From Clusters to Grids

Master 2 Research Lecture: Parallel Systems

Vincent Danjean, MCF UJF, LIG/INRIA/Moais
Derick Kondo, CR INRIA, LIG/INRIA/Mescal
Arnaud Legrand, CR CNRS, LIG/INRIA/Mescal
Jean-François Méhaut, PR UJF, LIG/INRIA/Mescal
Bruno Raffin, CR INRIA, LIG/INRIA/Moais
Jean-Louis Roch, MCF ENSIMAG, LIG/INRIA/Moais
Alexandre Termier, MCF UJF, LIG/Hadas

LIG laboratory, arnaud.legrand@imag.fr

October 13, 2008

Outline

1. Clusters

2. What next?

Outline

1. Clusters
2. What next?

Motivation

Parallel Machines

- Parallel machines are expensive.
- The development tools for workstations are more mature than the contrasting proprietary solutions for parallel computers - mainly due to the non-standard nature of many parallel systems.

Workstation evolution

- Surveys show utilization of CPU cycles of desktop workstations is typically $< 10\%$.
- Performance of workstations and PCs is rapidly improving
- The communications bandwidth between workstations is increasing as new networking technologies and protocols are implemented in LANs and WANs.
- As performance grows, percent utilization will decrease even further! Organizations are reluctant to buy large supercomputers, due to the large expense and short useful life span.

Towards clusters of workstations

- Workstation clusters are easier to integrate into existing networks than special parallel computers.
- Workstation clusters are a cheap and readily available alternative to specialized High Performance Computing (HPC) platforms.
- Use of clusters of workstations as a distributed compute resource is very cost effective - incremental growth of system!!!

Definition.

A cluster is a type of parallel or distributed processing system (MIMD), which consists of a collection of interconnected stand-alone/complete computers cooperatively working together as a single, integrated computing resource.

Definition

A typical cluster

- A cluster is mainly homogeneous and is made of high performance and generally rather low cost components (PCs, Workstations, SMPs).
- Composed of a few to hundreds of machines.
- Network: Faster, closer connection than a typical LAN network; often a high speed low latency network (e.g. Myrinet, InfiniBand, Quadrix, etc.); low latency communication protocols; looser connection than SMP.

Typical usage

- Dedicated computation (rack, no screen and mouse).
- Non dedicated computation: Classical usage during the day (word, latex, mail, gcc) / HPC applications usage during the night and week-end.

Biggest clusters can be split in several parts:

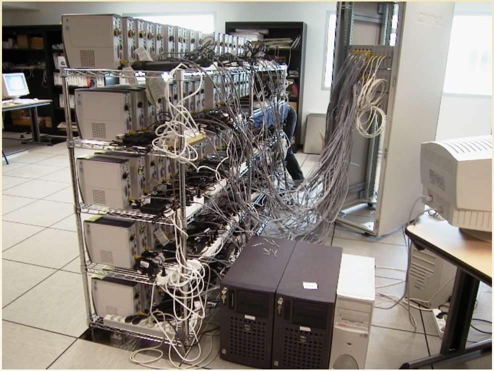
- computing nodes;
- I/O nodes;
- front (interactive) node.

A few examples

Berkeley NOW (1997)

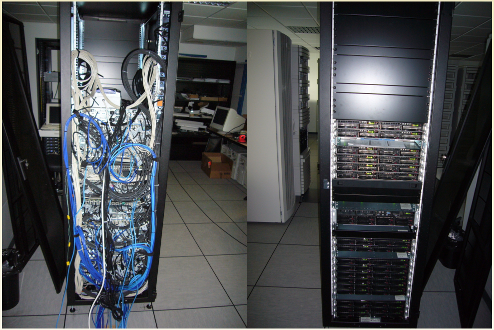
- 100 SUN UltraSPARCs.
- Myrinet 160MB/s.
- Fast Ethernet.

A few examples



Icluster (2000)

- 225 HP iVectra PIII 733 Mhz.
- Fast Ethernet.
- 81.6 Gflops (216 nodes).
- top 500 (385) June 2001.



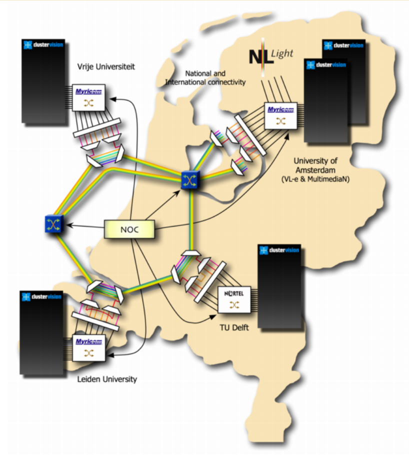
Digitalis (2008)

- 34 nodes (2 xeon quad cores) $\rightsquigarrow$ 272 cores with $2 \times 8Gb$ of RAM and $2 \times 160Gb$ of HD each.
- Infiniband.
- Giga Ethernet.

Outline

Clusters of clusters (HyperClusters)

DAS3: ASCI (Advanced School for Computing and Imaging),Netherlands.



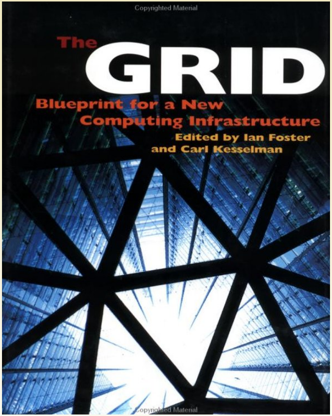
- Five Linux supercomputer clusters with 550 AMD Opteron processors.
- 1TB of memory and 100TB of storage.
- Myricom Myri-10G network inside clusters.
- Clusters are interconnected by a SURFnet's multi-color optical backbone.

The concept of Grid...

The Grid: Blueprint for a New Computing Infrastructure (1998); Ian Foster, Carl Kesselman, Jack Dongarra, Fran Berman, ...

Analogy with the electric supply:

- You don't know where the energy comes from when you turn on your coffee machine.
- You don't need to know where your computations are done.



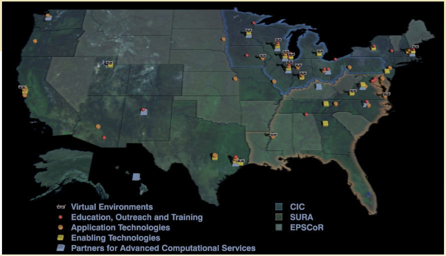
The concept of Grid (cont'd)

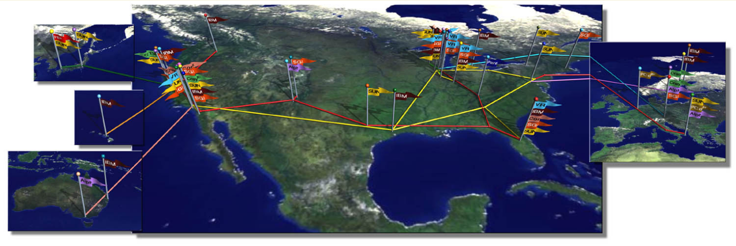
A grid is an infrastructure that couples:

- Computers (PCs, workstations, clusters, traditional supercomputers, and even laptops, notebooks, mobile computers, PDA, and so on);
- Software Databases (e.g., transparent access to human genome database);
- Special Instruments (e.g., radio telescope–SETI@Home Searching for Life in galaxy, Astrophysics@Swinburne for pulsars, a cave);
- People (maybe even animals who knows ?;-)

across the local/wide-area networks (enterprise, organizations, or Internet) and presents them as an unified integrated (single) resource.

What does a Grid look like?





It is very big and very heterogeneous!

Various versions of *"Grid"*

You have probably heard of many *buzzwords*.

- Super-computing;
- Global Computing;
- Internet Computing;
- Grid Computing;
- Meta-computing;
- Web Services;
- Cloud Computing;
- Ambient computing;
- Peer-to-peer;
- Web;

Large Scale Distributed Systems

"A distributed system is a collection of independent computers that appear to the users of the system as a single computer"
Distributed Operating System. A. Tannenbaum, Prentice Hall, 1994

Tentative taxonomy

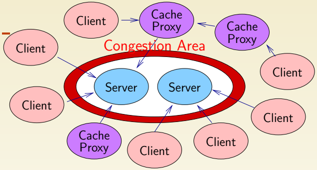
Purpose

- Information: share knowledge.
- Data: large-scale data storage.
- Computation: aggregate computing power.

Deployment model

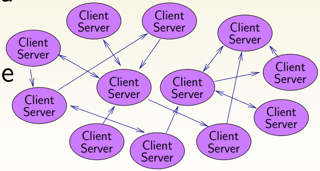
- Not necessarily fully centralized.
- Use of caches and proxys to reduce congestion.
- Hierarchical structure is often used.
- Centralized information

Client/server



- Each peer acts both as a client and a server.
- The load is distributed over the whole network.
- Distributed information.

Peer-to-peer



Example: Web sites
Client/server; information grid

Context

- Probably the first "grid".
- Information is accessed through a URL or more often through a search engine.
- Information access is fully transparent: you generally don't know where the informations comes from (mirrors, RSS feeds,...).

Challenges Going peer-to-peer ? Web 2.0: users also contribute.

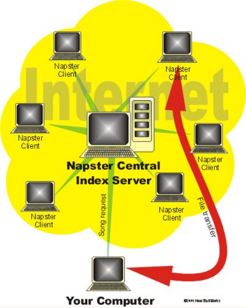
- Social networks (Facebook).
- Recommendations (google and amazon.com).
- Crowdsourcing (wikipedia, marmiton).
- Video and photo sharing (youtube).
- Media improvement (e.g., linking picassa and google maps).
- Ease of finding relevant information and ability to tag data.

Context

- The first massively popular "peer-to-peer" file (MP3 only) sharing system (1999).
- Central servers maintain indexes of connected peers and the files they provide.
- Actual transactions are conducted directly between peers.

Drawbacks

- More client/server than truly peer-to-peer.
- Hence, servers have been attacked (by courts and by others to track peers offering copyrighted materials).

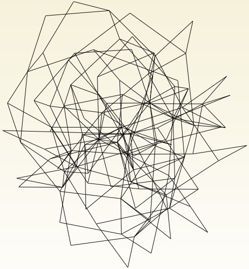


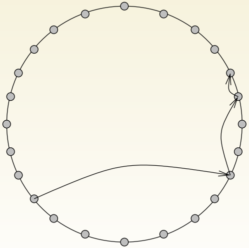
Example: Gnutella, Kazaa, Freenet, Chord

P2P; data grid

Context

- Removal of servers: searching can be done by flooding in unstructured overlays.
- Use of supernodes/ultrapeers (nodes with a good CPU and high bandwidth) for searching.
- Structured (hypercubes, torus, ...) overlay networks.
- Downloading from multiple sources using hash blocks and redundancy.





Example: Gnutella, Kazaa, Freenet, Chord

P2P; data grid

Context

- Removal of servers: searching can be done by flooding in unstructured overlays.
- Use of supernodes/ultrapeers (nodes with a good CPU and high bandwidth) for searching.
- Structured (hypercubes, torus, . . .) overlay networks.
- Downloading from multiple sources using hash blocks and redundancy.

Challenges

- Ensuring anonymity.
- Ensuring good throughput and efficient multi-cast (network coding, redundancy).
- Avoiding polluted data.
- Publish-subscribe overlays for fuzzy or complex queries.
- Free-riders.

Example: Internet Computing (SETI@home)

Client/server; computation grid

Context

- Search for possible evidence of radio transmissions from extraterrestrial intelligence using data from a telescope.
- The client is generally embedded into a screensaver.
- The server distributes the work-units to volunteer clients.
- Attracting volunteers with hall of fame and teams.
- Need to cross-check the results to detect false positives.
- 5.2 million participants worldwide, over two million years of aggregate computing time since its launch in 1999. 528 TeraFLOPS (Blue Gene peaks at just over 596 TFLOPS with sustained rate of 478 TFLOPS).
- Evolved into BOINC: Berkeley Open Infrastructure for Network Computing (climate prediction, protein folding, prime number factorizing, fight cancer, Africa@home, . . .).

Example: Internet Computing (SETI@home)

Client/server; computation grid

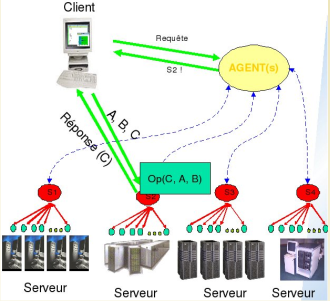
Challenges

- Attract more volunteers: credits, ribbons and medals, connect with facebook.
- Volunteer thinking: use people's brains (intelligence, knowledge, cognition) to locate' solar dust, fossils, fold proteins.
- Works well for computation intensive embarrassingly parallel applications.
  - Really parallel applications.
  - Data intensive applications.
  - Soft real-time applications.
- Security.
  - Would you let anyone execute anything on your PC?
  - Use sandboxing and virtual machines.
- Need to go peer-to-peer (CGP2P, OurGrid).

Example: Meta-computing
Client/server; computation grid

Context

- Principle: buy computing services (pre-installed applications + computers) on the Internet.
- Examples: Netsolve (UTK), NINF (Tsukuba), DIET and Scilab // (ENS Lyon/INRIA),



Challenges

- Data storage and distribution: avoid multiple transfers between clients and servers when executing a sequence of operations.
- Efficient data redistribution.
- Security for file transfers
- Peer-to-peer deployment.

Context

- Principle: use a *virtual* supercomputer and execute applications on remote resources.
  "I need 200 64 bits machines with 1Tb of storage from 10:20 am to 10:40 pm."
- Need to match and locate resources, schedule applications, handle reservations, authentication, ...
- Examples: Globus, Legion, Unicore, Condor, ...

Challenges

- Obtaining good performances while deploying parallel codes on multiple domains.
- Communication and computation overlap. High-performance communications on heterogeneous networks.
- Need for new parallel algorithms that handle heterogeneity, hierarchy, dynamic resources,
- Complex applications $\rightsquigarrow$ code coupling (message passing $\rightsquigarrow$ distributed objects, components).

Summary

Usage \ Deployment	Client/Server	Peer-to-peer
Data	Napster	Gnutella, Kazaa, Chord, Freenet...
Information	Web 1.0 and 1.5 Search Engines	Web 2.0
Computing	Internet Computing; Meta-computing; Grid Computing	OurGrid

A few other challenges

- Security, Authentication, Trust, Error management.
- Middleware vs. Operating System.
- Algorithms for Grid Computing.
- Software engineering.
- Social aspects (fairness, selfishness, cooperation).
- Energy saving!

Conclusion

- No real new theme but rather a combination of already existing technologies for parallel and distributed computing.
- Such combinations and ambitious goals are very hard to achieve.
- This clearly requires a pluri-disciplinary approach with a good understanding of all aspects (OS, network, middleware, security, storage, algorithms, applications, ...).
- It would be a mistake to restrict only to computing. Research on all these aspects should be encouraged.
- It is very important to identify and discriminate new concepts from technology and fad.
- A crucial question is:

  "Should we hide the complexity or expose it?"