Grid Middleware

- Grid is as Operating System:
  - different middleware = different Grid
- Globus alliance (Globus Toolkit)
- EMI (EU middleware)
- Unicore (DE)
- GridBus
- GRIA

Cloud Computing

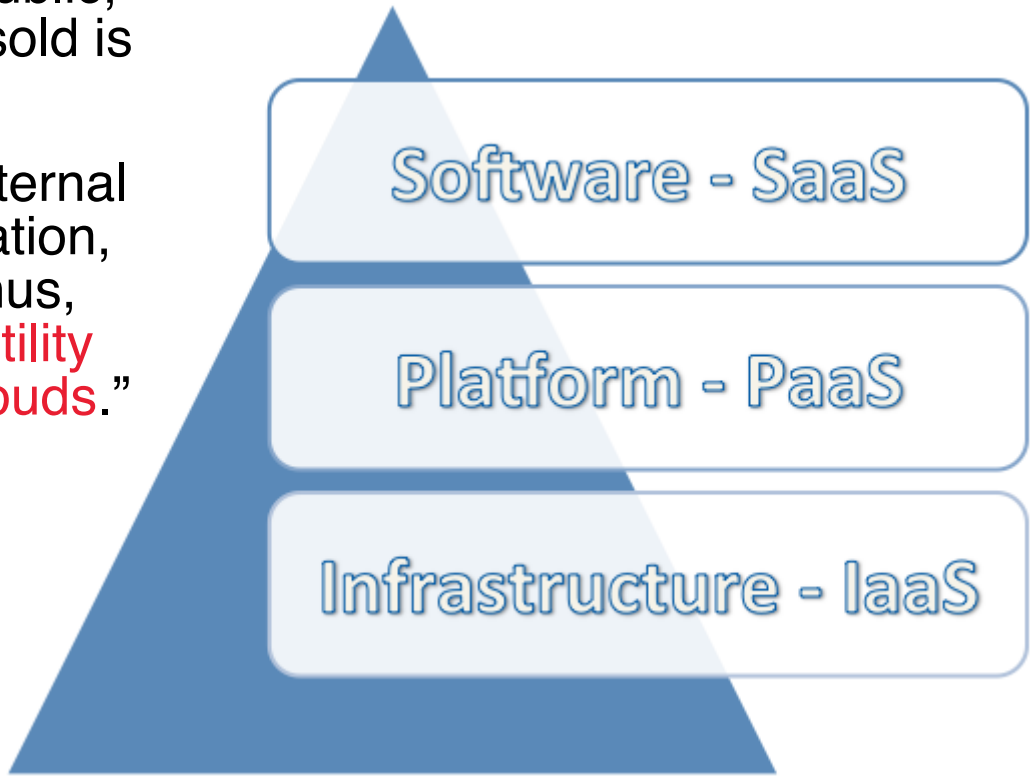


What is CC?

Cloud Computing refers to both the applications delivered as services over the Internet and the hardware and systems software in the data centers that provide those services.

The services themselves have long been referred to as Software as a Service (SaaS). The data center hardware and software is what we will call a Cloud. When a Cloud is made available in a pay-as-you-go manner to the general public, we call it a Public Cloud; the service being sold is Utility Computing.

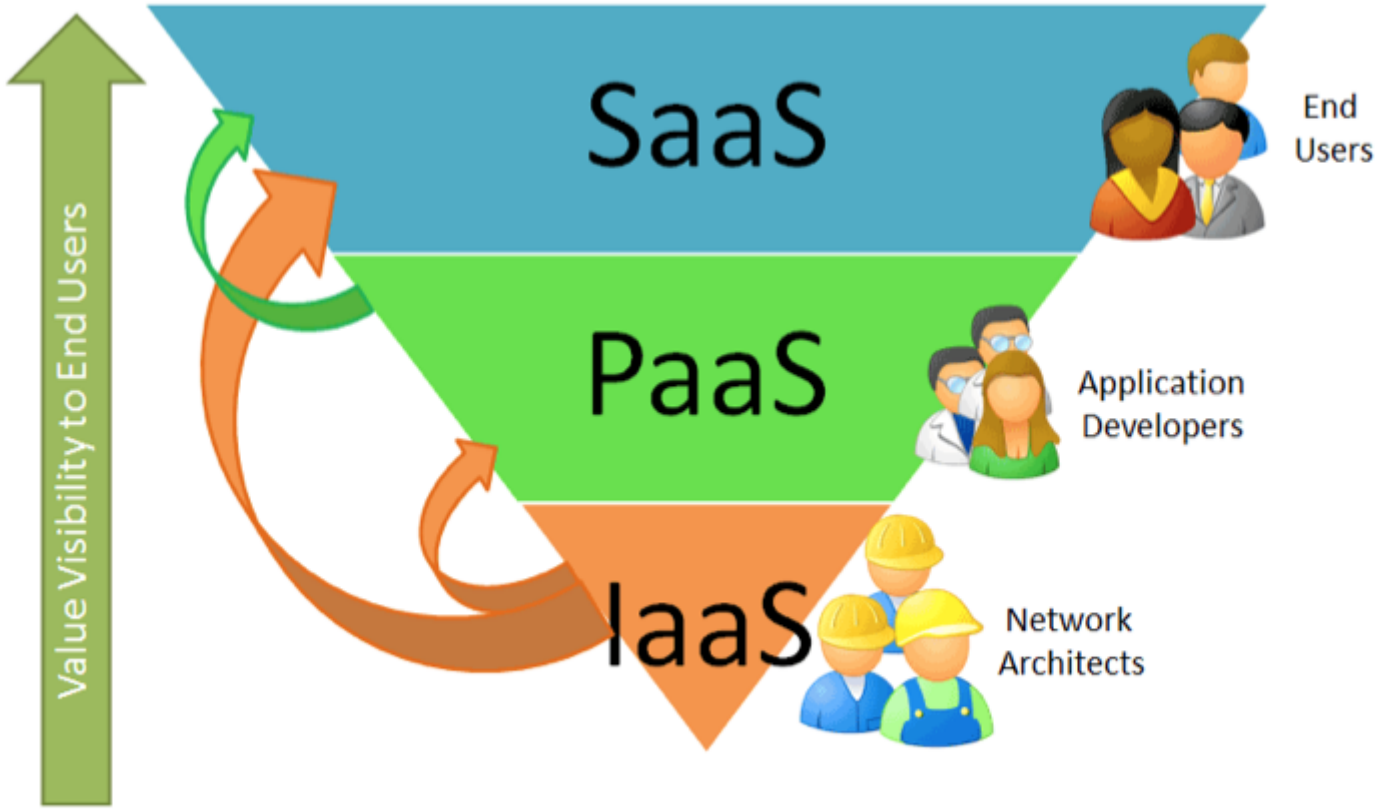
We use the term Private Cloud to refer to internal data centers of a business or other organization, not made available to the general public. Thus, Cloud Computing is the sum of SaaS and Utility Computing, but does not include Private Clouds." (Berkeley RC)



Basic Definition

- A model of computation and data storage based on “pay as you go” access to “unlimited” remote data center capabilities
- A cloud infrastructure provides a framework to manage scalable, reliable, on-demand access to applications
- Cloud services provide the “invisible” backend to many of our mobile applications
- High level of elasticity in consumption
- Historical roots in today’s Internet apps
  - Search, email, social networks
  - File storage (Live Mesh, Mobile Me, Flicker, …)

How fare are from end users?



How can I use a Cloud

- Amazon, Google, etc.
- Make your own one:
  - Eucalyptus (http://www.eucalyptus.com/)
  - OpenNebula (GLOBUS)
  - OpenQRM (http://www.openqrm.com/)
  - WS API, compatible with Amazon EC2 and S3

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
- CLOUD are outside there...

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 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
- Computing on demand: (Cloud computing)
  - Just some tasks now and then....

which kind of “computational experiment ?”

- just a single application to run
  - parallel
    - embarrassingly /tightly coupled
  - serial
- a bunch of applications linked together (workflow)

Which kind of software?

- home made codes
  - easy: users are supposed to know everything about it
- Some well know packages used with slight modification/adaptations
  - less easy: users do not know too much
- “black box” application (a.k.a. legacy application)
  - difficult: nobody knows about it

Which cluster do I need ?

- Which applications ?
  - Parallel
    - Tightly coupled
    - Loosely coupled
  - Serial
    - Memory / I/O requirements
- Which user's community ?
  - Large /Small
  - Homogeneous /heterogeneous
- Budget considerations

Which Grid infrastructure do I need ?

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

Which kind of applications/computational experiments fit/unfit on GRID ?

- FIT:
  - Parameter sweep computational experiments
  - Embarrassingly parallel computations
  - SMP/openMP programs
- UNFIT:
  - Tightly coupled parallel programs
  - I/O, Memory bounded applications

Our method to port applications..

- Identify them through user's discussions and meeting:
  - Understand the computational requirements of the group
  - Understand the level of IT skills in the group
  - Understand their concept of e-infrastructure and applications
  - Understand the application requirements
  - Propose a solution (if exists)

Steps to follow..

- Step 0: awareness of computing infrastructures opportunities /analysis of the computational requirement
- Step 1: technical deployment on the infrastructure
- Step 2: benchmarking procedures and assessment of the efficiency
- Step 3: production runs and final evaluation
- Step 4: dissemination of the results among peers.

Some tips to users (1)

- State clearly the computational requirements:
  - CPU intensive
  - I/O requirements
  - memory requirements
  - scientific libraries needed (if any)

Some tips to users (2)

- try to estimate the actual need of resource for your computational problem
  - how many times do I need to run my program ?
  - how much does it take to run ?
  - is there any room from improvement in term of performances ?

Lesson learned

- Role of human interaction is fundamental and sometimes underestimated
- GRID users speak different language from GRID providers
- GRID providers need to adopt their language NOT viceversa
- Technical tools are important BUT not sufficient to overcome the user inertia..

Our final message:

- Understand your computational problems before
  - buying/building a cluster !
  - accessing a grid !
  - Accessing Amazon E2C !

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
- Clouds means services on-demand
- HPC and GRID and Cloud computing are not mutually exclusive but can be all used to address computational resources in a transparent way.
- The challenge is now to build your own computational infrastructure driven by real needs.

Which infrastructure for my computation ?

Stefano Cozzini

Democrito and SISSA/eLAB - Trieste

Agenda

- Introduction:
  - E-infrastructure and computing infrastructures
  - What is available on the market ?
- Every Thing You Always Wanted to Know About ~~Sex~~ Computing infrastructure But Were Afraid to Ask

What is e-infrastructure ?

- E-infrastructure includes:
  - Networks (internet, light paths…)
  - Computers (workstations, servers, HPC…)
  - Access controls (security, AAA…)
  - Middleware (metadata…)
  - Finding tools (portals, search engines…)
  - Digital libraries (bibliographic, text, images, sound…)
  - Research data (national and scientific databases, individual data…)

What is a computing infrastructure?

infrastructure for computational science

- powerful and modern clusters of multicores.
  - hardware
  - software

High Performance Computing

- pooling of resources geographically distributed
- distribute collaborations

GRID COMPUTING

- Infrastructure as a Service

CLOUD COMPUTING

HPC stands for:

- High Performance Computing
- The term is most commonly associated with computing used for scientific research.

[wikipedia]

- it is not only on hardware but involves software and people as well
- Performance is not always what is matter..

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.

[wikipedia]

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.

Question: Which kind of productivity are you interested in ?

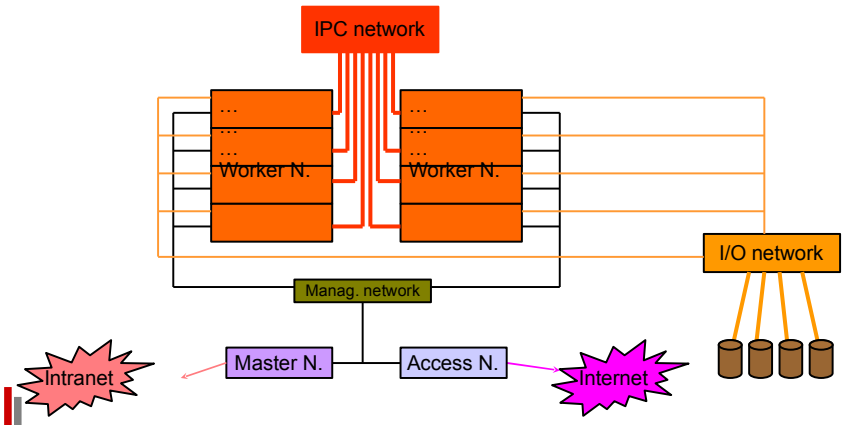
Please describe your productivity concept in your blog on the moodle platform..

HPC for masses: the cluster

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!)

HPC cluster logical structure

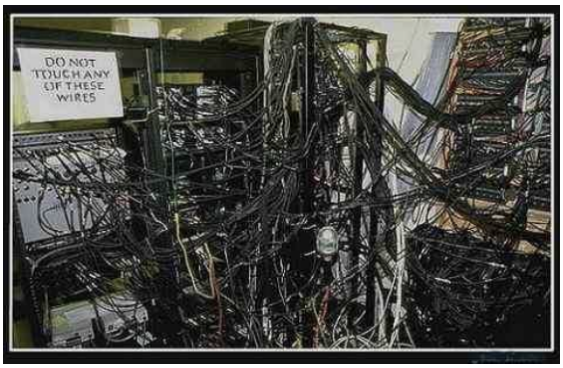


HPC clusters: 3 networks

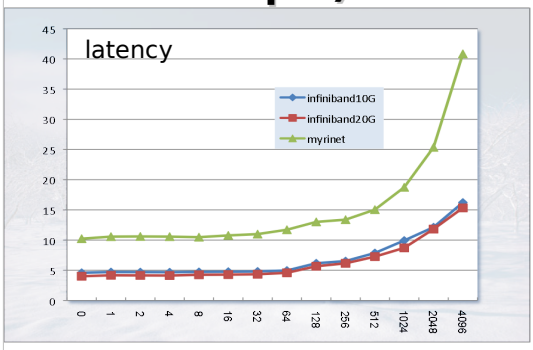
- HIGH SPEED NETWORK
  - parallel computation
    - low latency /high bandwidth
    - One choice now : Infiniband...
- I/O NETWORK
  - I/O requests (NFS and/or parallel FS)
    - latency not fundamental/ good bandwidth
    - infiniband/gigabit
- Management network
  - management traffic
    - any standard network (fast ethernet OK)

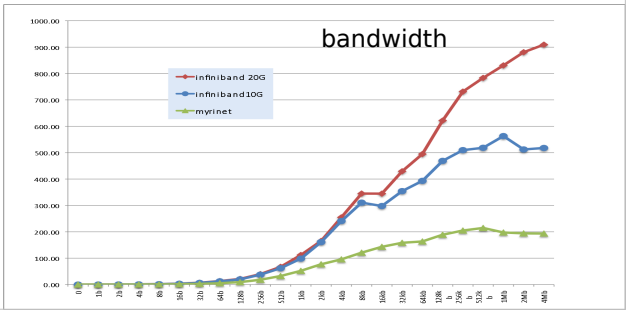
About network for clusters

- The performance of the network cannot be ignored
  - ***Latency:*** Initialization time before data can be sent
  - ***Per-link Peak Bandwidth:*** Maximum data transmission rate (varies with packet size)



HPC Network performance (SISSA example)



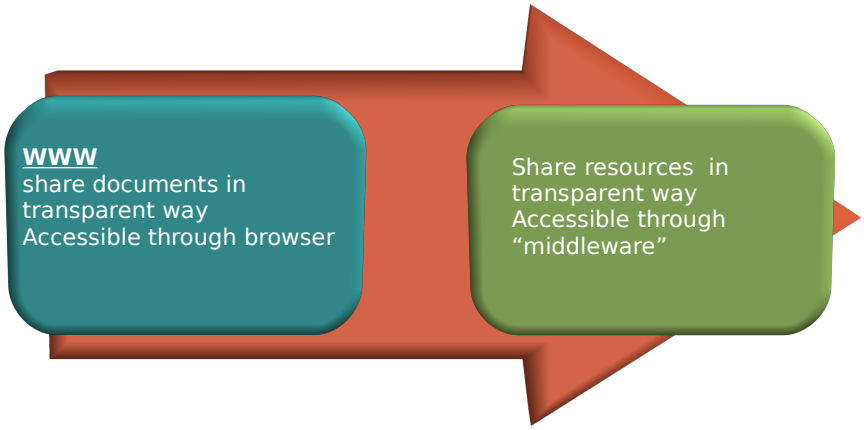


Why the GRID?

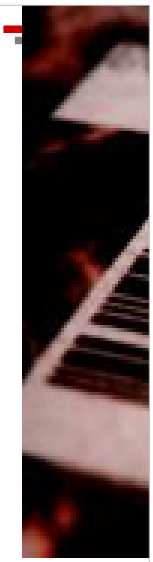
- Motivation: When communication is close to free we should not be restricted to local resources when solving problems.
- A Grid Infrastructure built on the Internet and the Web to enable and exploit large scale sharing of resources
- It should provides Scalable Secure Reliable mechanisms for discovery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

A new paradigm

- Click to add an outline



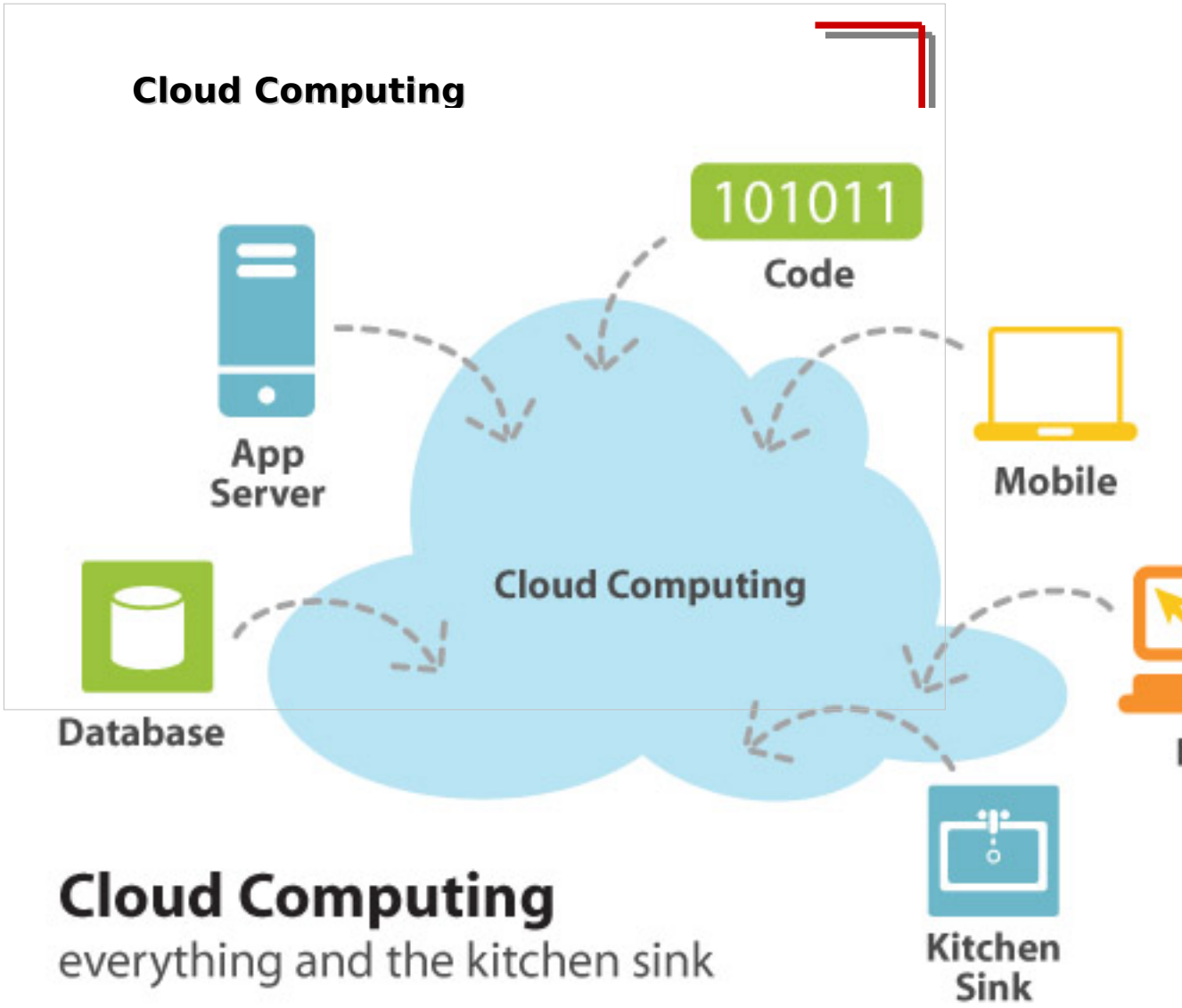
Grid Resource



- Storage systems
- Computer clusters
- HPC clusters
- Supercomputers (IBM SP, blue jean, etc)
- Databases
- Keyword: heterogeneous as regards hardware and software

Grid Middleware

- Grid is as Operating System:
  - different middleware = different Grid
- Globus alliance (Globus Toolkit)
- EMI (EU middleware)
- Unicore (DE)
- GridBus
- GRIA

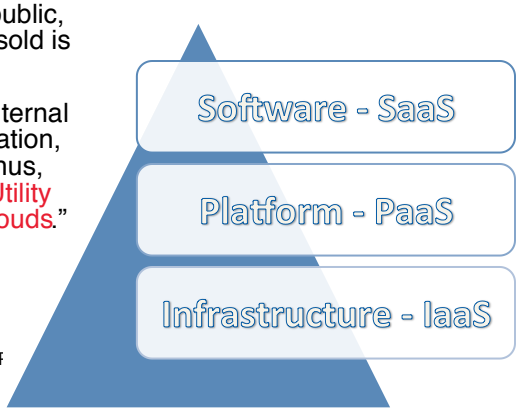

Cloud Computing
101011
Code
App
Server
Mobile
Cloud Computing
Database
Cloud Computing
everything and the kitchen sink
Kitchen
Sink

What is CC?

Cloud Computing refers to both the applications delivered as services over the Internet and the hardware and systems software in the data centers that provide those services.

The services themselves have long been referred to as Software as a Service (SaaS). The data center hardware and software is what we will call a Cloud. When a Cloud is made available in a pay-as-you-go manner to the general public, we call it a Public Cloud; the service being sold is Utility Computing.

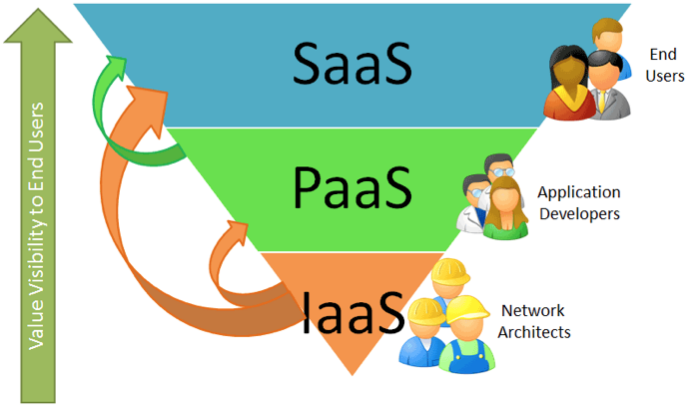
We use the term Private Cloud to refer to internal data centers of a business or other organization, not made available to the general public. Thus, Cloud Computing is the sum of SaaS and Utility Computing, but does not include Private Clouds." (Berkeley RC)



Basic Definition

- A model of computation and data storage based on “pay as you go” access to “unlimited” remote data center capabilities
- A cloud infrastructure provides a framework to manage scalable, reliable, on-demand access to applications
- Cloud services provide the “invisible” backend to many of our mobile applications
- High level of elasticity in consumption
- Historical roots in today’s Internet apps
  - Search, email, social networks
  - File storage (Live Mesh, Mobile Me, Flicker, …)

How fare are from end users?



How can I use a Cloud

- Amazon, Google, etc.
- Make your own one:
  - Eucalyptus (http://www.eucalyptus.com/)
  - OpenNebula (GLOBUS)
  - OpenQRM (http://www.openqrm.com/)
  - WS API, compatible with Amazon EC2 and S3

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
- CLOUD are outside there...

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 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
- Computing on demand: (Cloud computing)
  - Just some tasks now and then....

which kind of “computational experiment ?”

- just a single application to run
  - parallel
    - embarrassingly /tightly coupled
  - serial
- a bunch of applications linked together (workflow)

Which kind of software?

- home made codes
  - easy: users are supposed to know everything about it
- Some well know packages used with slight modification/adaptations
  - less easy: users do not know too much
- “black box” application (a.k.a. legacy application)
  - difficult: nobody knows about it

Which cluster do I need ?

- Which applications ?
  - Parallel
    - Tightly coupled
    - Loosely coupled
  - Serial
    - Memory / I/O requirements
- Which user's community ?
  - Large /Small
  - Homogeneous /heterogeneous
- Budget considerations

Which Grid infrastructure do I need ?

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

Which kind of applications/computational experiments fit/unfit on GRID ?

- FIT:
  - Parameter sweep computational experiments
  - Embarrassingly parallel computations
  - SMP/openMP programs
- UNFIT:
  - Tightly coupled parallel programs
  - I/O, Memory bounded applications

Our method to port applications..

- Identify them through user's discussions and meeting:
  - Understand the computational requirements of the group
  - Understand the level of IT skills in the group
  - Understand their concept of e-infrastructure and applications
  - Understand the application requirements
  - Propose a solution (if exists)

Steps to follow..

- Step 0: awareness of computing infrastructures opportunities /analysis of the computational requirement
- Step 1: technical deployment on the infrastructure
- Step 2: benchmarking procedures and assessment of the efficiency
- Step 3: production runs and final evaluation
- Step 4: dissemination of the results among peers.

Some tips to users (1)

- State clearly the computational requirements:
  - CPU intensive
  - I/O requirements
  - memory requirements
  - scientific libraries needed (if any)

Some tips to users (2)

- try to estimate the actual need of resource for your computational problem
  - how many times do I need to run my program ?
  - how much does it take to run ?
  - is there any room from improvement in term of performances ?

Lesson learned

- Role of human interaction is fundamental and sometimes underestimated
- GRID users speak different language from GRID providers
- GRID providers need to adopt their language NOT viceversa
- Technical tools are important BUT not sufficient to overcome the user inertia..

Our final message:

- Understand your computational problems before
  - buying/building a cluster !
  - accessing a grid !
  - Accessing Amazon E2C !

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
- Clouds means services on-demand
- HPC and GRID and Cloud computing are not mutually exclusive but can be all used to address computational resources in a transparent way.
- The challenge is now to build your own computational infrastructure driven by real needs.