Architectures Hautes Performances

Arnaud Legrand
Chargé de Recherches, CNRS/ID-IMAG

Jean-François Méhaut
Professeur, UJF/ID-IMAG

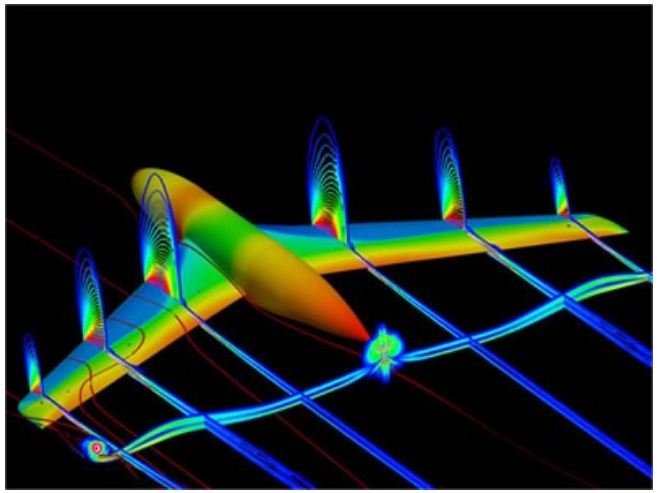
Objectifs

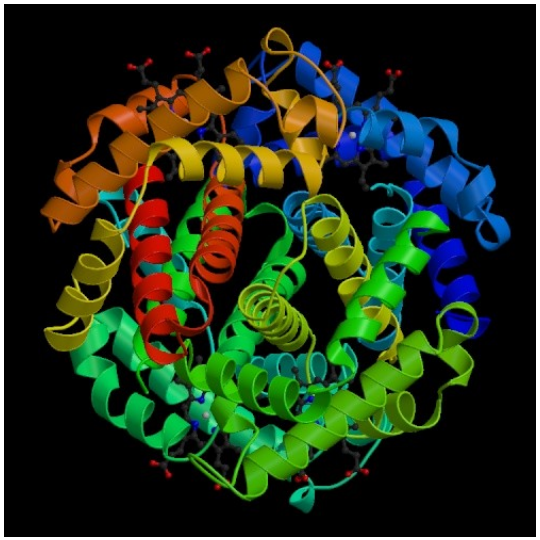
- Comprendre
  - Architecture matérielle et logicielle des grappes et des grilles de calcul
  - Aspects fondamentaux et technologiques
  - Différents types d'application
  - Nouvelles problématiques scientifiques

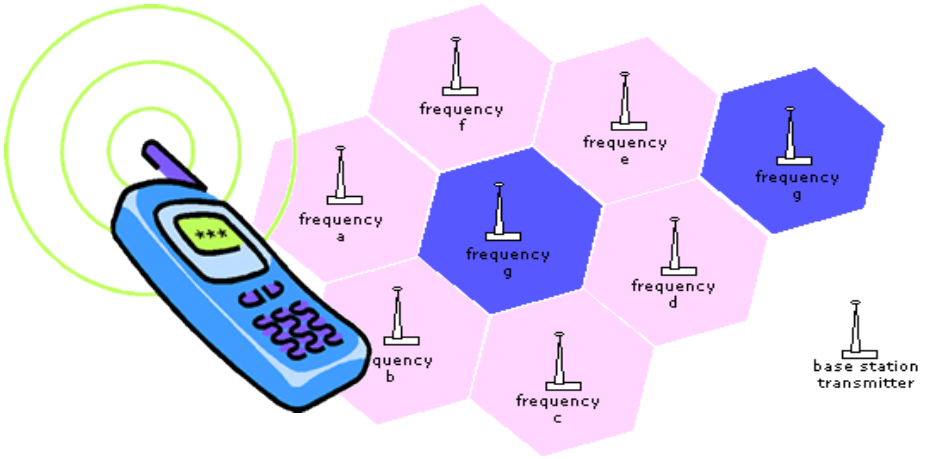
Plan du cours

- Architectures pour le calcul haute performance (JFM)
  - Cluster Computing, Grid Computing, Internet Computing, Intranet Computing
- Modélisation et algorithmique parallèle (AL)
  - Tâches parallèles
  - Ordonnancement avec communications
  - Tâches divisibles
  - Steady state
- Communication sur réseaux rapides (JFM)
  - Techniques et méthodes de transfert des données, MPI
  - Pourquoi MPI ne peut pas optimiser davantage ?

Calcul hautes performances
Pourquoi ?



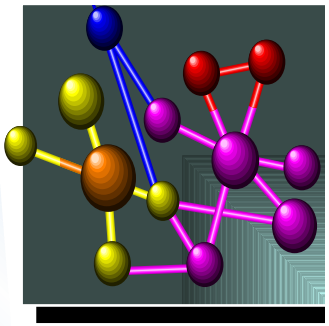




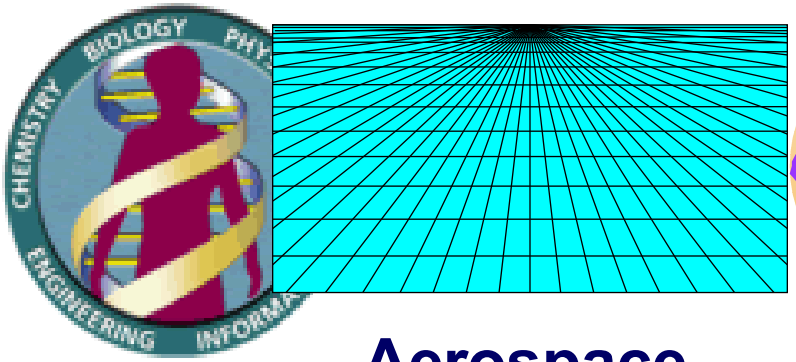
Computing Power Drivers

Résolution d'applications “Grand Challenge” en utilisant la puissance des ordinateurs

modélisation, *simulation* et *analyse*

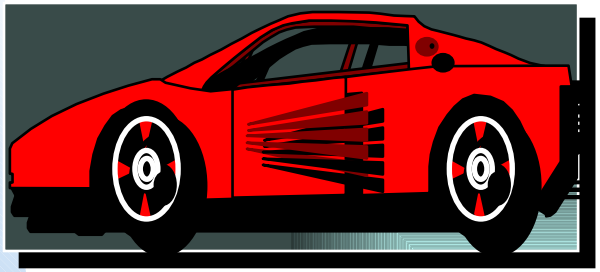


Life Sciences



Aerospace

E-commerce

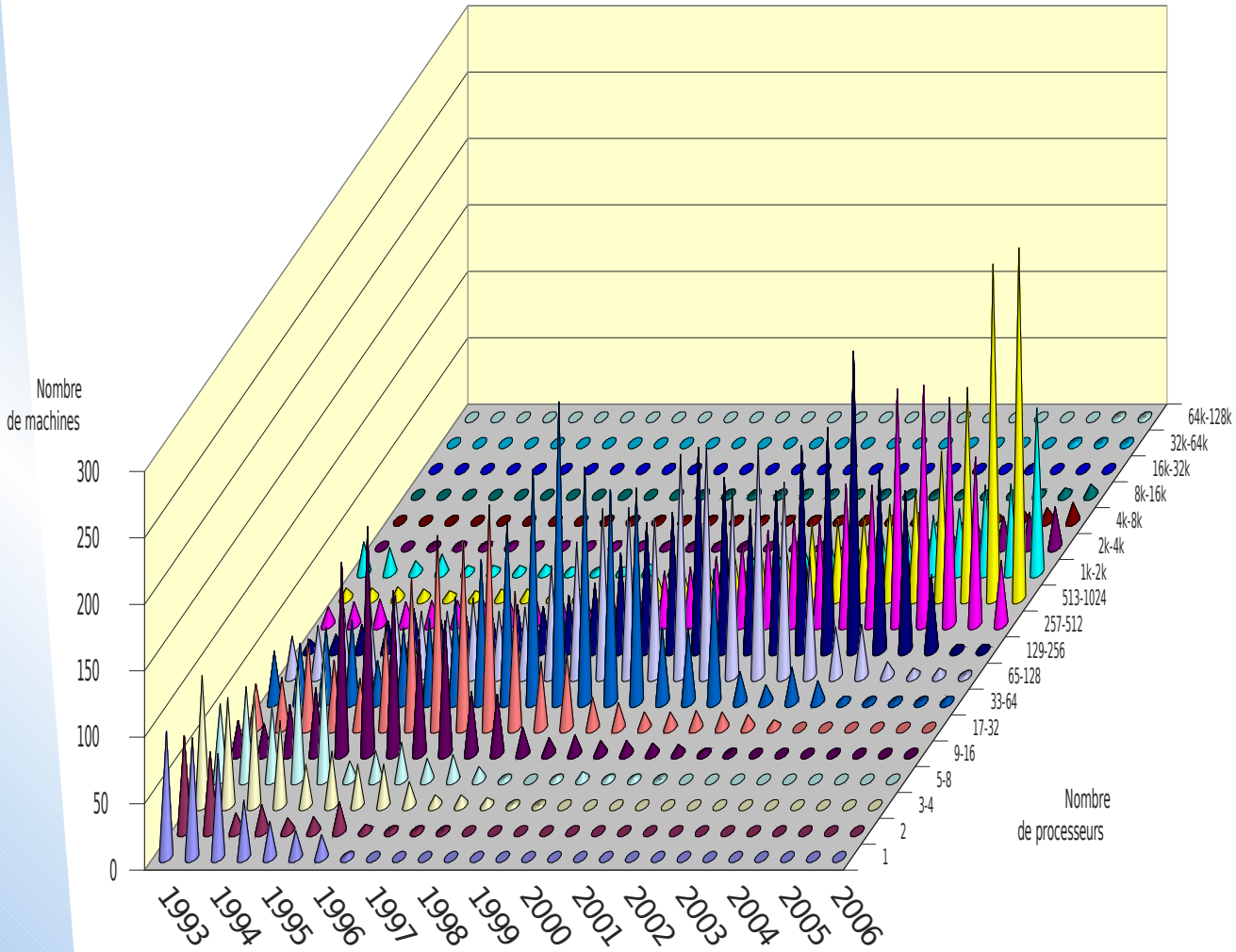


CAD/CAM

Digital Biology

Military Applications

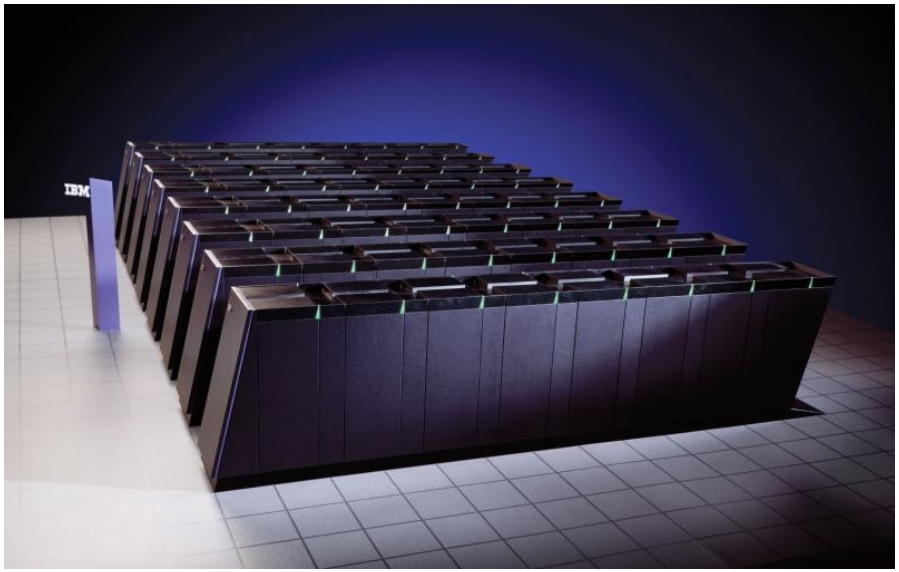
Calcul hautes performances
Évolution du nombre de processeurs

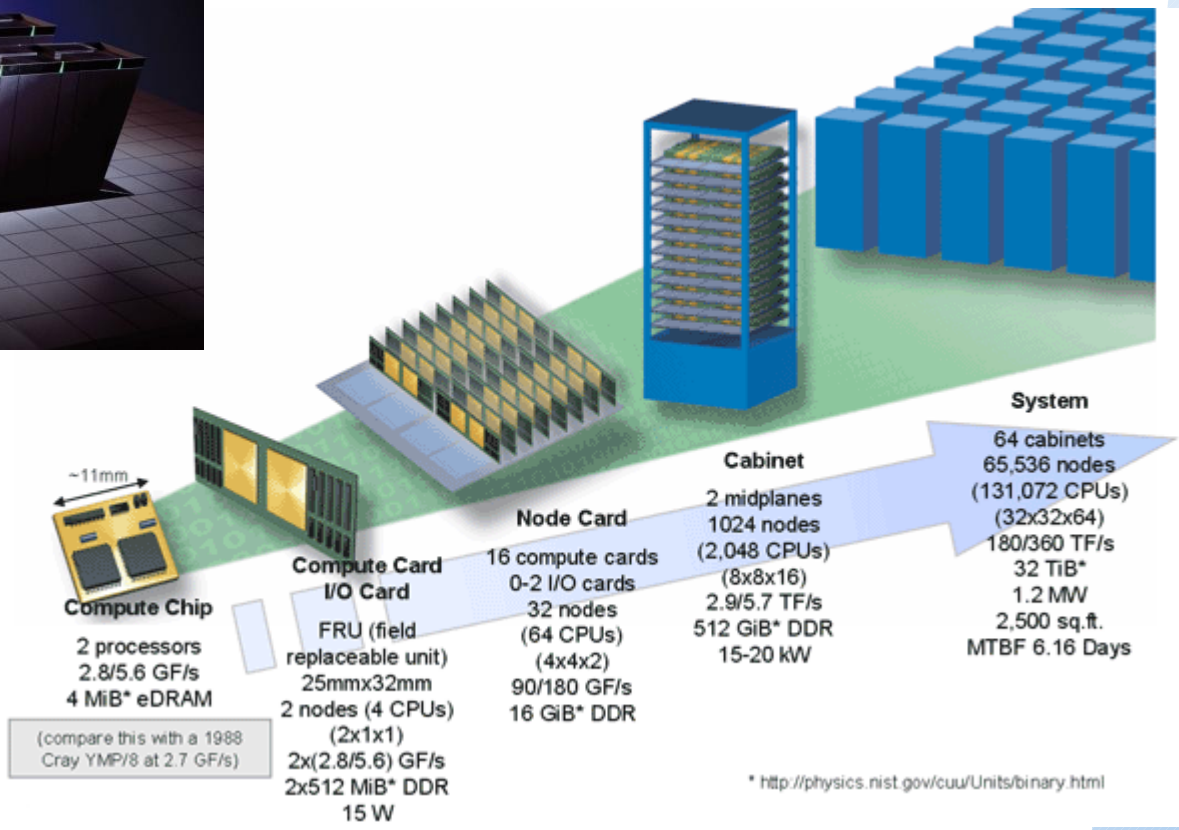


2006 Xbox 360
1TFlop
1 GPU

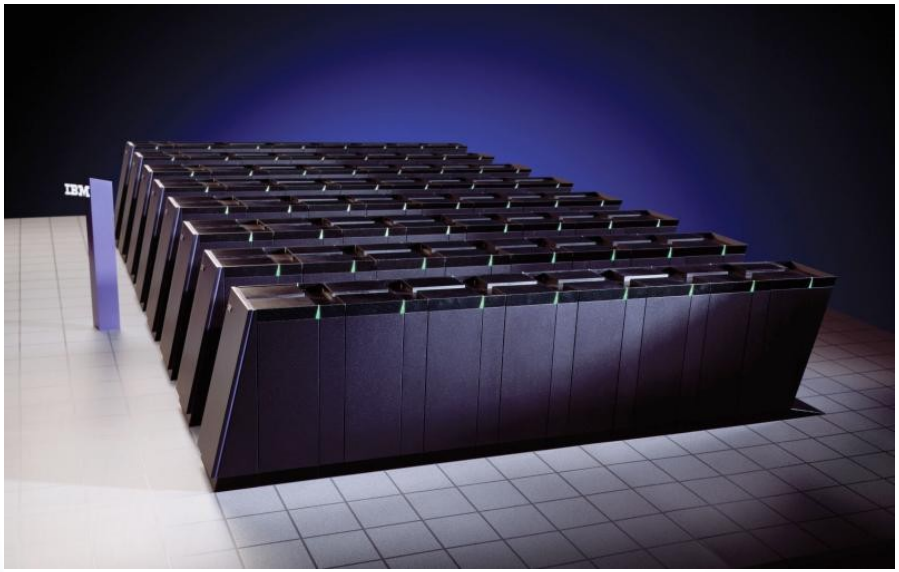
2006 AMD
Opteron
3Gflop

Calcul hautes performances
Programmation des machines parallèles

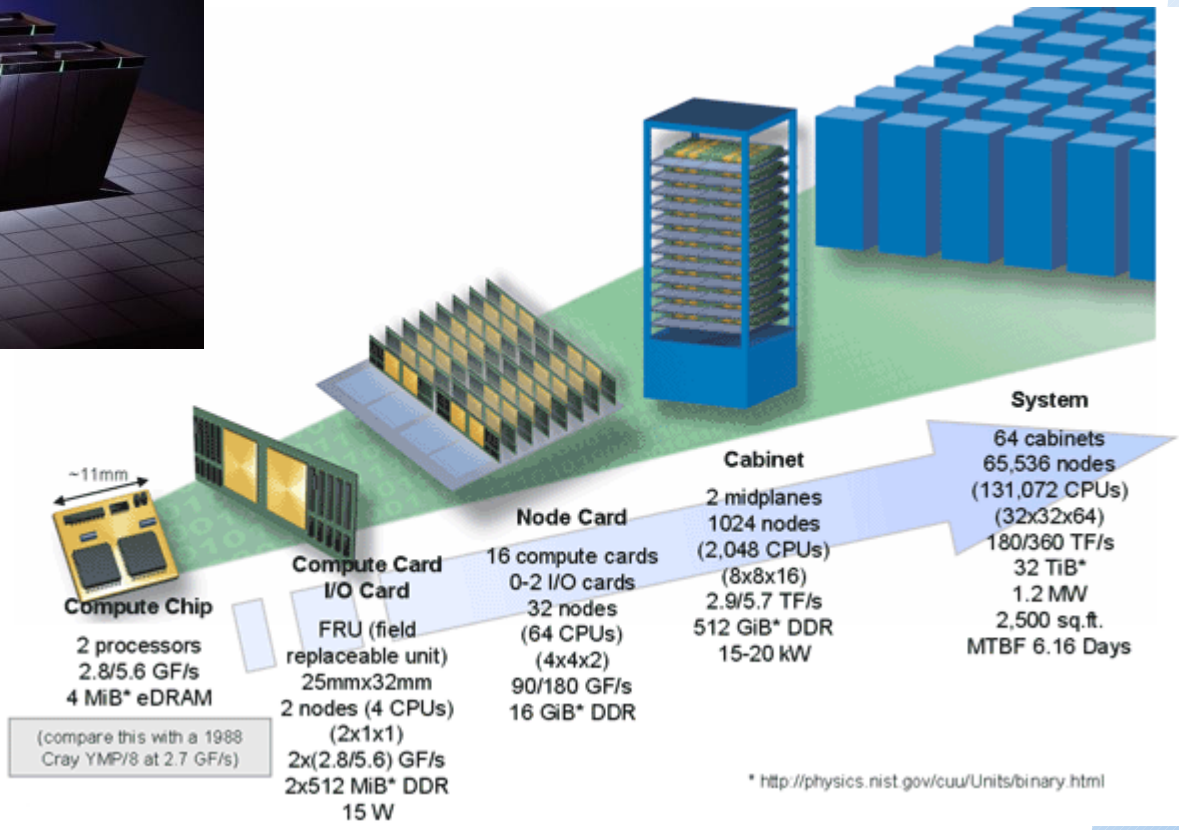




Calcul hautes performances
Programmation des machines parallèles



Communications entre processus
Mémoire partagée : OpenMP
Passage de message : MPI



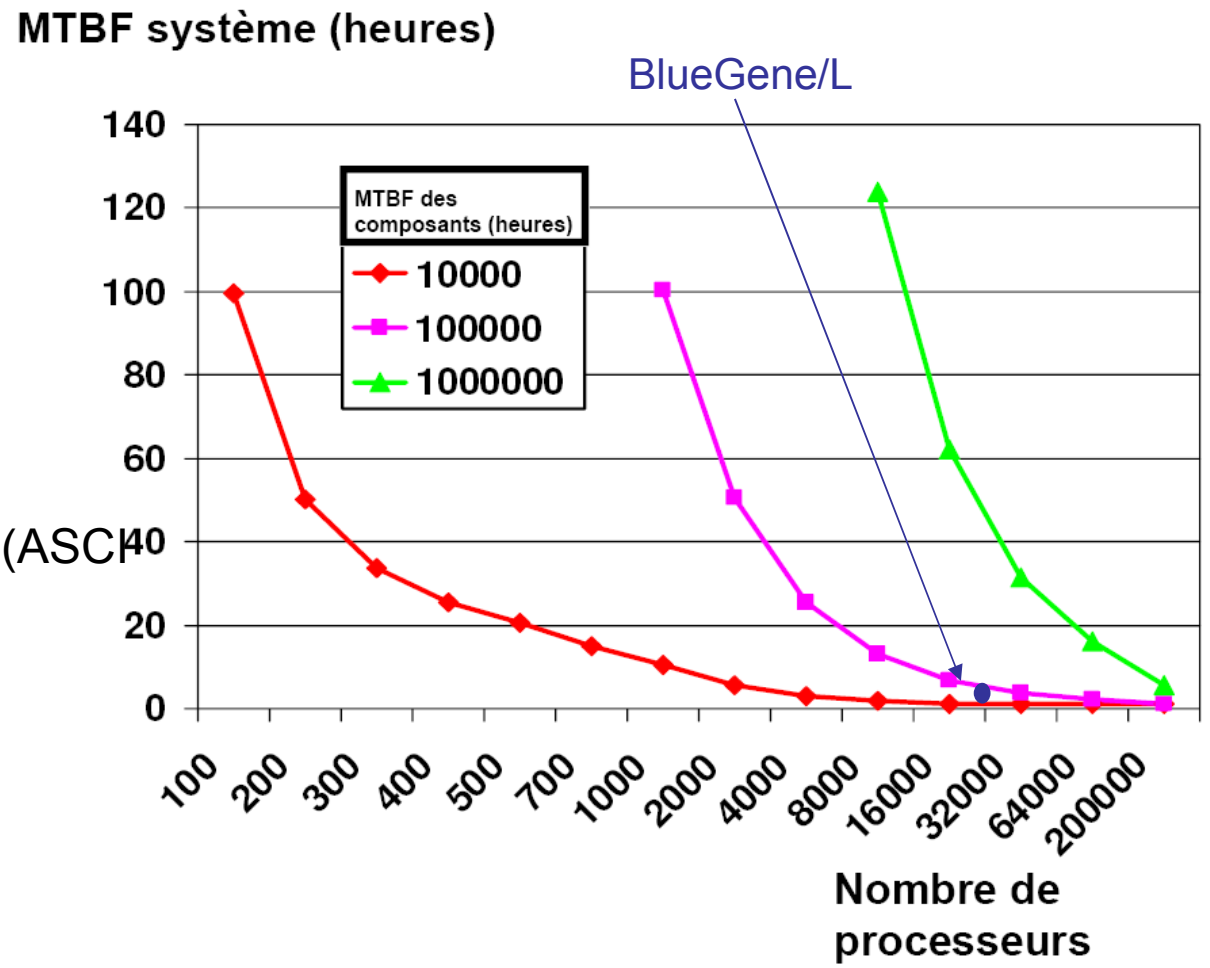
Calcul hautes performances
Évolution du MTBF dans le futur

1 PetaFlop =

200k 5Gflop CPU

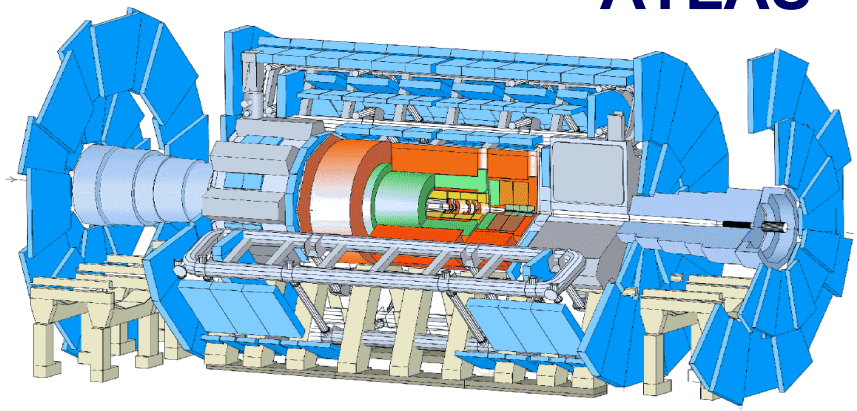
Avec du matériel fiable actuel (ASCI white), une machine de cette dimension subit

1 défaillance par heure

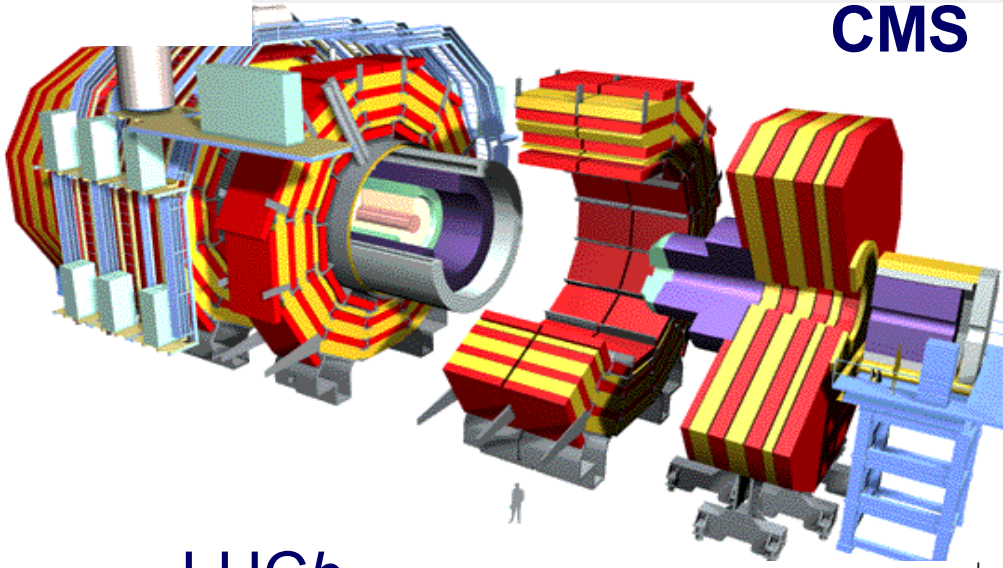


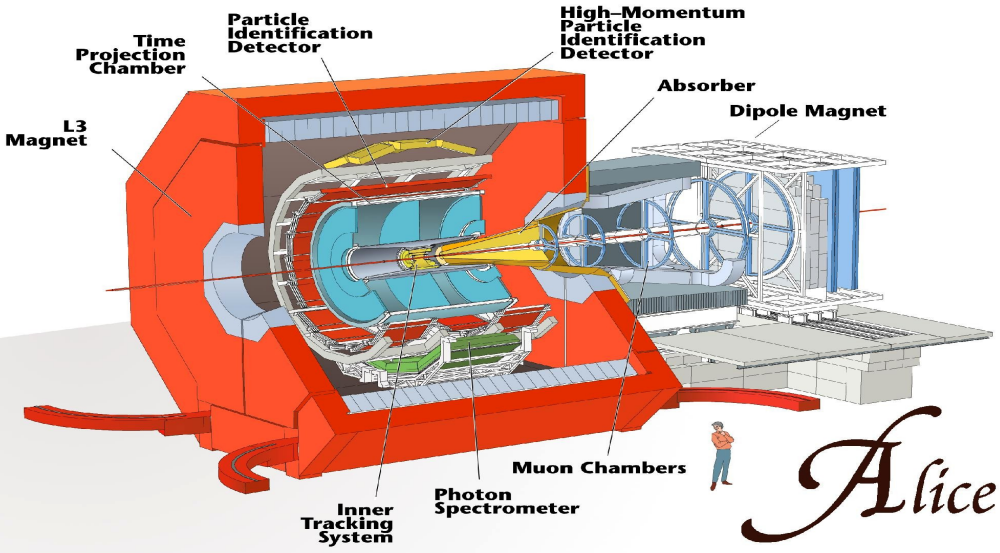
The Large Hadron Collider Project
4 detectors

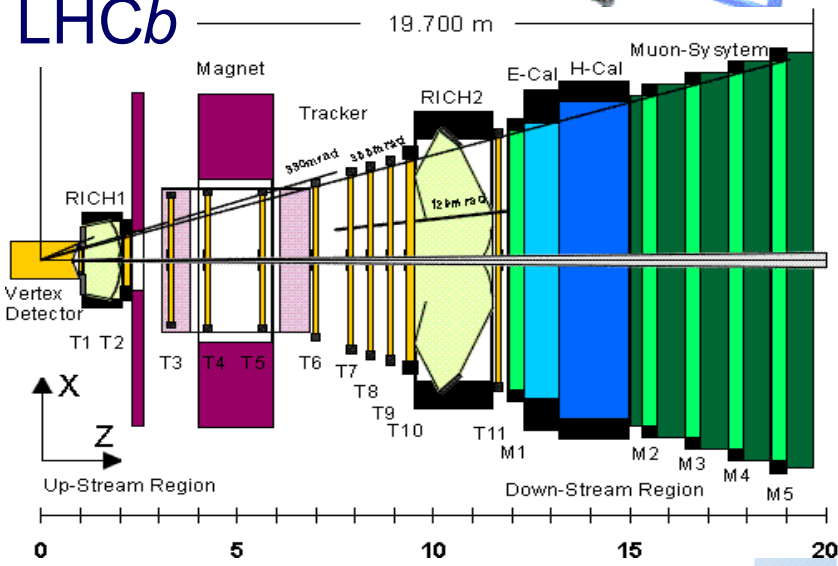
ATLAS



CMS





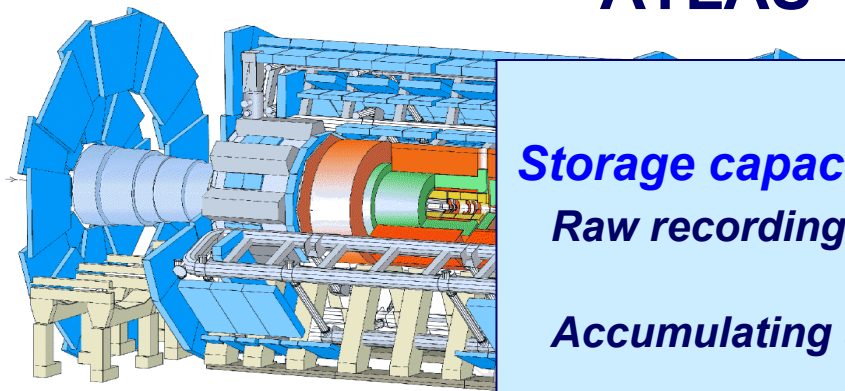


CERN

The Large Hadron Collider Project
4 detectors

ATLAS

CMS



Storage capacity–

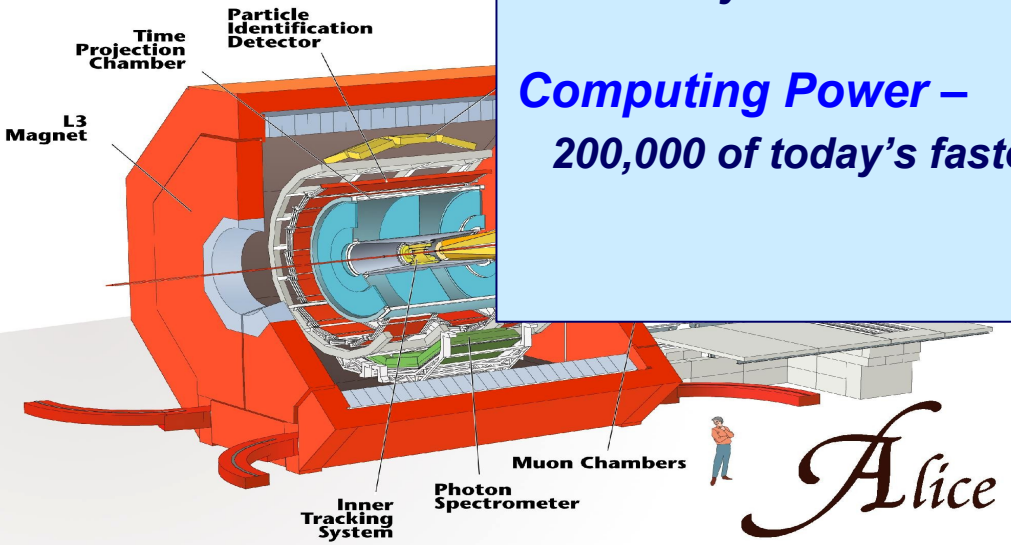
Raw recording rate 0.1 – 1 GBytes/sec

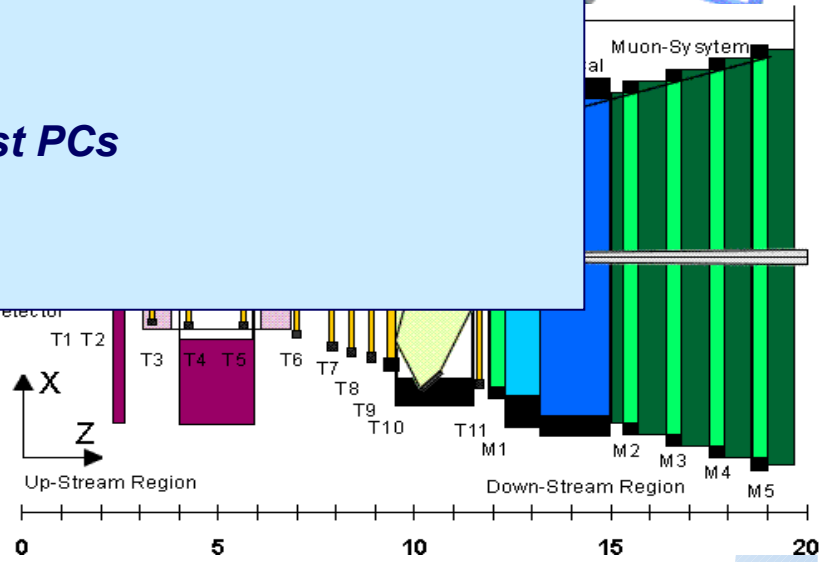
Accumulating at 5-8 PetaBytes/year

10 PetaBytes of disk

Computing Power –

200,000 of today's fastest PCs





CERN

Earthquake Hazard Assessment

2001 Gujarati (M 7.7) Earthquake, India

Use parallel computing to simulate earthquakes

Learn about structure of the Earth based upon seismic waves (tomography)

Produce seismic hazard maps (local/regional scale) e.g. Los Angeles, Tokyo, Mexico City, Seattle

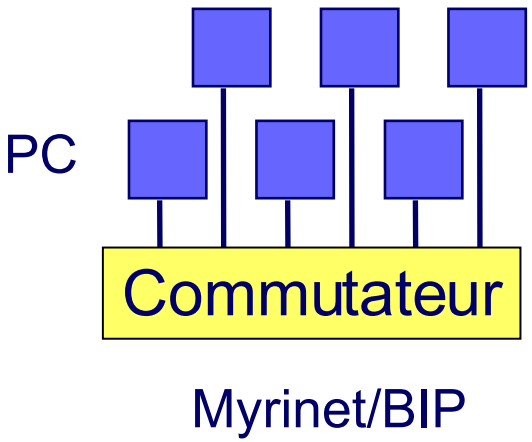
20,000 people killed
167,000 injured
≈ 339,000 buildings destroyed
783,000 buildings damaged

What is a cluster?

- 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.
- A typical cluster:
  - Network: Faster, closer connection than a typical network (LAN)
  - Low latency communication protocols
  - Looser connection than SMP
- Cluster 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

Cluster computing

- Architecture homogène, faiblement hétérogène
  - Grappes (Cluster, COW), machines //
    - PC, stations de travail
    - SCI, Myrinet, Giganet, MPC, ...
  - Protocoles de communication
    - BIP, SISCI, SciOS, VIA, TCP, UDP, ...



81.6 Gflops (216 nodes) + top 500 (385) June 2001
http://clic.mandrakesoft.com

HP Cluster : 104 HPBi Itanium 2

Example Clusters: Berkeley NOW

- 100 Sun UltraSparcs
  - 200 disks
- Myrinet SAN
  - 160 MB/s
- Fast comm.
  - AM, MPI, ...
- Ether/ATM switched external net
- Global OS
- Self Config

Motivation for using Clusters

- The communications bandwidth between workstations is increasing as new networking technologies and protocols are implemented in LANs and WANs.
- Workstation clusters are easier to integrate into existing networks than special parallel computers.

Motivation for using Clusters

- Surveys show utilisation of CPU cycles of desktop workstations is typically <10%.
- Performance of workstations and PCs is rapidly improving
- As performance grows, percent utilisation will decrease even further!
- Organisations are reluctant to buy large supercomputers, due to the large expense and short useful life span.

Motivation for using Clusters

- 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 clusters are a cheap and readily available alternative to specialised High Performance Computing (HPC) platforms.

- Use of clusters of workstations as a distributed compute resource is very cost effective - incremental growth of system!!!

Cluster Computing - Research Projects

- OAR/ (Grenoble) - France
- Beowulf (CalTech and NASA) - USA
- CCS (Computing Centre Software) - Paderborn, Germany
- Condor - Wisconsin State University, USA
- DQS (Distributed Queuing System) - Florida State University, US.
- EASY - Argonne National Lab, USA
- HPVM -(High Performance Virtual Machine),UIUC&UCSB,US
- MOSIX - Hebrew University of Jerusalem, Israel
- MPI (MPI Forum, MPICH is one of the popular implementations)
- NOW (Network of Workstations) - Berkeley, USA
- NetSolve - University of Tennessee, USA
- **PM²** (Lyon, Lille, Bordeaux) - France

Cluster Computing - Commercial Software

- Codine (Computing in Distributed Network Environment) - GENIAS GmbH, Germany
- LoadLeveler - IBM Corp., USA
- LSF (Load Sharing Facility) - Platform Computing, Canada
- OpenFrame - Centre for Development of Advanced Computing, India
- RWPC (Real World Computing Partnership), Japan
- Unixware (SCO-Santa Cruz Operations), USA
- ClusterTools (A number for free HPC clusters tools from Sun)
- A number of commercial vendors worldwide are offering clustering solutions including IBM, HP-Compaq, Microsoft, a number of startups like TurboLinux, HPTI, Scali, BlackStone…..)

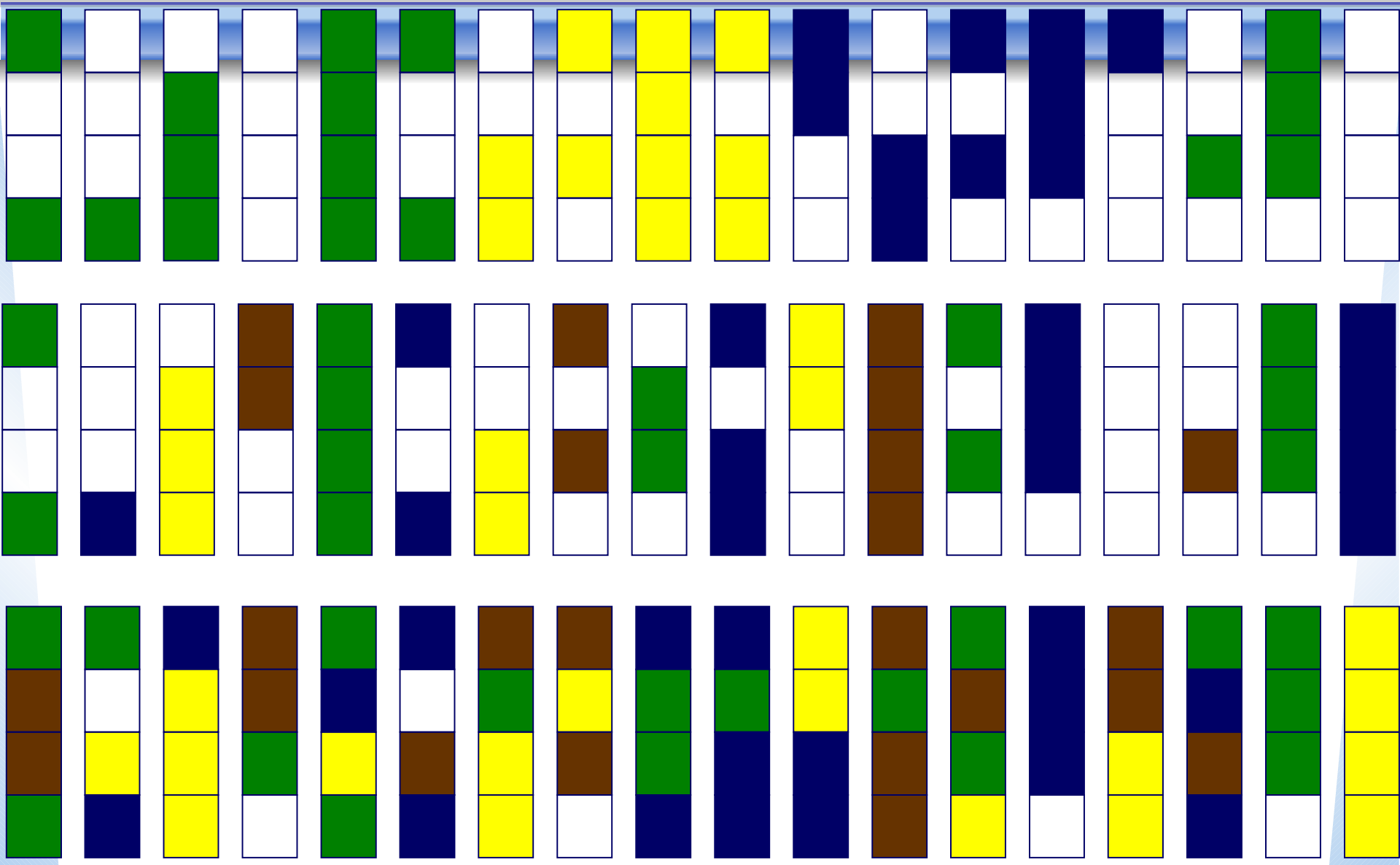
Cluster Components...1a Nodes

- Multiple High Performance Components:
  - PCs
  - Workstations
  - SMPs (CLUMPS)
- Cluster is mainly homogeneous
  - Node (processor, memory, cache, disk)
  - Operating System
  - Network

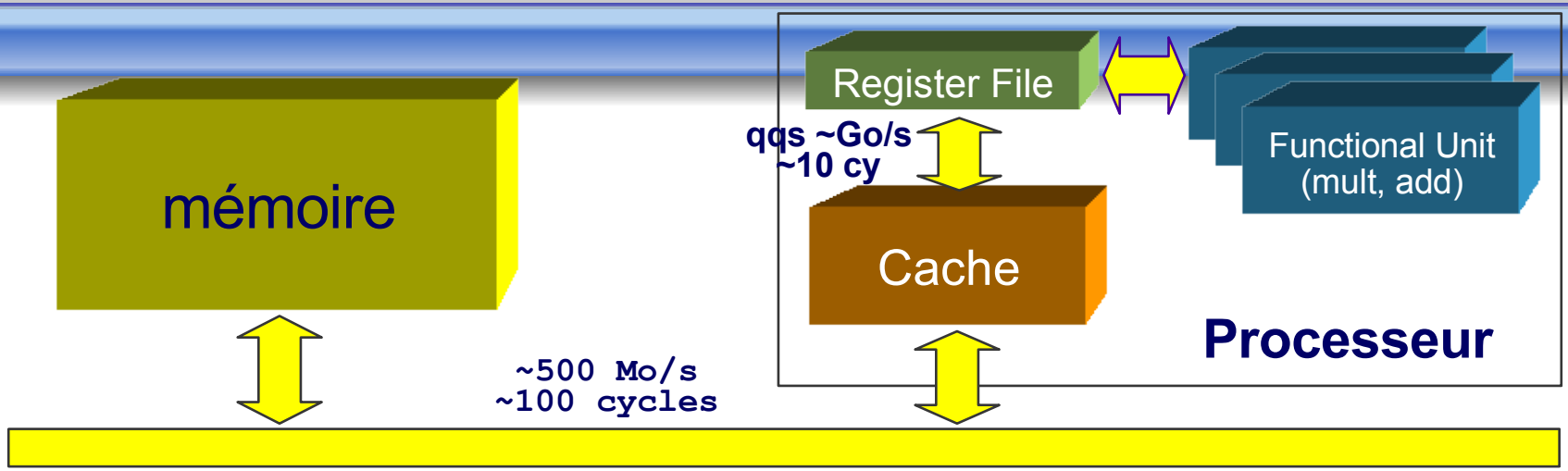
Cluster Components...1bProcessors

- There are many (CISC/RISC/VLIW/Vector..)
  - Intel: Pentiums, Xeon, Merceed….
  - Sun: SPARC, ULTRASPARC
  - HP PA
  - IBM RS6000/PowerPC
  - SGI MIPS
  - Digital Alphas
- Integrate Memory, processing and networking into a single chip
  - IRAM (CPU & Mem): (http://iram.cs.berkeley.edu)
  - Alpha 21366 (CPU, Memory Controller, NI)

Hyperthreading, SMT, NUMA

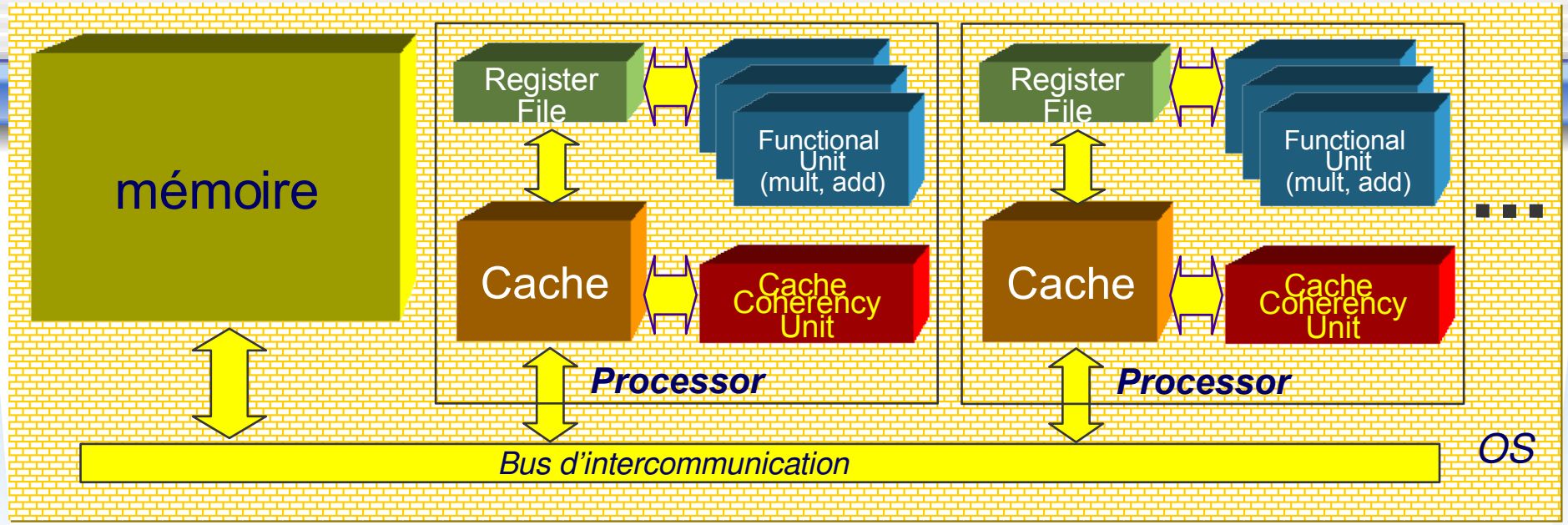


Architecture Scalaire



- Reduced Instruction Set (RISC) Architecture:
  - Les instructions load/store font référence à la mémoire
  - Les unités fonctionnelles travaillent sur des données stockées dans les registres
  - Hiérarchie mémoire dans une architecture scalaire :
    - Les éléments utilisés récemment sont copiés dans le cache,
    - Les accès au cache sont plus rapides que les accès à la mémoire.

Architecture SMP UMA



Protocole de cohérence cache

- **Plusieurs processeurs modifient des éléments de la même ligne de cache**

Mémoire centrale et E/S

- **Partagées par tous les processeurs**

Modèle de programmation

- **Extension du modèle de programmation monoprocesseur**

Bus d'interconnexion entre la mémoire et les processeurs

Faible nombre de processeurs

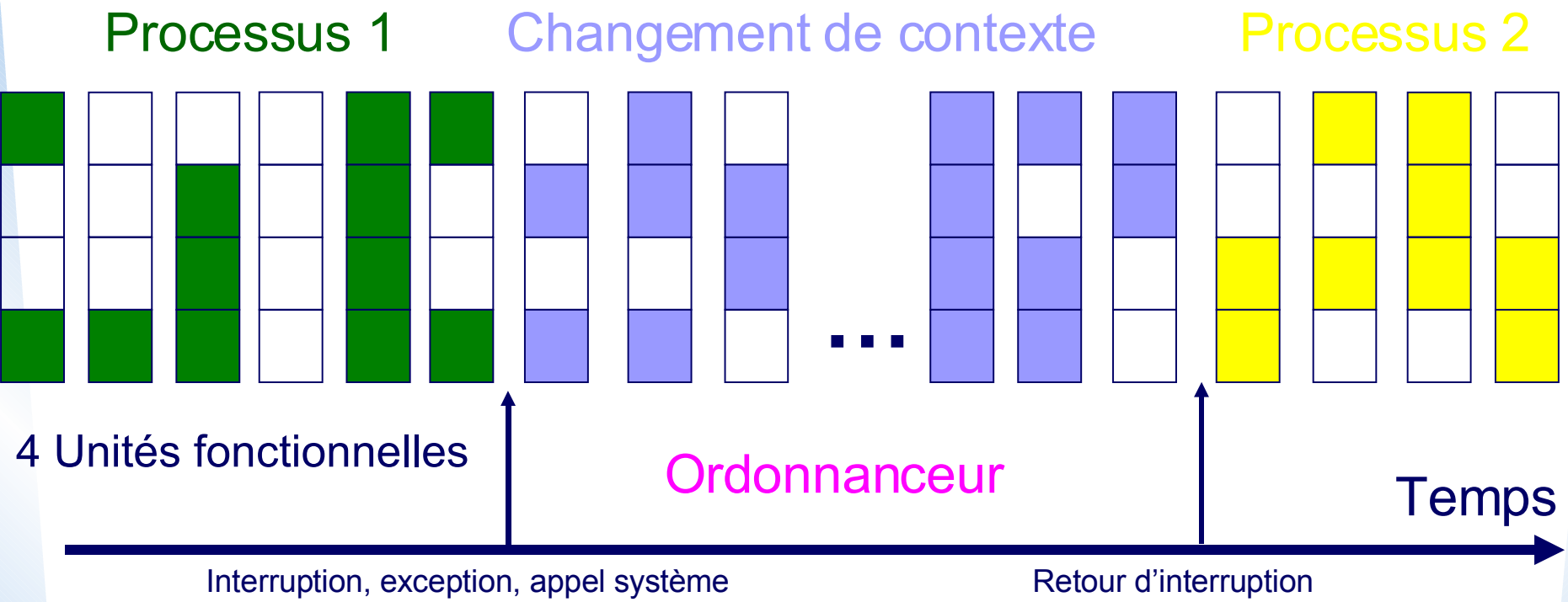
Programmation Mono-Threadée

- Exécution séquentielle d'un programme
  - Instruction par instruction
  - Instructions: Calcul, Mémoire, Branchement, appel de procédure,…
- Processus, processus Lourd
  - Structuration des systèmes d'exploitation
  - Multi-programmation, temps-partagé
- Caractéristiques
  - Entité active directement supportée par l'OS
    - Flot d'exécution
    - Pile des contextes de procédure
    - Espace d'adressage privé
    - Ressources systèmes
- Coût de gestion élevé
  - Allocation des ressources (mémoire,…)
  - Appels systèmes (Fork, exec, …)

Instruction Level Parallelism

- Programme séquentiel
  - N'y aurait-il pas des instructions indépendante qui pourraient être exécutées en parallèle?
- Comment générer de l'ILP?
  - Pipe-line du processeur
    - Recouvrement d'exécution d'instructions
    - Limité par la divisibilité de l'instruction
  - Superscalaire
    - Plusieurs unités fonctionnelles
    - Limité par le parallélisme intrinsèque du programme seq.
- Comment accroître l'ILP?
  - Prédictions sur les branchements conditionnels
  - Réordonnancer les instructions (Out of Order Execution)
- Recherche
  - Domaines de l'Architecture-compilation

Processeur SuperScalaire



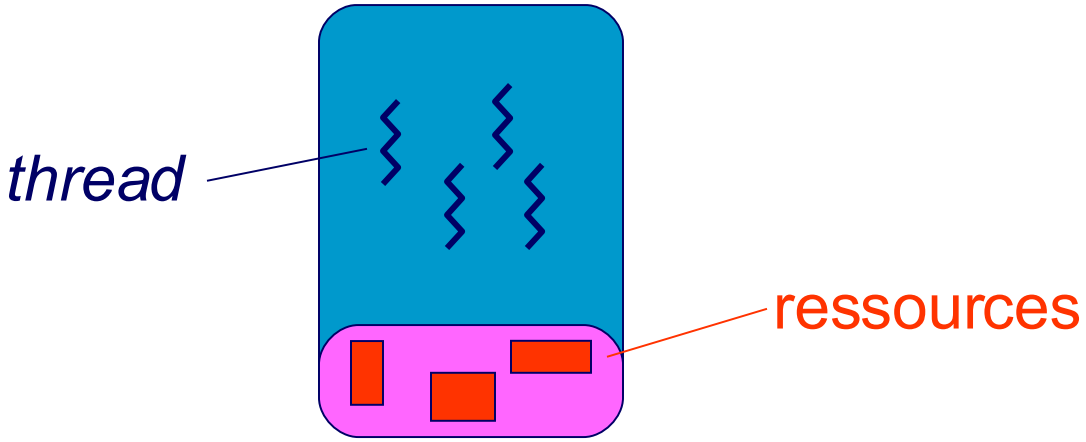
- Réduire le temps des changements de contexte (cases grisées-bleues)
- Accroître l'utilisation des unités d'exécution (cases blanches)

Processus Légers/Threads

- Objectifs
  - Mener plusieurs activités indépendantes au sein d'un processus
  - Exploitation des architectures SMP
  - Améliorer l'utilisation du processeurs (context-switch)
- Exemples
  - Simulations
  - Serveurs de fichiers
  - Systèmes d'exploitation (!)
- Solution sans l'aide du multithreading
  - Automate à états finis implanté « à la main » (sauvegardes d'états)

Les processus légers

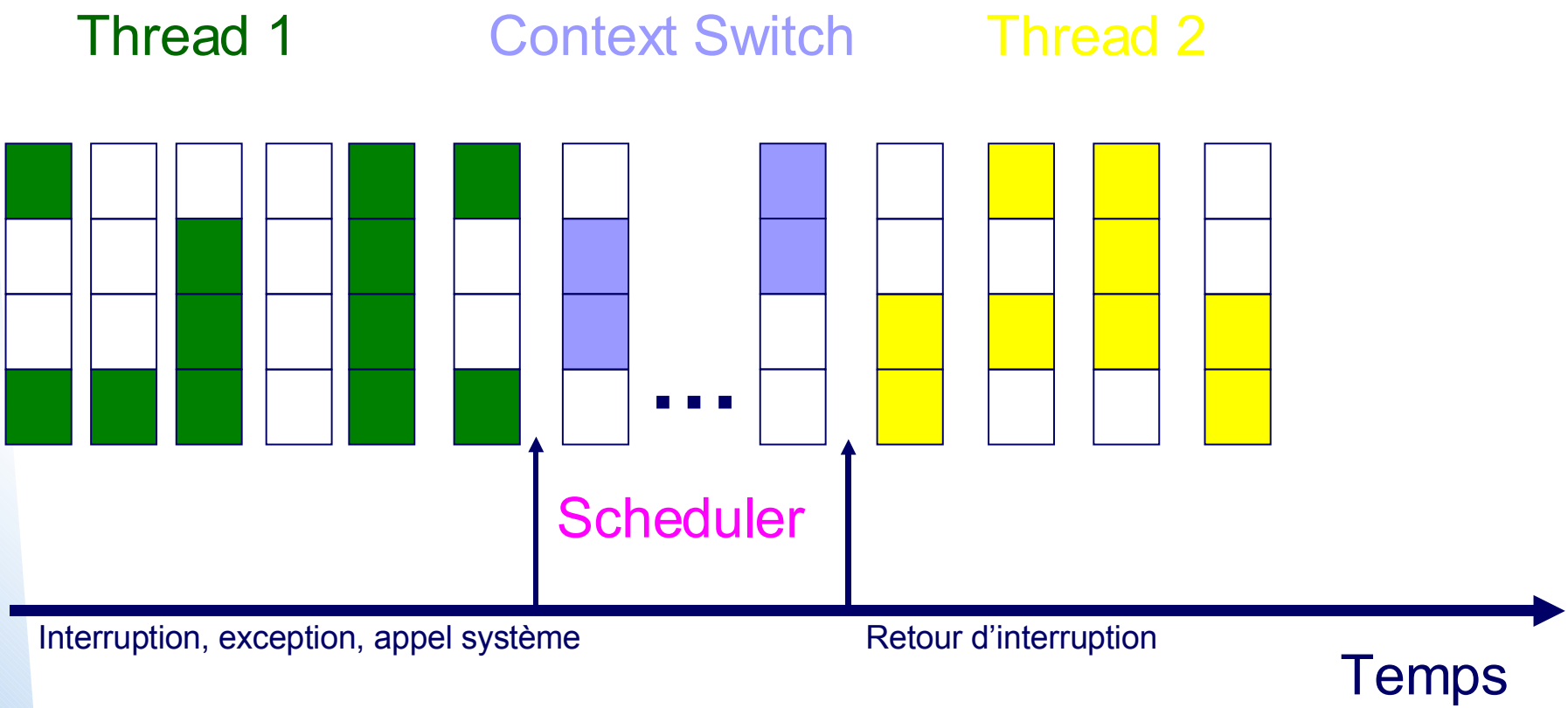
- Principe
  - Détacher flot d'exécution et ressources



- Introduits dans divers langages & systèmes
  - Programmation concurrente
  - Recouvrement des E/S

Multithreading et Processeur SS

Processeur SuperScalaire



Hyperthreading

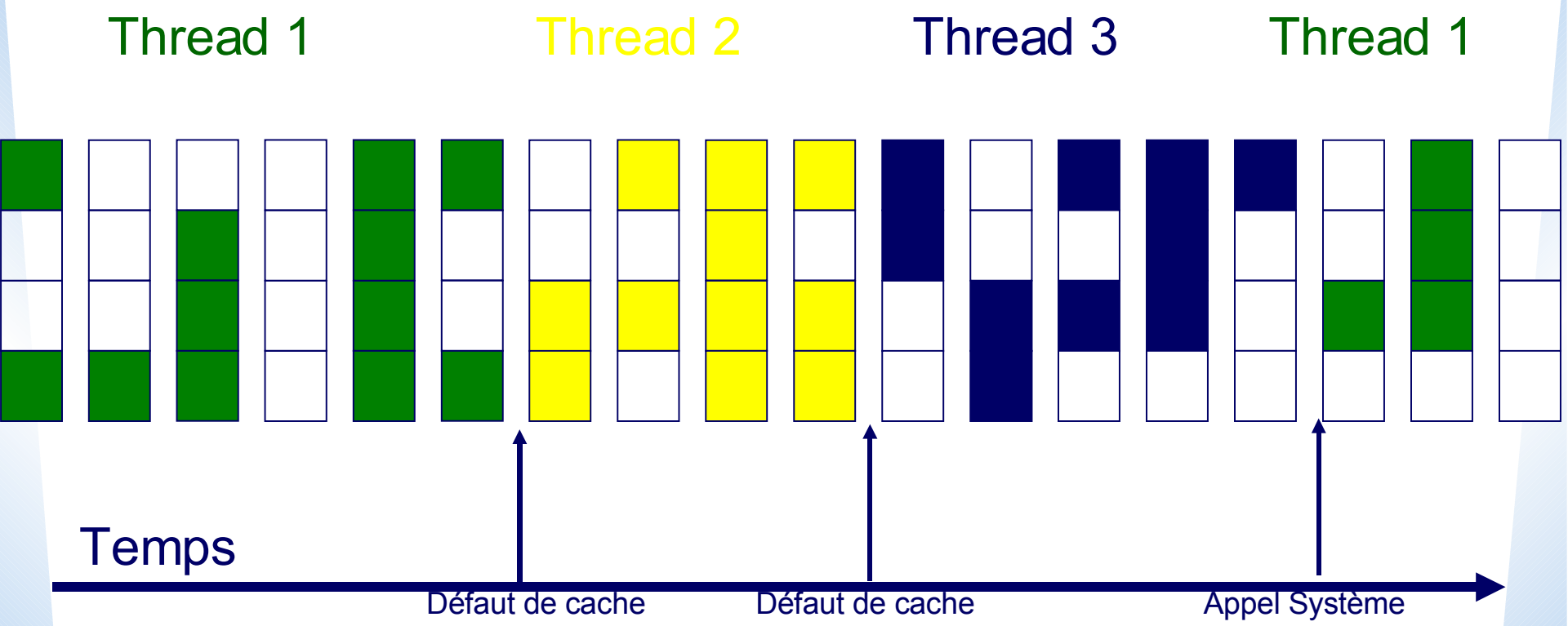
Evolution des architectures de processeurs

Phase 1: Processeurs Multithreadés

- Modification de l'architecture du processeur
  - Incorporer au processeur deux (ou plus) jeux de registres pour les contextes des threads
    - Registres généraux
    - Program Counter (PC), registre d'instruction
    - Process Status Word (PSW), registre d'état
  - A tout instant, un thread et son contexte sont actifs
  - Changement du contexte courant instantané
    - Appel système, IT
    - Défaut de cache
- Processeur IBM PowerPC RS 64
  - Recherche, non commercialisé
- Processeur Intel Xeon Hyperthreading
  - Serveur Bi-processeur Xeon Hyperthreadé
    - Vue de Linux ou Windows, 4 processeurs

Phase 1: Processeurs Multithreadés (2)

Multithread à Gros Grain (Coarse-Grained Multi-threaded)



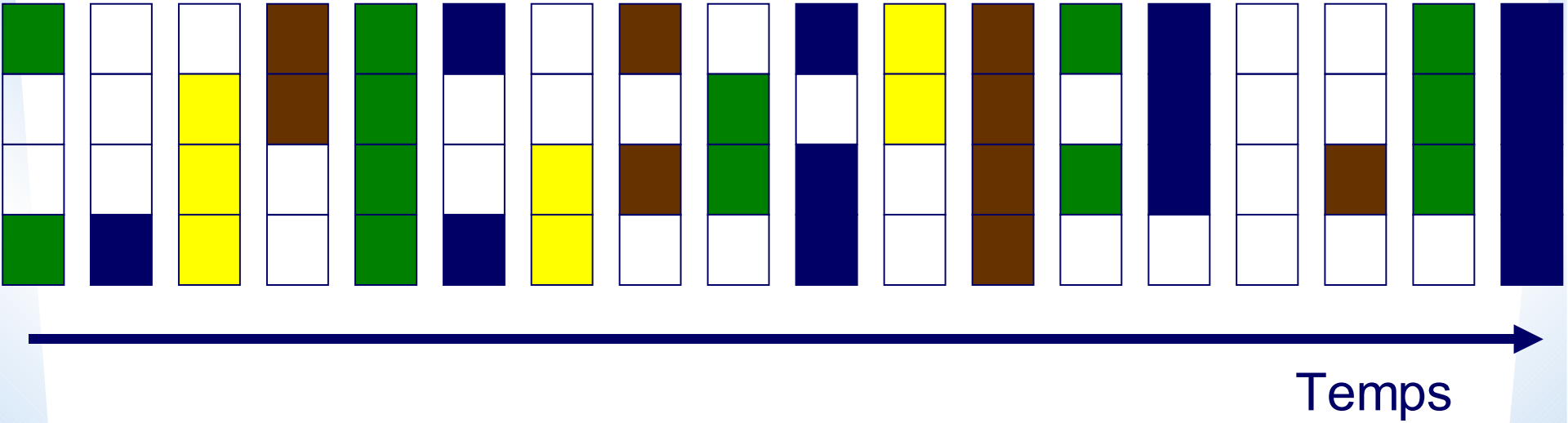
Phase 2: Processeurs Multithreads (1)

- Modification de l'architecture du processeur
  - Incorporer au processeur N jeux de registres pour les contextes des threads
    - Registres généraux
    - Program Counter (PC), registre d'instruction
    - Process Status Word (PSW), registre d'état
  - A tout instant, un thread et son contexte sont actifs
  - Changement de contexte à chaque cycle
    - Chaque thread dispose de son ratio du processeur (1/N)
- Processeur TERA

Phase 2: Processeurs Multithreadés (2)

Multithread à Grain Fin (Fine-Grained Multi-threaded)

4 registres de threads : ¼ temps processeur par thread



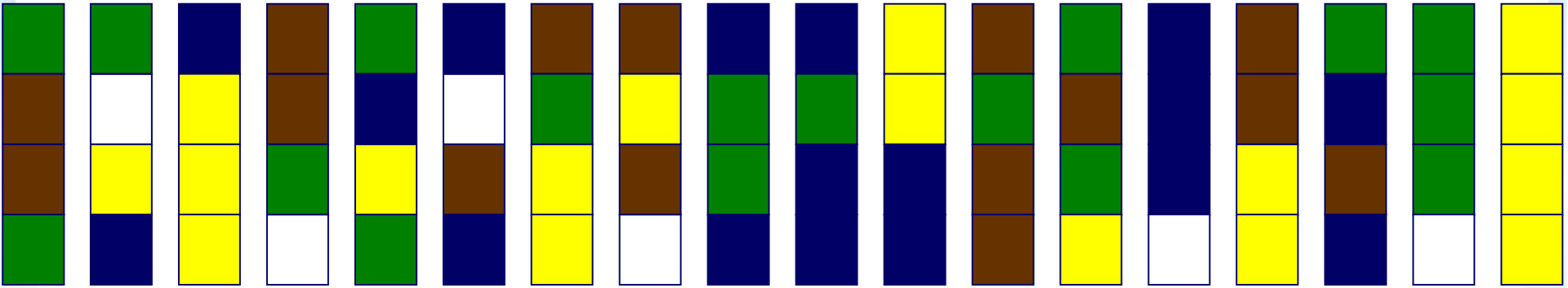
Temps

Thread 1 Thread 2 Thread 3 Thread 4

Processeurs HyperThreadés (1)

- Modification de l'architecture du processeur
  - Incorporer au processeur N jeux de registres pour les contextes des threads
    - Registres généraux
    - Program Counter (PC), registre d'instruction
    - Process Status Word (PSW), registre d'état
  - A un instant donné, les unités du processeurs peuvent être partagées entre plusieurs threads
- Eviter de stresser les mêmes ressources
  - Conflit d'accès à certaines unités d'exécution
- Processeur ALPHA EV8

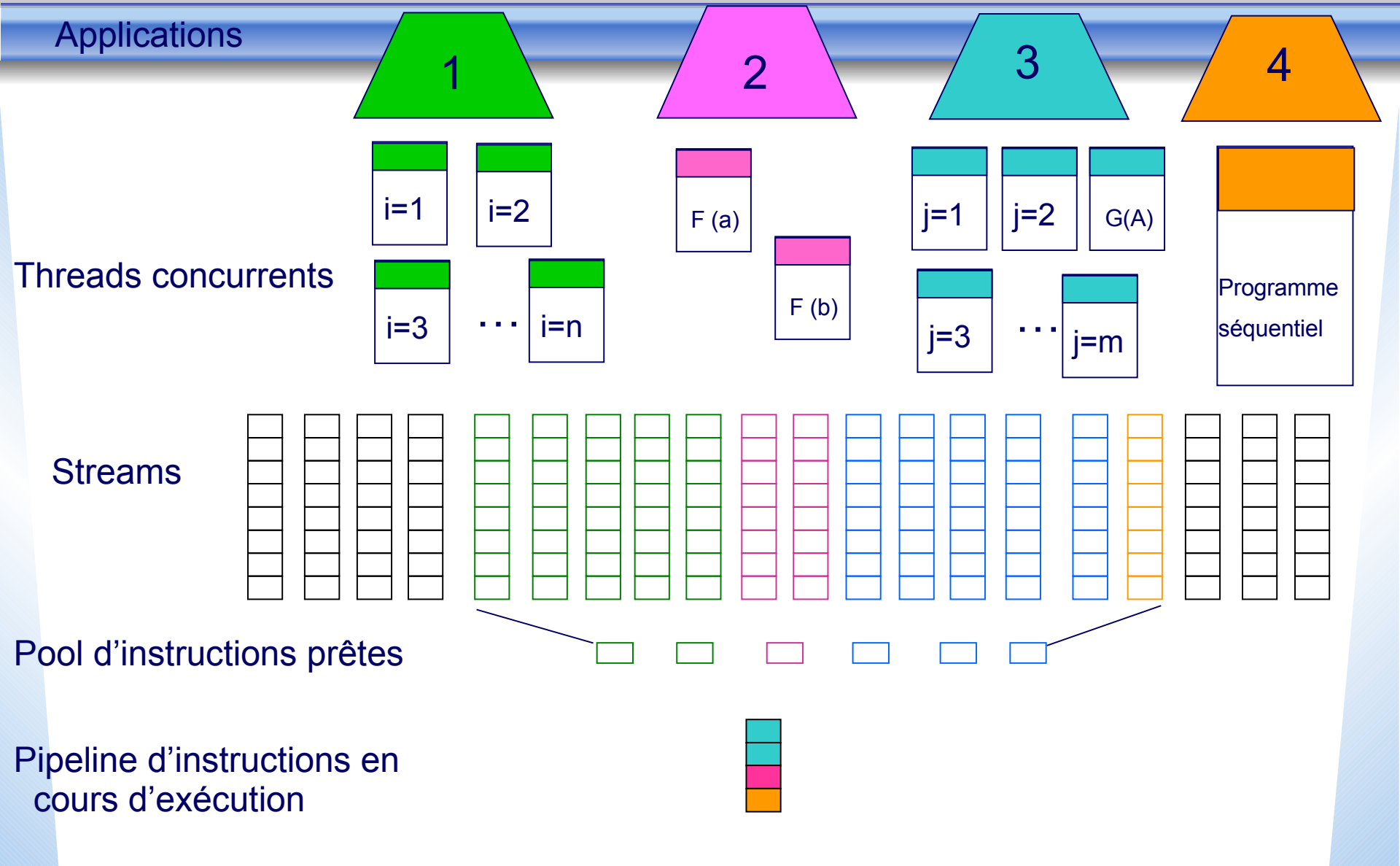
Processeurs Hyper-Threadés (2)



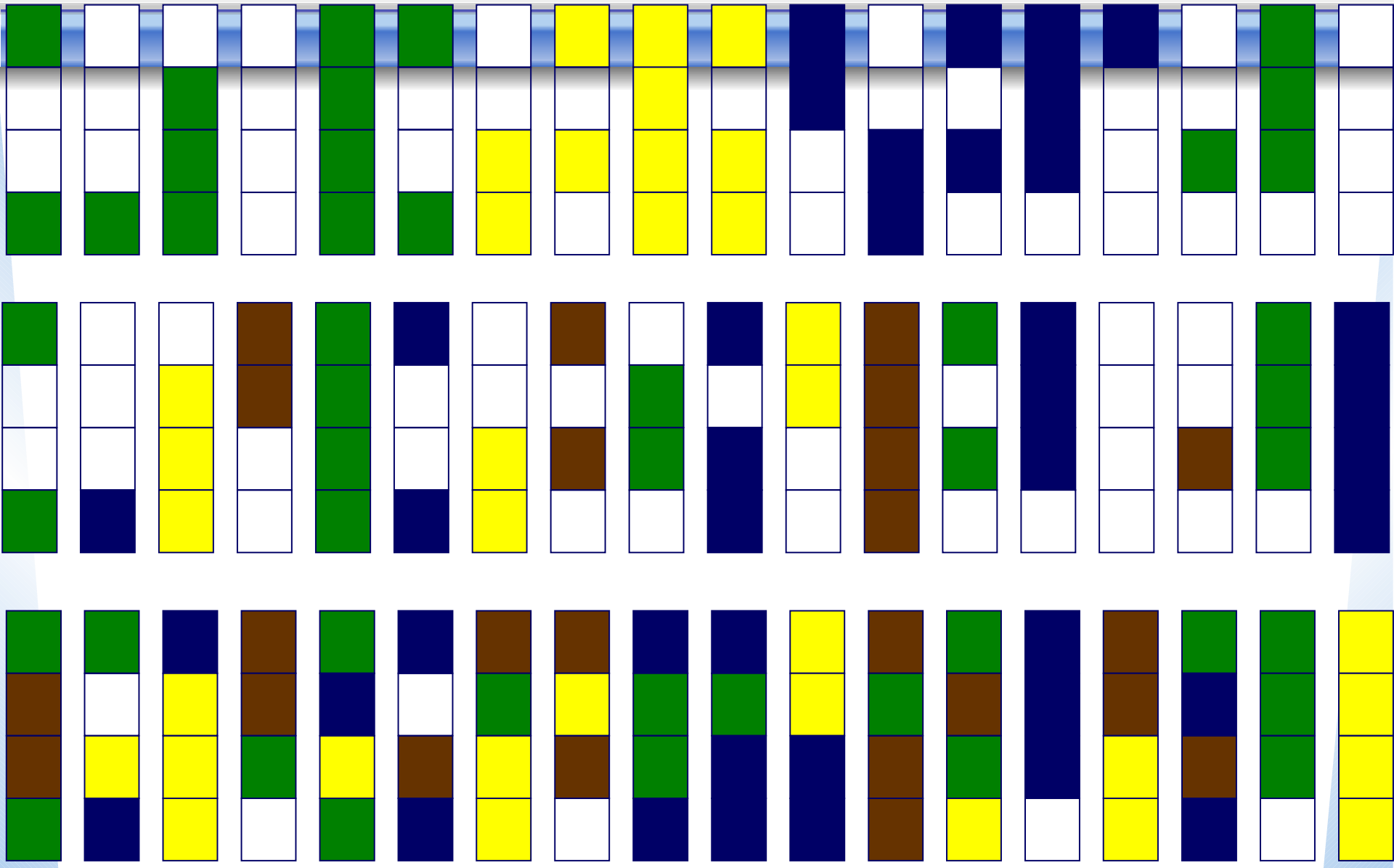
Thread 1 Thread 2 Thread 3 Thread 4

Temps

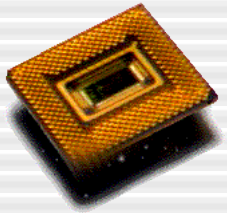
Architecture processeur Cray-TERA



Résumé



What is Alpha?



1. A microprocessor, a chip, a cpu: EV4, EV5, EV6, EV7, EV8 . . .

digital

2. A 64-bit RISC architecture introduced by Digital in ’92

COMPAQ

3. Acquired by Compaq in ’98; continued investment

4. ~~CPU for future Non-Stop Himalaya systems (EV7)~~
   Cancelled in Sept 2001

TANDEM NSK

5. Designed and engineered by Compaq;
   outsourced manufacturing (fabrication);
   licensed to API, sold to other system vendors

SAMSUNG ELECTRONICS

IBM Microelectronics

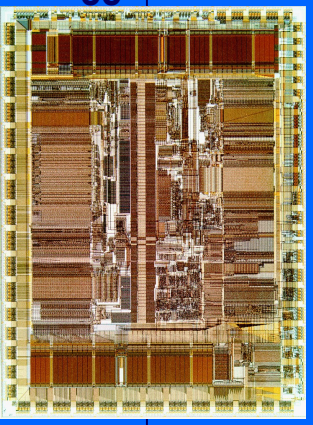
Api

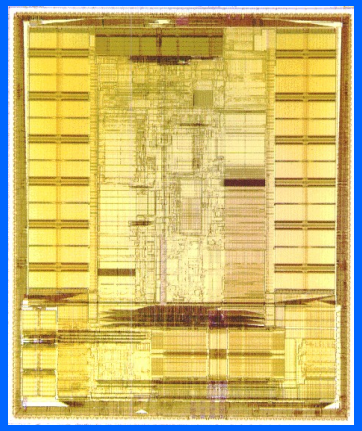
Alpha chip

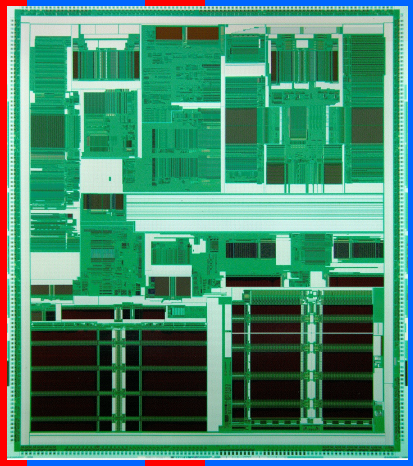
SPECint95

Source: Standard Performance Evaluation Corporation
SPEC - November, 2000

"Alpha has nearly continuously excelled all comers since 1992, an eternity in this industry."
Illuminata, Inc. March 1999







50
45
40
35
15
10
5

2.95
5.18
7.43
10.1
13.6
15.0
18.8
30.3
40.1
50

EV68
EV67
EV6

.75
.50µm
.35µm
.25µm
.18µm

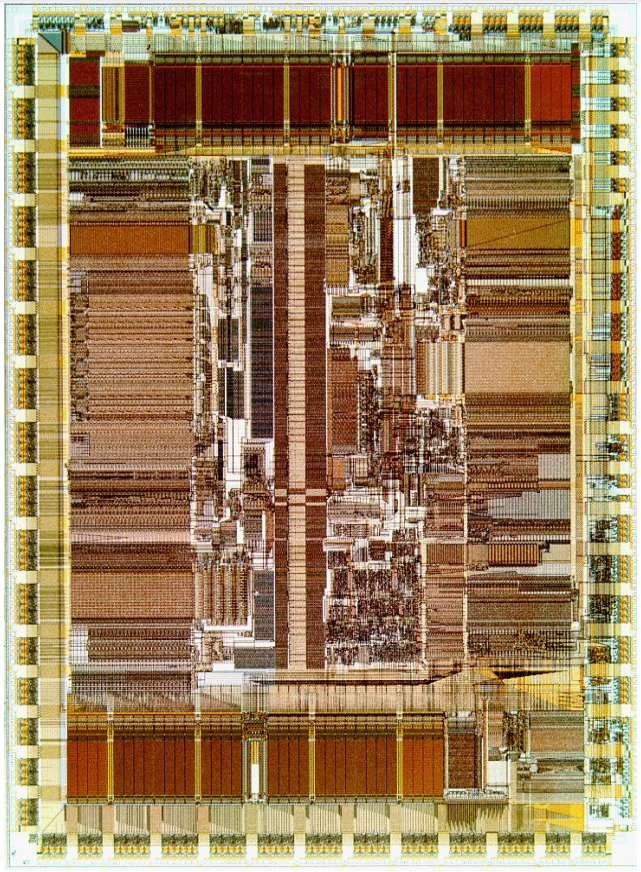
1992
1995
1997
1999
2000
2001
2002

166
266
300
350
440
500
600/625
525
667
833 MHz
1250

1.5M
1.5M
9.3M
9.3M
15 M transistors

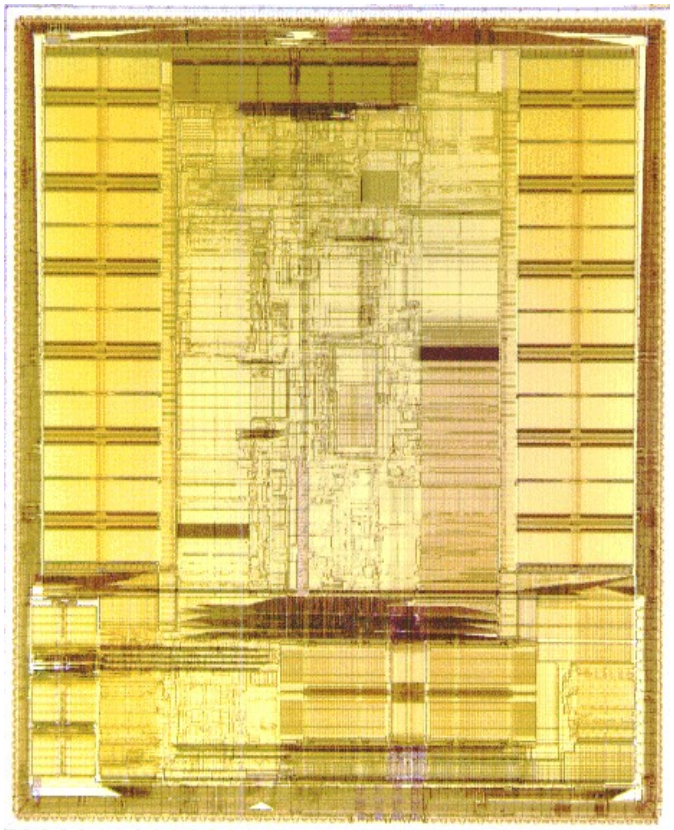
EV4
EV45
EV5
EV56
EV6 EV67 EV68

EV4 Chip Overview



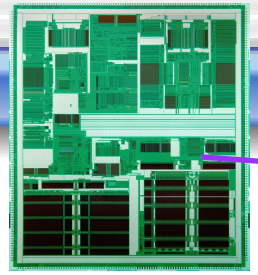
- 0.75µm 3LM CMOS, 3.3V
- 200MHz @30W
- 233 mm^2 ,1.7M, 431-PGA
- Dual in-order issue
- Fully pipelined
- 8kB I-Cache, 8kB D-Cache
- 32 64b I-Regs, 32 64b FP-Regs
- 1-bit branch prediction
- Shared L2, system interface

EV5 Chip Overview



- 0.50µm 4LM CMOS, 3.3V
- 350MHz @60W
- 298 mm^2, 9.3M, 499-PGA
- Quad in-order issue
- FP latencies reduced 2 cycles
- 8kB I-Cache, 8kB D-Cache
- 96kB unified on-chip L2 cache
- 2-bit branch prediction
- Non-blocking cache scheme

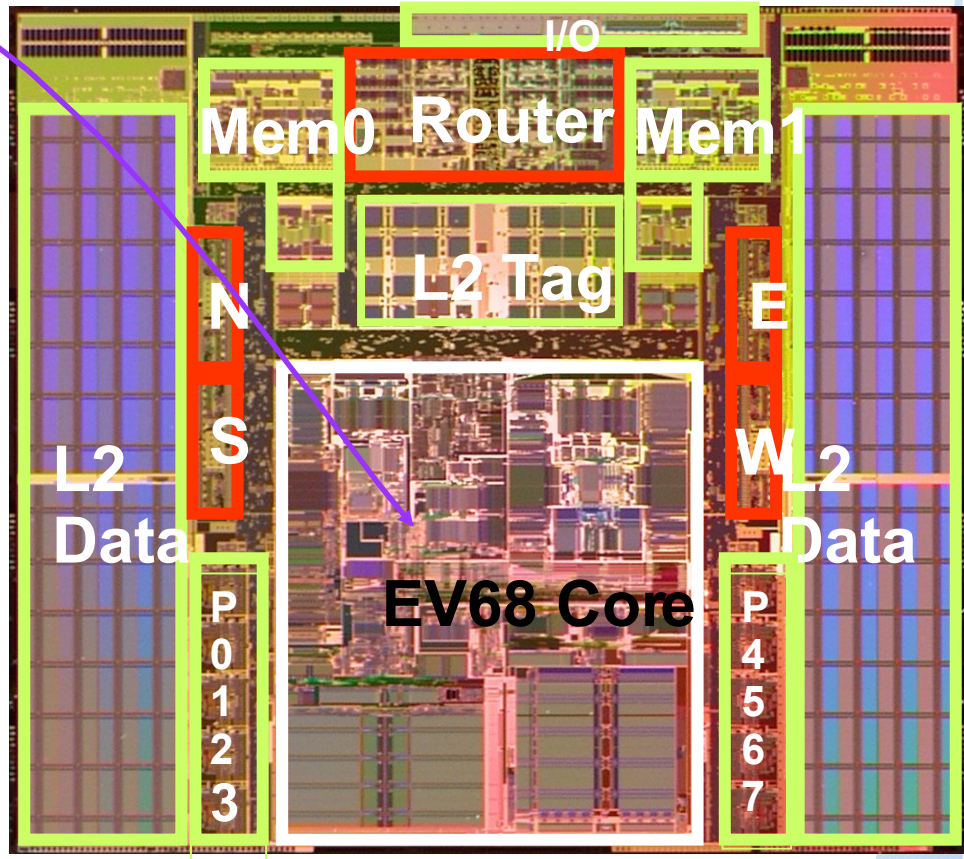
EV7 – The System is the Silicon



SMP CPU interconnect was external logic..

Now it's on the chip !

- 21264 (EV68) core with enhancements
- Integrated L2 cache
  - 1.75 MB (ECC)
  - 20 GB/s bandwidth
- Integrated memory controllers
  - Direct RAMbus (ECC)
  - 12 GB/s bandwidth
  - Optional RAID in memory
- Integrated network interface
  - Direct processor-processor interconnects
  - 4 links - 25.6 GB/s aggregate bandwidth
  - ECC (single error correct, double error detect)
  - 3.2 GB/s I/O interface per processor



Alpha systems family

servers and
workstations

AlphaServer sc Series

1000s processors

gs320
32-way

gs160
16-way

AlphaServer gs80

AlphaServer gs series

AlphaServer
es40, es45 8-way
1- 4 Processors

AlphaServer
ds10, ds10L,
ds20E, ds20L
1- 2 Processors

Processeur Itanium2 ® Intel ®

◆Architecture Itanium2 :

4ème génération de processeurs 64 bits Intel : Itanium2 (Madison9M)

EPIC: Explicit Parallel Instruction Computing

Fréquence : 1.5 Ghz

Puissance crête : 6 Gflops/s

- 1500 MHz * 2 madd/cycle → 6 GFLOPS

◆Intel Itanium2 :

L1I : 16ko; 64o/line ; 4 way

L1D : write through; 16ko; 1/- cycle; 64o/line ; 4 way ; (2ld&2st)/cycle

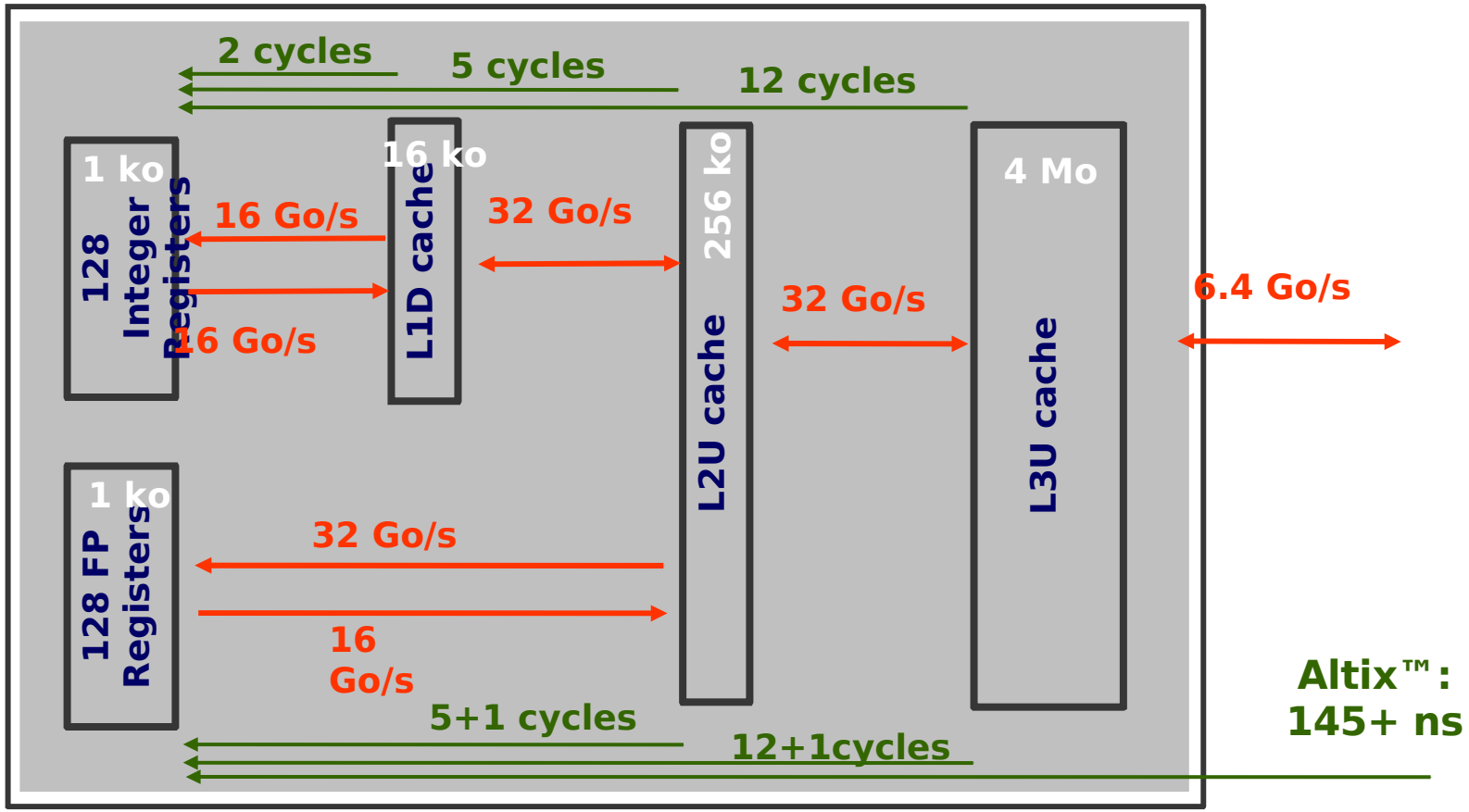
L2U : write back; 256ko; 5/6cycle; 128o/line; 8 way; (4ldf) | (2ldf[p]&2stf)

L3U : write back; 4Mo; 12/13cycle; 128o/line ; 24 way ; 48Go/s

Memory Front Side Bus (FSB) : 128o/line ; 6.4 Go/s

Processeur Itanium2 ® Intel ®

Débits et latences dans le processeur Itanium



Processeur Itanium2 ® Intel ® : Roadmap

4Gflops

Max.

2002

Itanium® 2

(McKinley)

- 900 Mhz, 3.6 Gflops, 1.5 Mo L3 cache
- 1 GHz, 4Gflops, 3 Mo L3 Cache)

>5Gflops

Max.

2003

Itanium® 2

(Madison)

- 1.5 GHz, 6Gflops, 6 Mo L3 Cache
- 1.3 Ghz, 5.2 Gflops, 3 Mo L3 cache

Low Power Itanium® 2 (Deerfield)

- 1.0 GHz, 4Gflops, 1.5Mo L3 Cache 62 Watts
- 1.4 Ghz, 4.6 Gflops, 1.5/3 Mo L3 cache

6.4Gflops

Fin 04

Itanium® 2

(Madison 9M)

-1.6 GHz, 6.4Gflops, 6/9 Mo L3 Cache

- 1.5 Ghz, 6 Gflops, 4 Mo L3 cache

Low Power Itanium® 2

Deerfield+ Processor

Follow-on

>16Gflops

2005

Montecito

(Dual Core on a Die)

Each Core

(>=2 GHz, >=8Gflops, 12Mo L3 Cache)

...

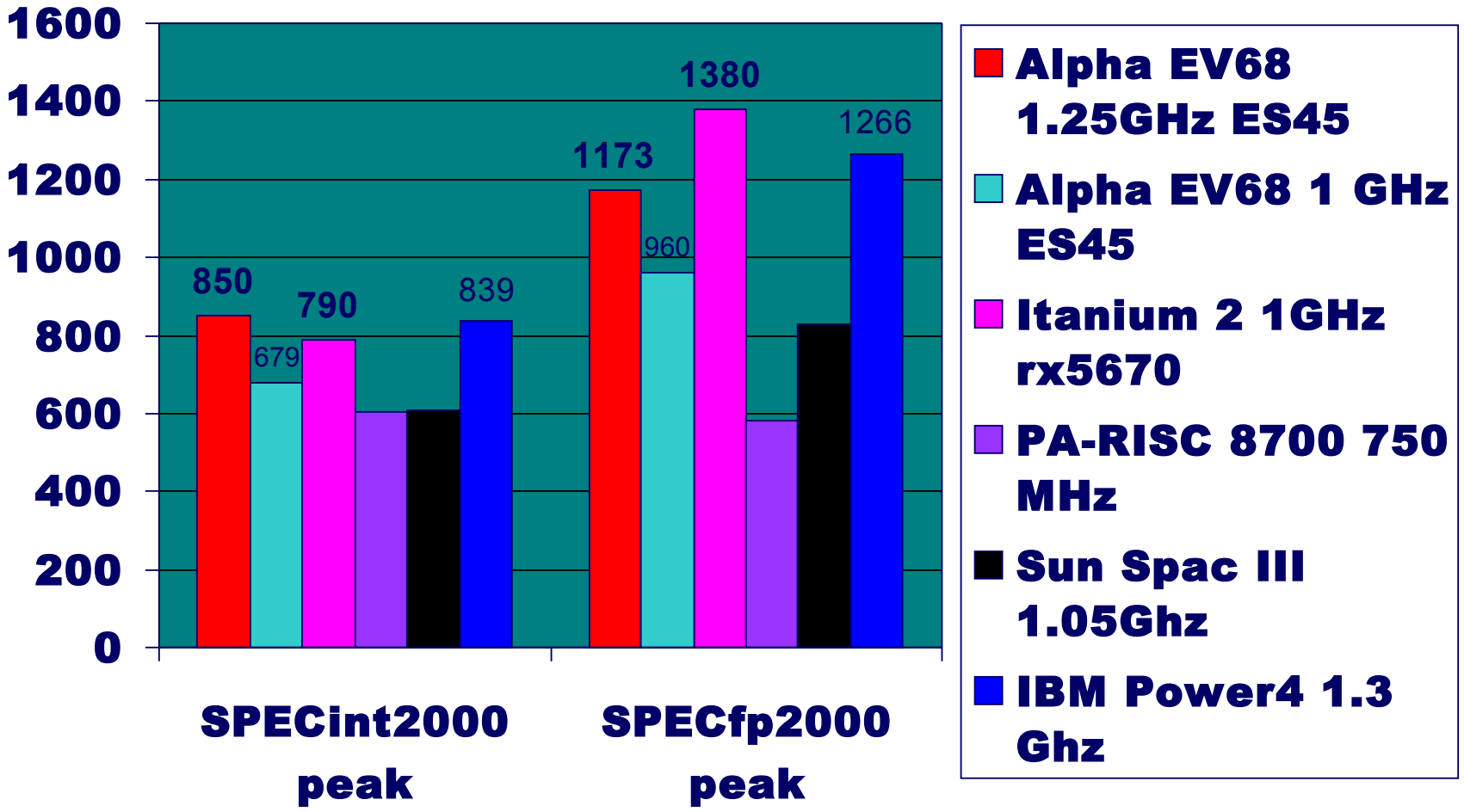
Low Power Montecito Dual Core Processor

Follow-on

Silicon Process

180 nm	130 nm	90 nm

Single CPU Performance Comparison

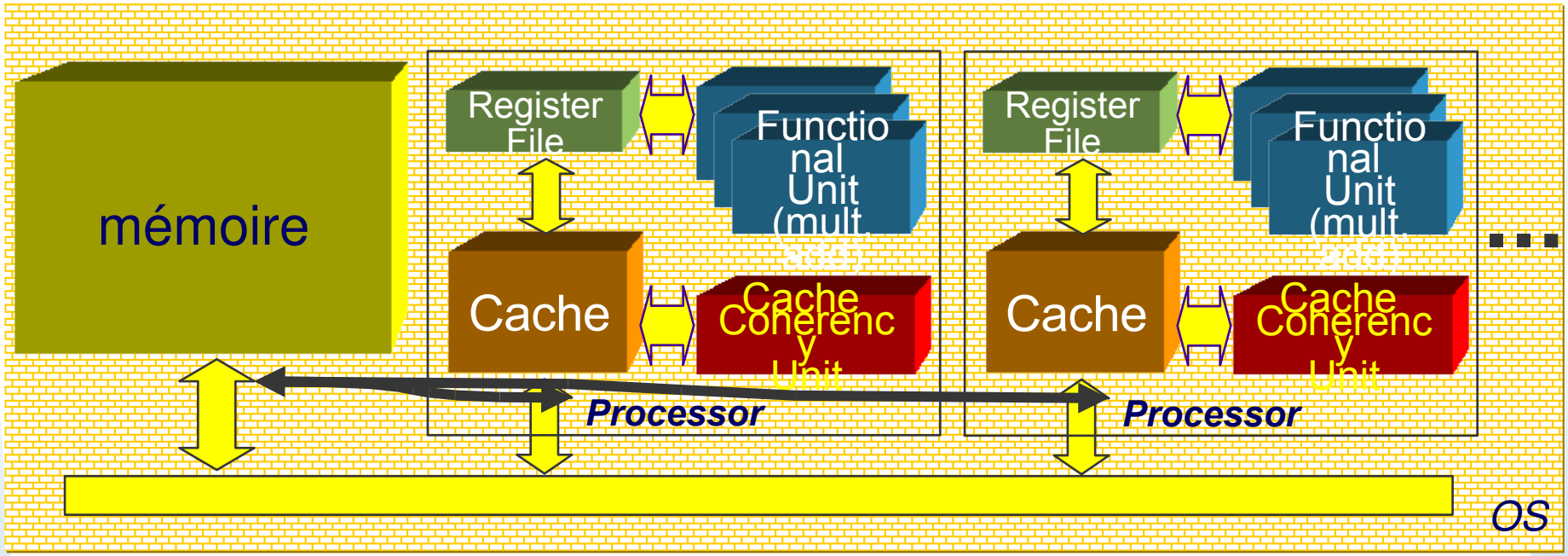


Problèmes des architectures UMA

Accès à la mémoire:

des niveaux de caches efficaces permettent d'économiser des références à la mémoire
pour les autres (non cachés), les accès concourants des processeurs à la mémoire partag

un *goulot d'étranglement*



Cette difficulté peut être levée avec les architectures à *mémoire distr*

Mémoire distribuée

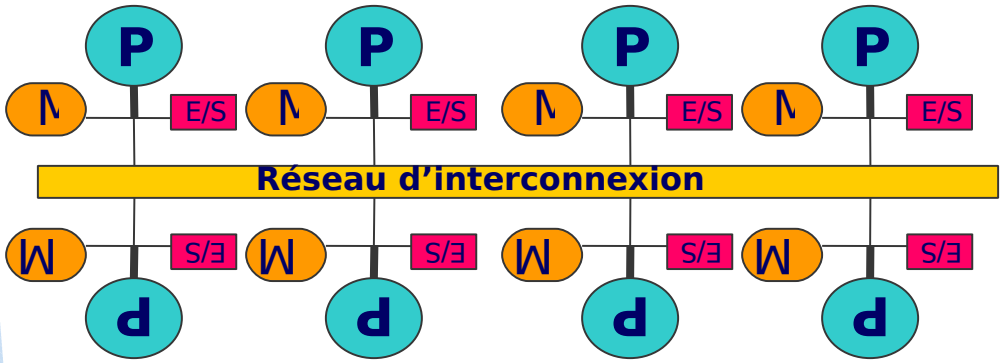
Technologie de masse

la bande passante globale mémoire-processeur est proportionnelle au nombre de processeurs

modèles d'exécution : SIMD, MIMD, SPMD

2 paradigmes de communications :

- Mémoire partagée : OpenMP (si adressage global), POSIX Threads
- Mémoire distribuée : MPI, PVM, ...



Les noeuds individuels peuvent contenir plusieurs processeurs connectés entre eux par la même technologie que le réseau.

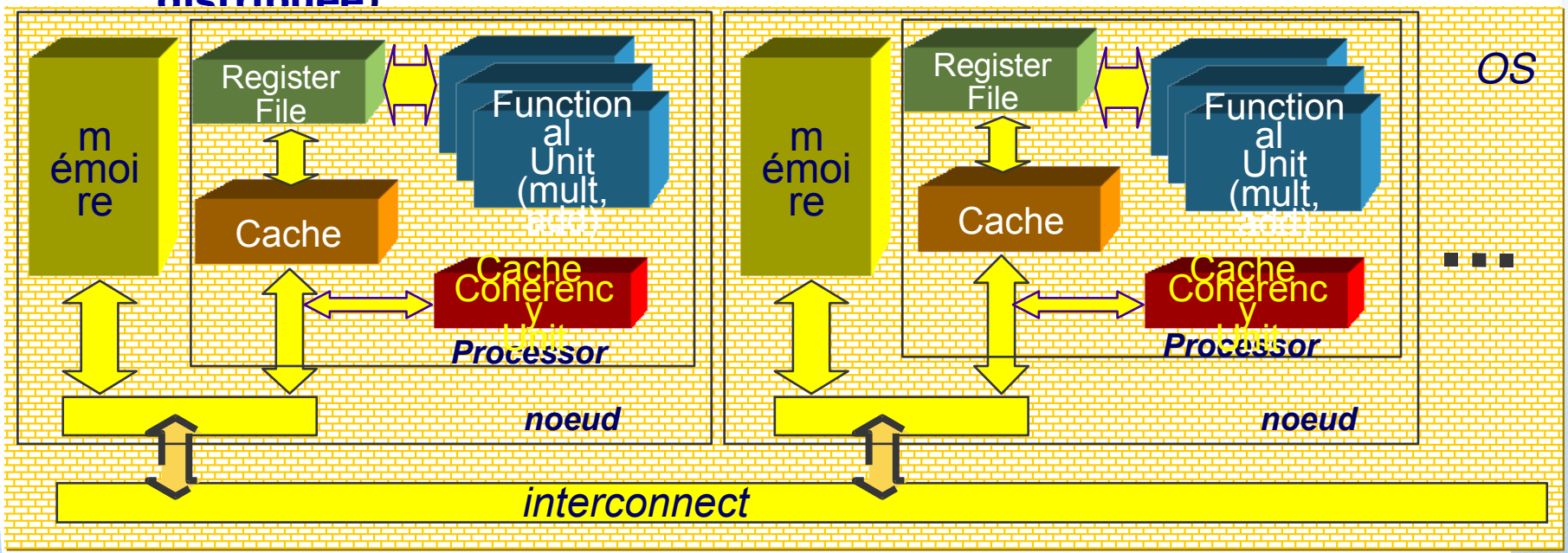
Architecture à mémoire partagée distribuée

Pour chaque processeur,les accès à la mémoire locale sont indépendants

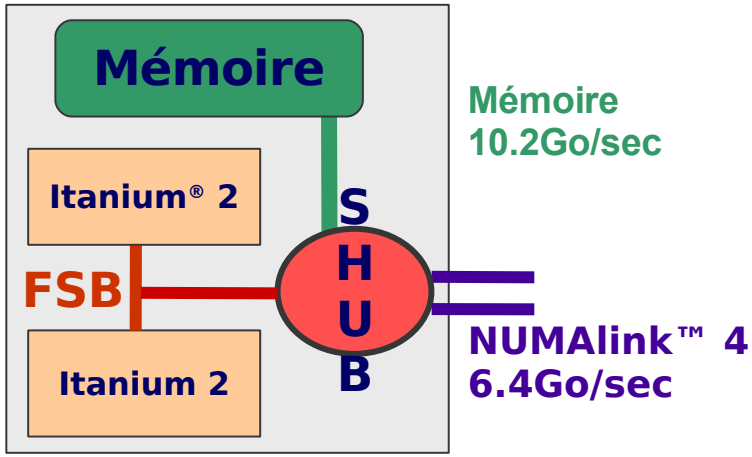
La mémoire totale est globalement adressable (point de vue du programmeur)

Non-uniform memory access (NUMA):

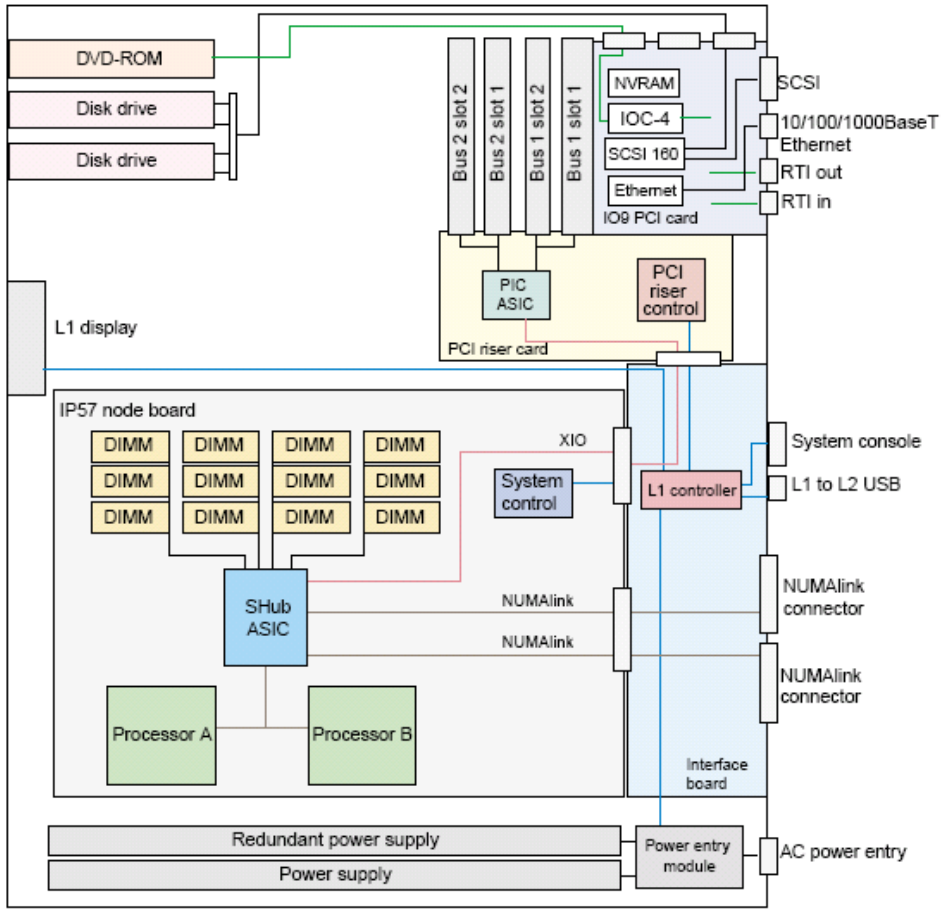
- **Les accès locaux sont plus rapides que les accès lointains (peu sensible sur SGI3000/SGIAltix)**
- **Les modèles de programmation en mémoire partagée sont utilisables**
- **la distribution des données est conseillée pour améliorer les performances (prise en compte de l ’architecture à mémoire distribuée)**



SGI ALTIX™ 350 : *module de base*



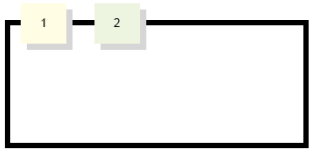
- 2 processeurs Intel® Itanium® 2
- 2 processeurs par frontside bus (6.4Go/sec)
- jusqu 'a 24 Go de mémoire par module
- contrôleur mémoire : SHUB
  8.51–10.2Go/sec bande passante mémoire
- 6.4GB/sec bande passante
  d 'interconnexion agrégé
- 4.8GB/sec bande passante I/O agrégé

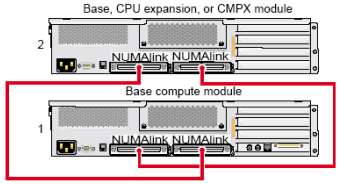


SGI ALTIX™ 350 : Topologies

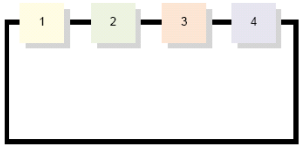
- Une Altix 350 (sans routeur):
  - jusqu ’a 16 cpu en SSI
  - Topologie : anneau

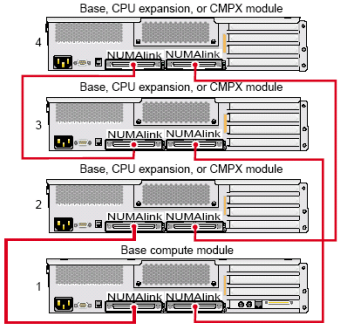
Altix350 4 cpu



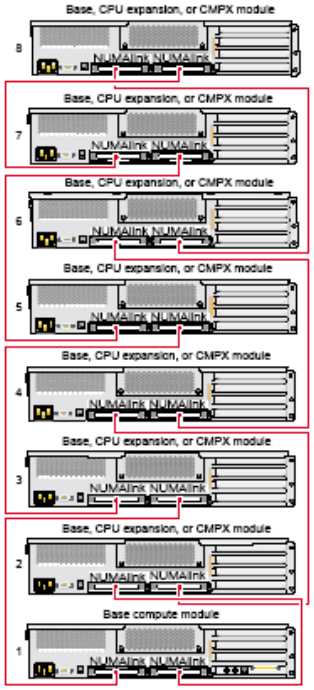


Altix350 8 cpu





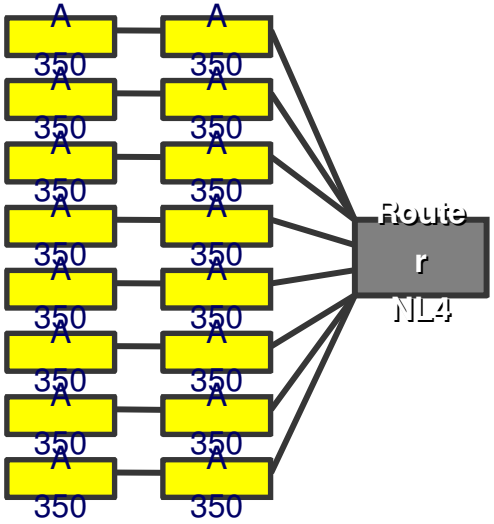
Altix350 16 cpu



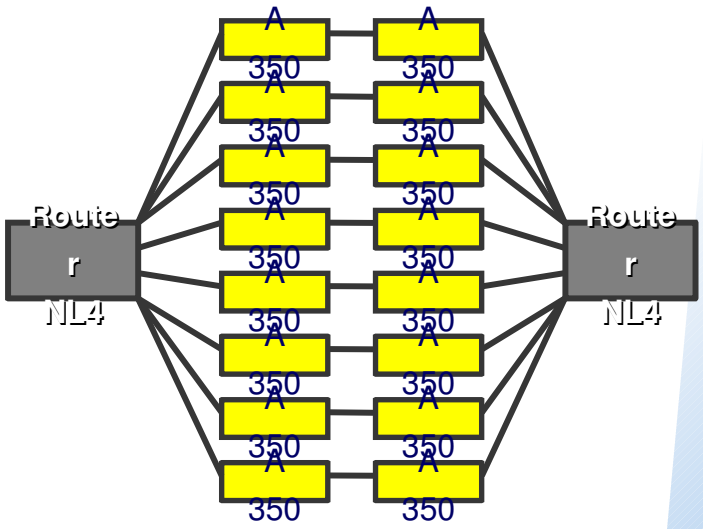
SGI ALTIX™ 350 : Topologie

- Une Altix 350 (avec routeur):
  - Jusqu'à 32 cpu en SSI
  - Topologie : simple/double plan

Configuration simple plan



Configuration double plan



NASA Ames Background

Origin 2000 256p

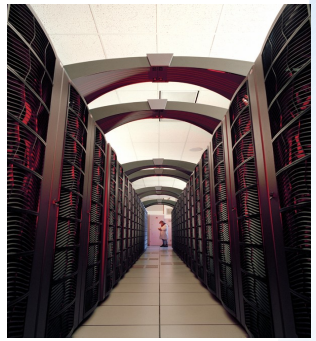
Origin 2000 512p

Premier client SGI

« Record extreme computing » :

- Avec un seul OS IRIX :
  - 1er système Origin 2000 128cpu
  - 1er système Origin 2000 256 cpu
  - 1er système Origin 2000/3000 512 cpu
  - 1er système Origin 3000 1024 cpu

Origin 3000 512p et 1024p



Altix 512p

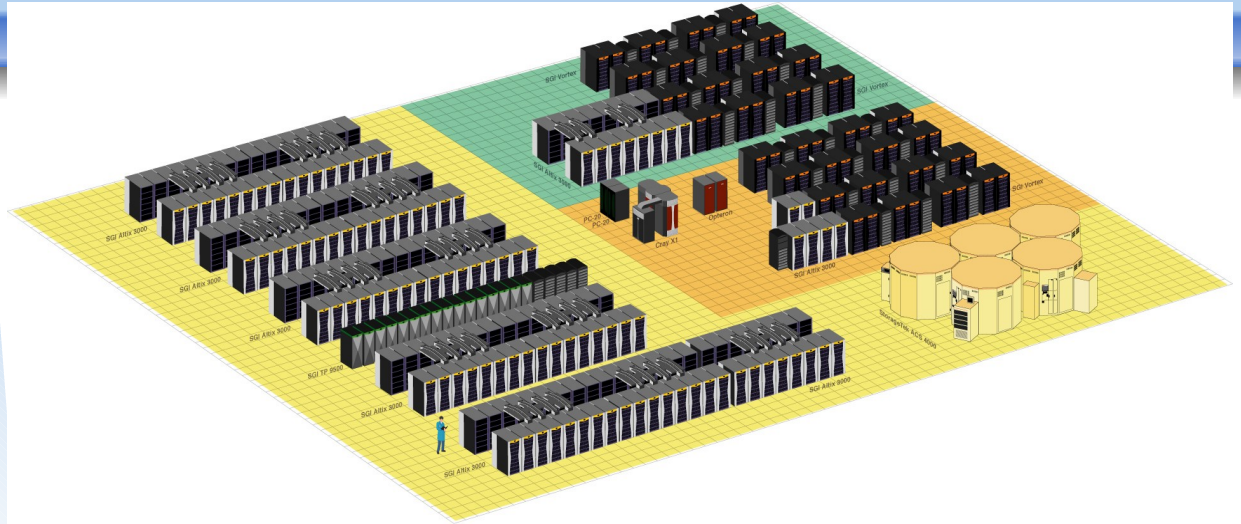
Avec un seul OS Linux :

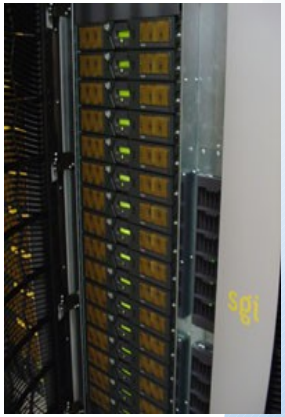
Project Columbia système Altix 512 cpu

Projet Columbia

- **Utilisateurs de la NASA et d'autres agences gouvernementales, de la recherche et de l'industrie**
- **10 240 processeurs Intel Itanium® 2**
  - 20 x 512 cpu SGI Altix
  - 2 x 64 cpu SGI Altix ⇒ frontal d'accès
- **20 téraoctets de mémoire totale total mémoire**
  - 1 téraoctets de mémoire par 512 cpu
- Réseau **infiniband**, 1 / 10 gigabit Ethernet
  - 6 Infiniband HCAs / système
- **~500 To stockage gérés par CXFS/DMF**
- 128 x pipes Silicon Graphics Prism
- TOP500 (Novembre 2004, www.top500.org) :
  - Classé numéro 2
  - Rpeak (GFlops):**60960**
  - Rmax (GFlops):**51870**

Configuration Columbia






hp
invent

Torus Grid: HP EV7 16P System

EV7 64P Latency

319	283	247	211	247	283	319
283	247	211	175	211	247	283
247	211	175	140	175	211	247
211	175	140	75	140	175	211
247	211	175	140	175	211	247
283	247	211	175	211	247	283
319	283	247	211	247	283	319
355	319	283	247	283	319	355

250ns average memory latency

Processor Sets

- A processor set is a collection of zero or more processors
- Every thread on the system belongs to a processor set
- A thread may only be scheduled on a processor within its processor set
- At bootup, a default processor set is created and this “default_pset” contains all processors on the system

Processor sets

- Psets can be created or destroyed
  - default processor set is the exception
- processors can be added or removed from processor sets

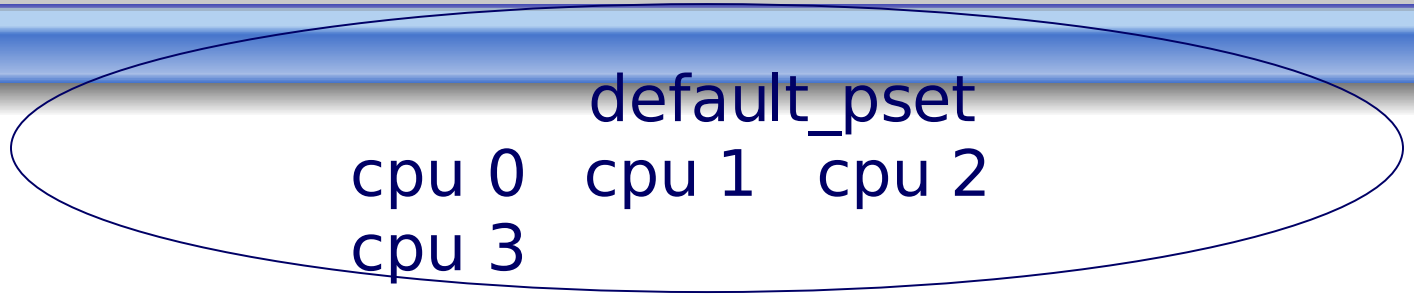
Processor sets

- Processors sets allow you to run jobs on a specific group of processor(s).
  - If there is only one process running on a processor, there can be a performance boost because of cache & scheduling efficiencies.
  - good for real-time applications because a processor can be guaranteed to be instantly available to run a time critical task.
  - Another way to guarantee cpu resource availability

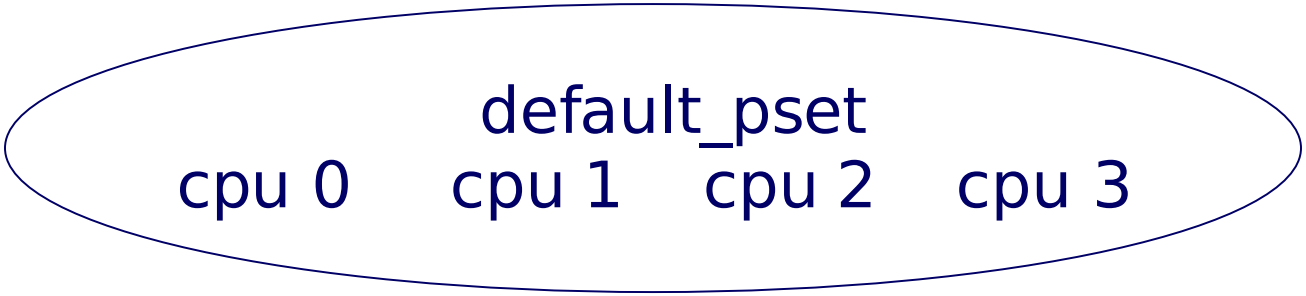
Processor Sets

- By default, all threads are assigned to the default_pset
- new pset can be created with the pset_create command. This pset will have no processors.
- Processors can be added to a pset with the pset_assign_cpu command.
- If a created pset is destroyed (pset_destroy command), all processors from that pset are returned to the default_pset (V4.X)

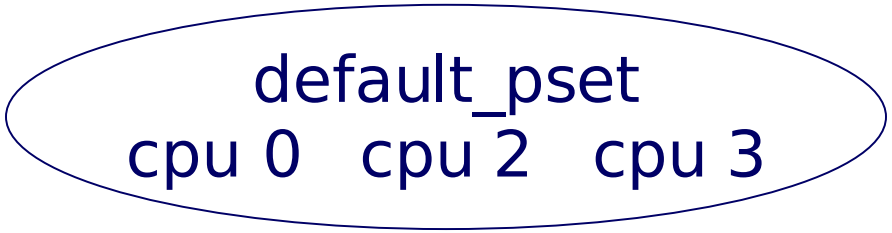
Processor Sets : example

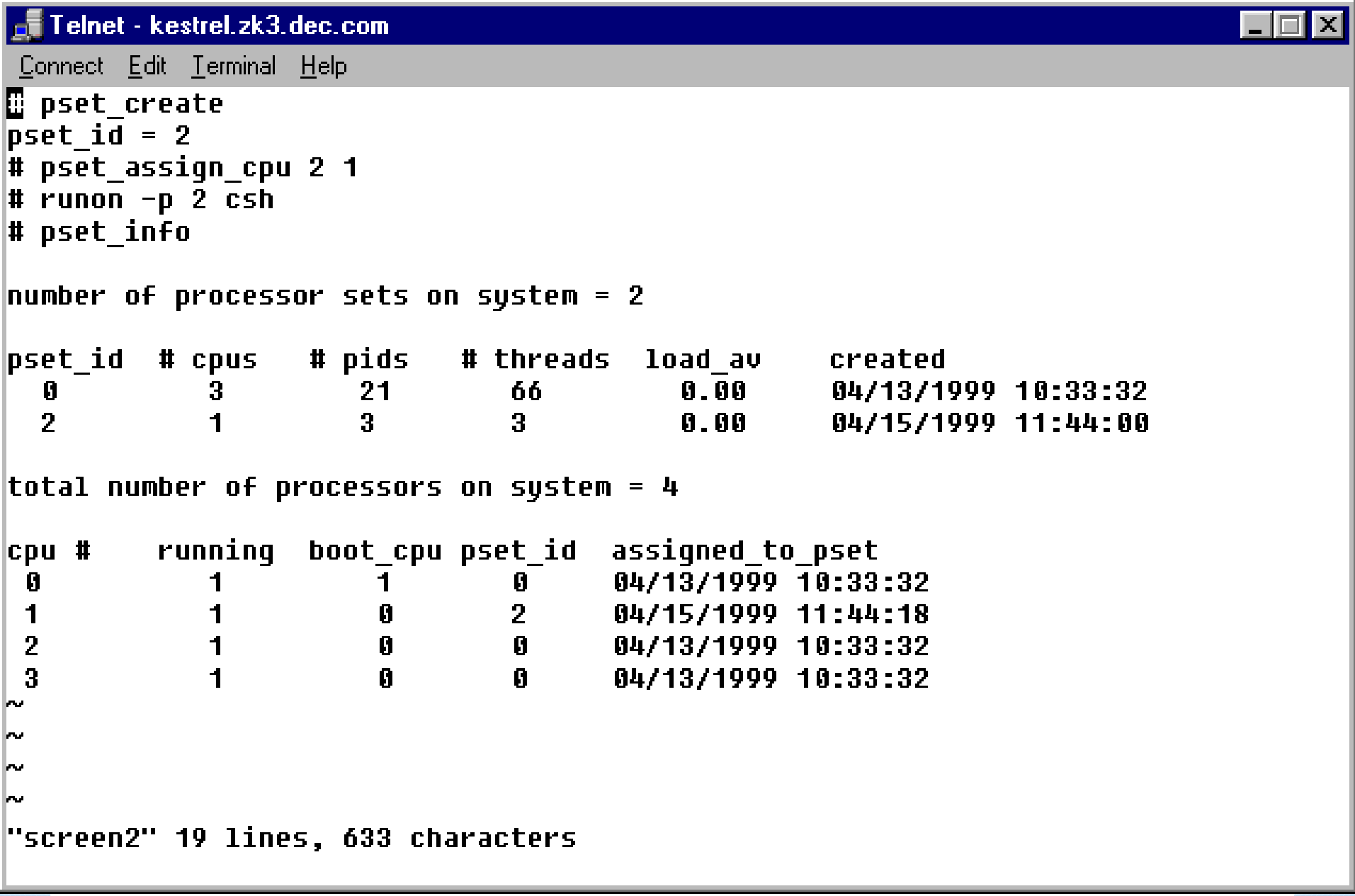


pset_create

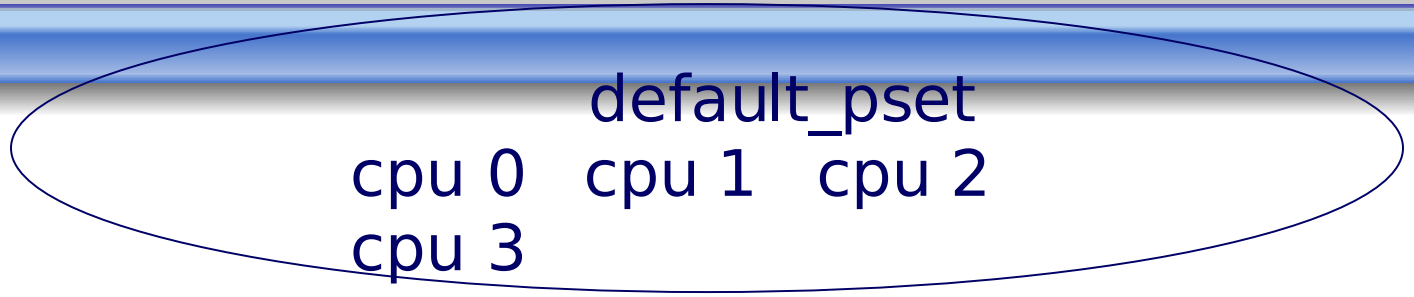


pset_assign_cpu 2 1

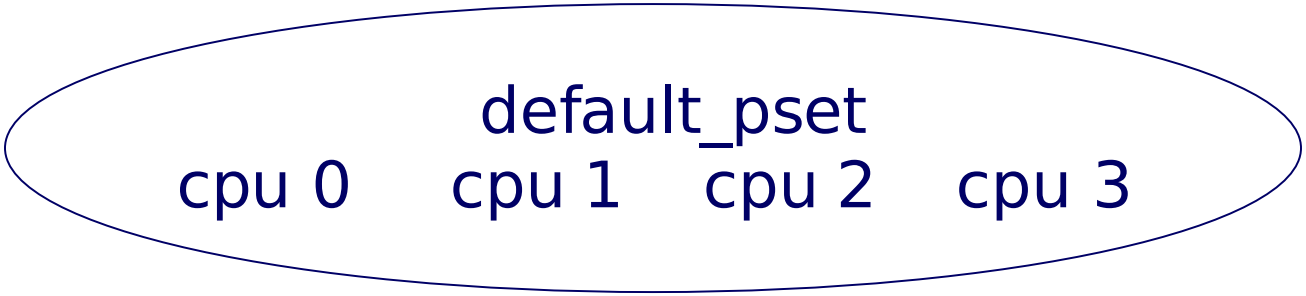



Telnet - kestrel.zk3.dec.com
Connect Edit Terminal Help
# pset_create
pset_id = 2
# pset_assign_cpu 2 1
# runon -p 2 csh
# pset_info
number of processor sets on system = 2
pset_id # cpus # pids # threads load_av created
0 3 21 66 0.00 04/13/1999 10:33:32
2 1 3 3 0.00 04/15/1999 11:44:00
total number of processors on system = 4
cpu # running boot_cpu pset_id assigned_to_pset
0 1 1 0 04/13/1999 10:33:32
1 1 0 2 04/15/1999 11:44:18
2 1 0 0 04/13/1999 10:33:32
3 1 0 0 04/13/1999 10:33:32
~
~
~
~
"screen2" 19 lines, 633 characters

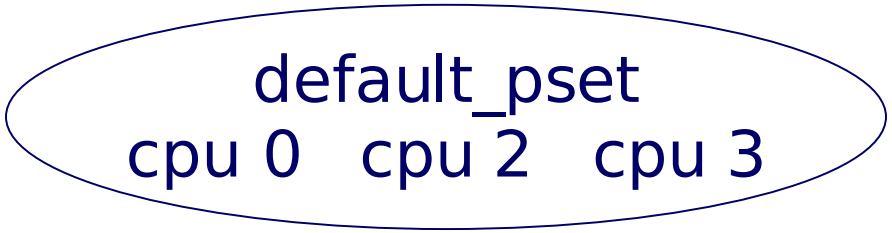
Processor Sets : example

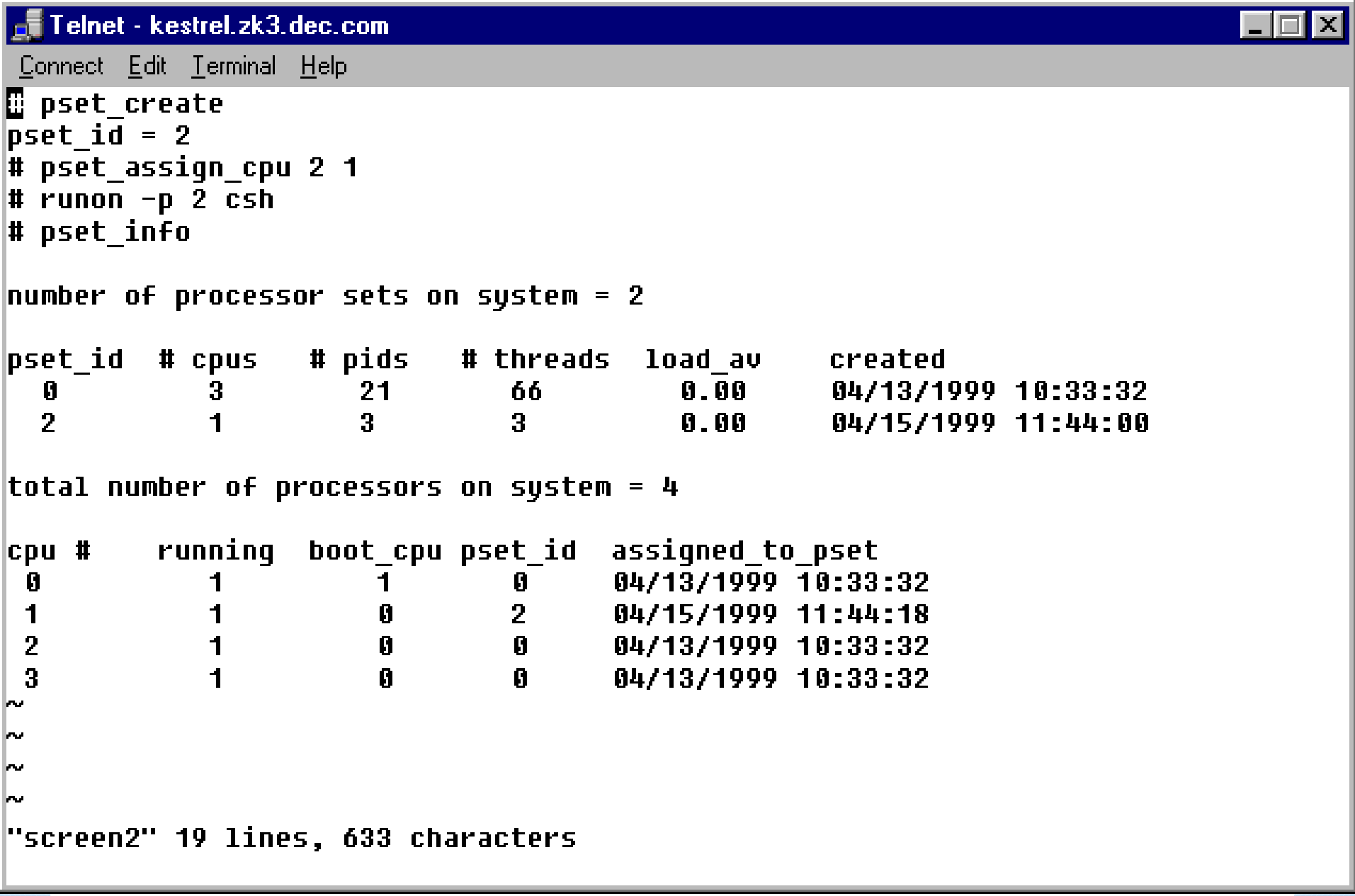


pset_create



pset_assign_cpu 2 1




Telnet - kestrel.zk3.dec.com
Connect Edit Terminal Help
# pset_create
pset_id = 2
# pset_assign_cpu 2 1
# runon -p 2 csh
# pset_info
number of processor sets on system = 2
pset_id # cpus # pids # threads load_av created
0 3 21 66 0.00 04/13/1999 10:33:32
2 1 3 3 0.00 04/15/1999 11:44:00
total number of processors on system = 4
cpu # running boot_cpu pset_id assigned_to_pset
0 1 1 0 04/13/1999 10:33:32
1 1 0 2 04/15/1999 11:44:18
2 1 0 0 04/13/1999 10:33:32
3 1 0 0 04/13/1999 10:33:32
~
~
~
~
"screen2" 19 lines, 633 characters

Using Processor sets

- Runon (1) command
  - generally binds a job to a processor within its processor set
    - runon 1 ls runs the ls command on processor 1
  - runon -p binds a job to another processor set
    - runon -p 2 csh runs a c-shell on processor set 2
    - now all commands issued from that shell will be bound to processor set 2
  - runon -p 2 -x csh gives exclusive use of the processor set

Cluster Components…2 OS

- Systèmes traditionnels
  - Linux
  - Microsoft Windows
  - MacOS
  - SUN Solaris
  - IBM AIX
- Single System Image
  - Virtualisation de l'architecture
  - Une seule image système pour gérer la grappe

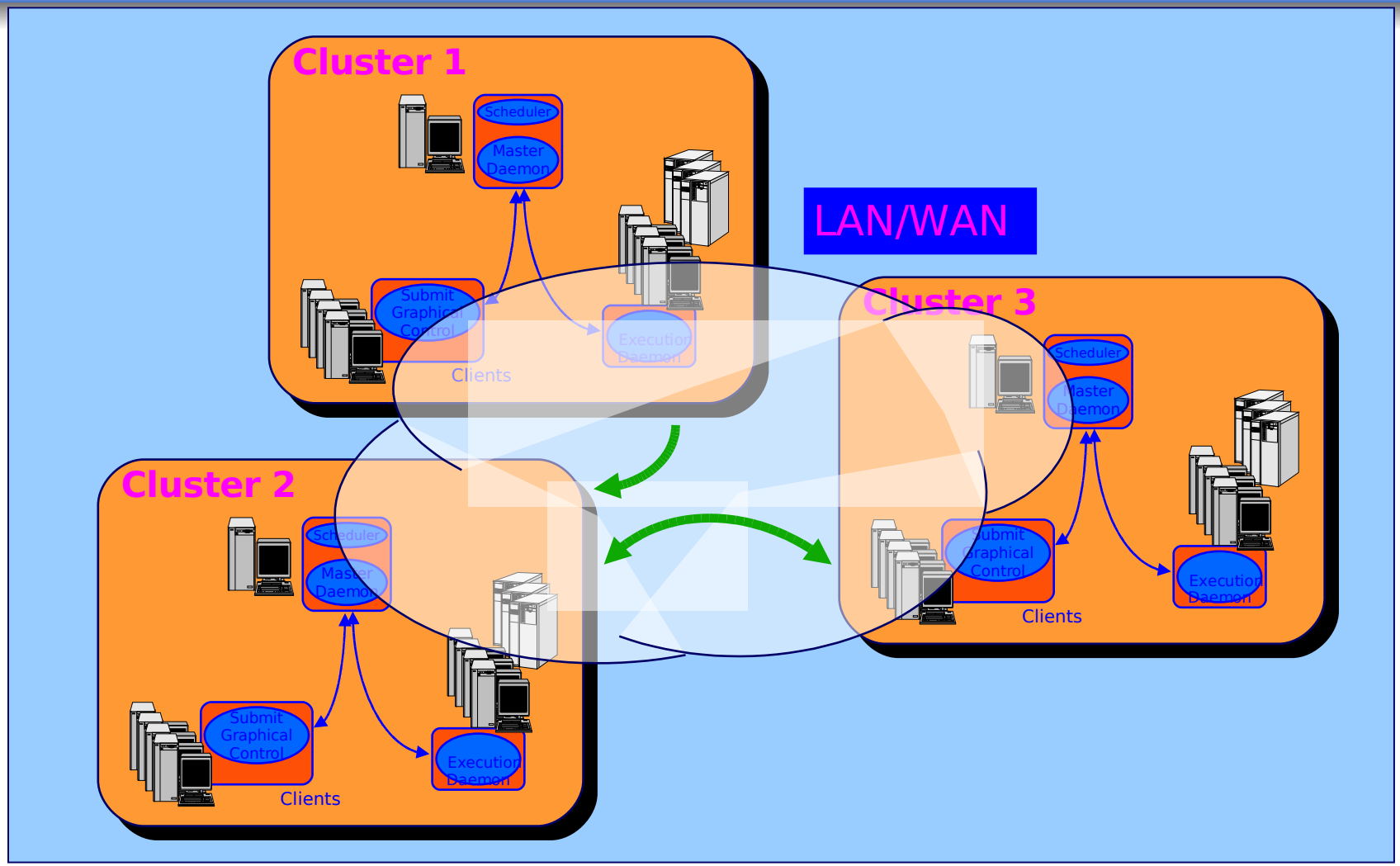
What Next ??

Clusters of Clusters (HyperClusters)

Computational Grid

Internet Computing

Clusters of Clusters (HyperClusters)



Liberté - Égalité - Fraternité
RÉPUBLIQUE FRANÇAISE
MINISTERE DE LA RECHERCHE

Action Incitative Concertée [ACI]
Globalisation des Ressources Informatiques et des Données [GRID]

Introduction aux grilles Globalisation des ressources informatiques et des données

Contenu de la présentation

- Grille informatique
- Les types de grilles informatiques et leur déploiement
- Quelques grands défis
- ACI Globalisation des Ressources Informatiques et des Données (GRID)

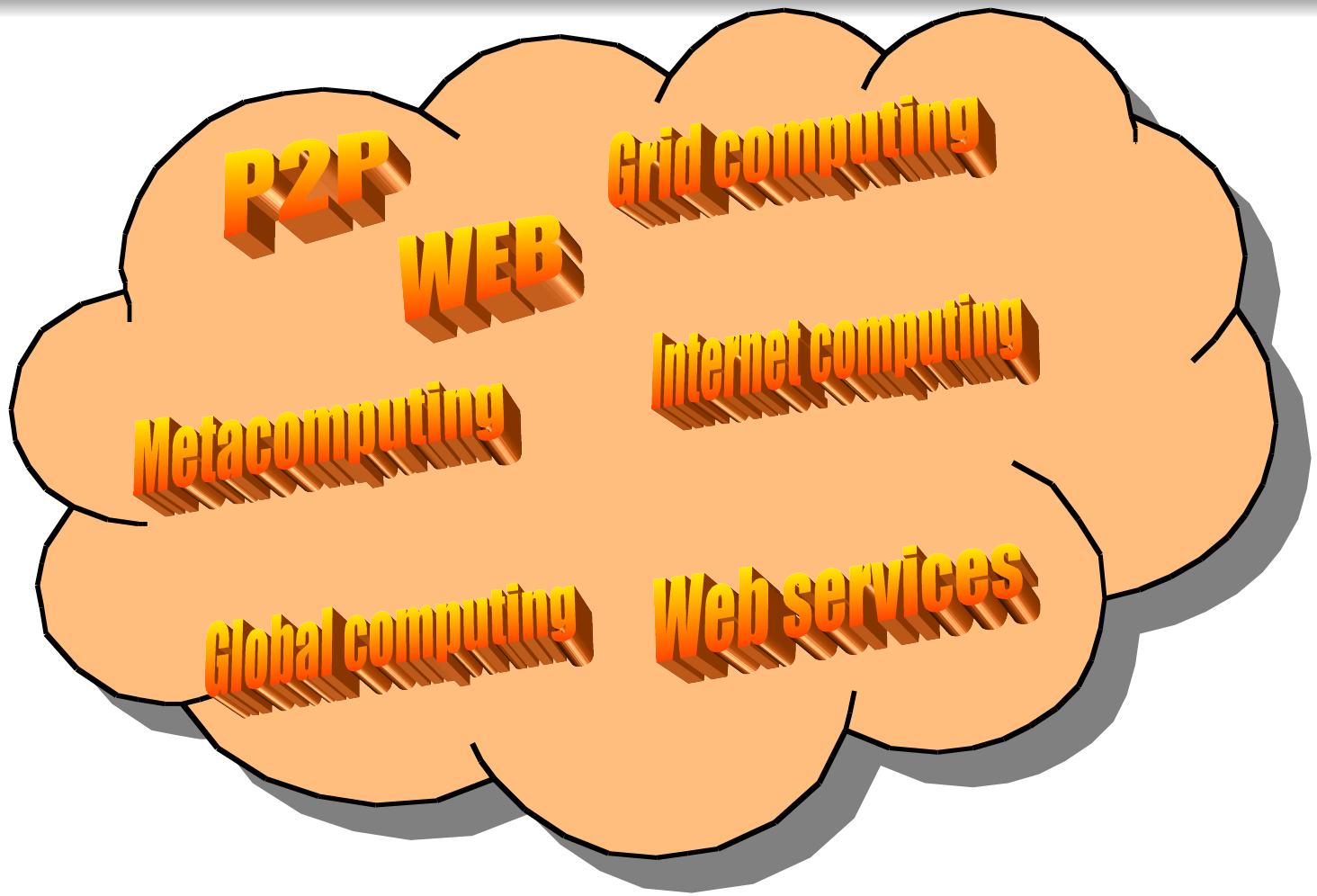
Le concept de Grille

le réseau électrique :
Approche pour la distribution de la puissance électrique et la haute-tension

Le concept de Grille informatique (GRID)

Approche pour la distribution de la puissance informatique
le réseau Internet et la haute-performance
(parallélisme et distribution)

Et ses différentes incarnations…



Une tentative de classification

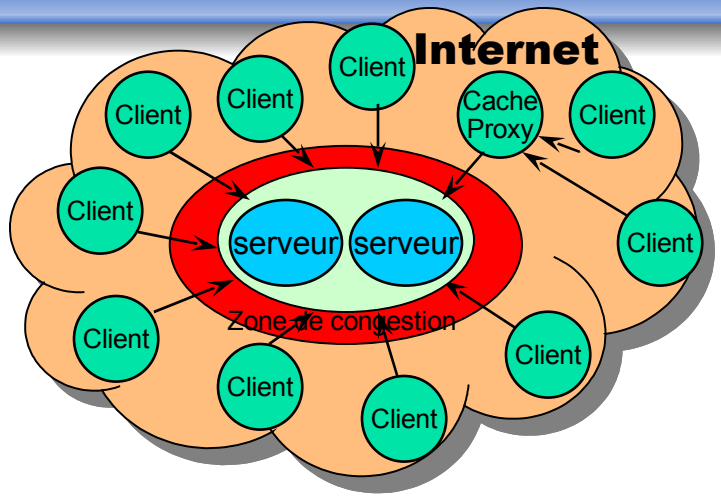
- Grille d'informations
  - Faire partager la connaissance
- Grille de données
  - Faire partager la grande échelle
- Grille de calcul
  - Agréger la puissance de calcul

Systèmes distribués ! *

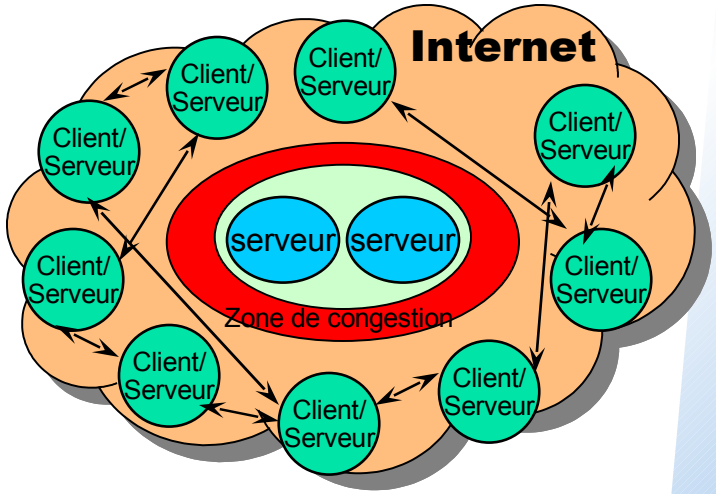
* «A distributed system is a collection of independent computers that appear to the users of the system as a single computer » Distributed Operating System. A. Tanenbaum, Prentice Hall, 1994

Modèles de déploiement

- Client-serveur
  - Centralisé ou Distribué
  - Utilisation de caches pour éviter la congestion
  - Information centralisée

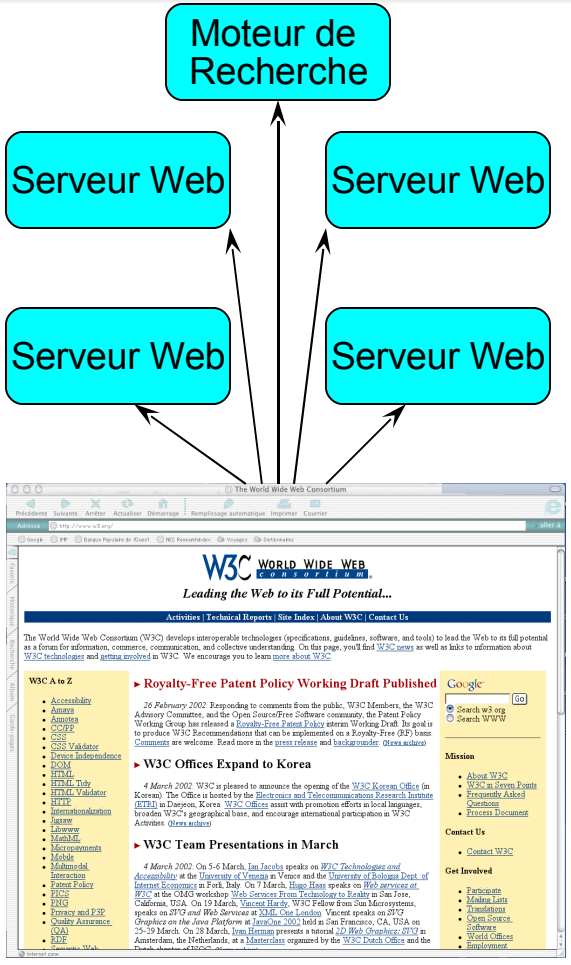


- Pair-à-Pair (P2P)
  - Chaque pair est à la fois client et serveur
  - Distribution de la charge dans le réseau
  - Information distribuée



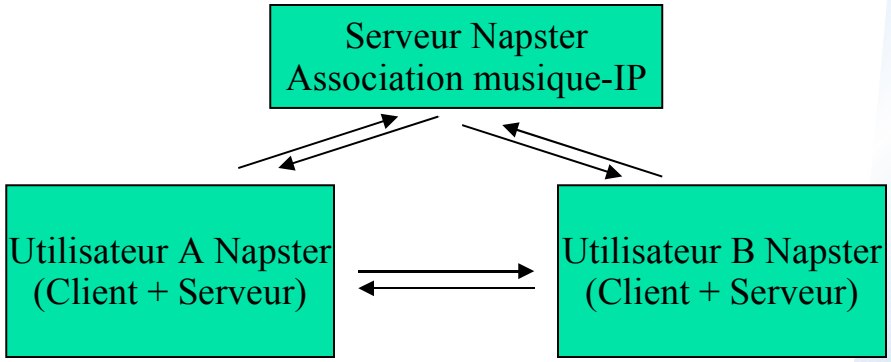
Modèle client/serveur pour les Grilles d'information : Le cas des sites WEB

- Sans doute la première incarnation du concept de grille
- Accès à l'information
  - A partir d'une adresse http
  - A partir d'un moteur de recherche
- Transparence lors de l'accès à l'information
  - On ne sait pas toujours d'où vient l'information



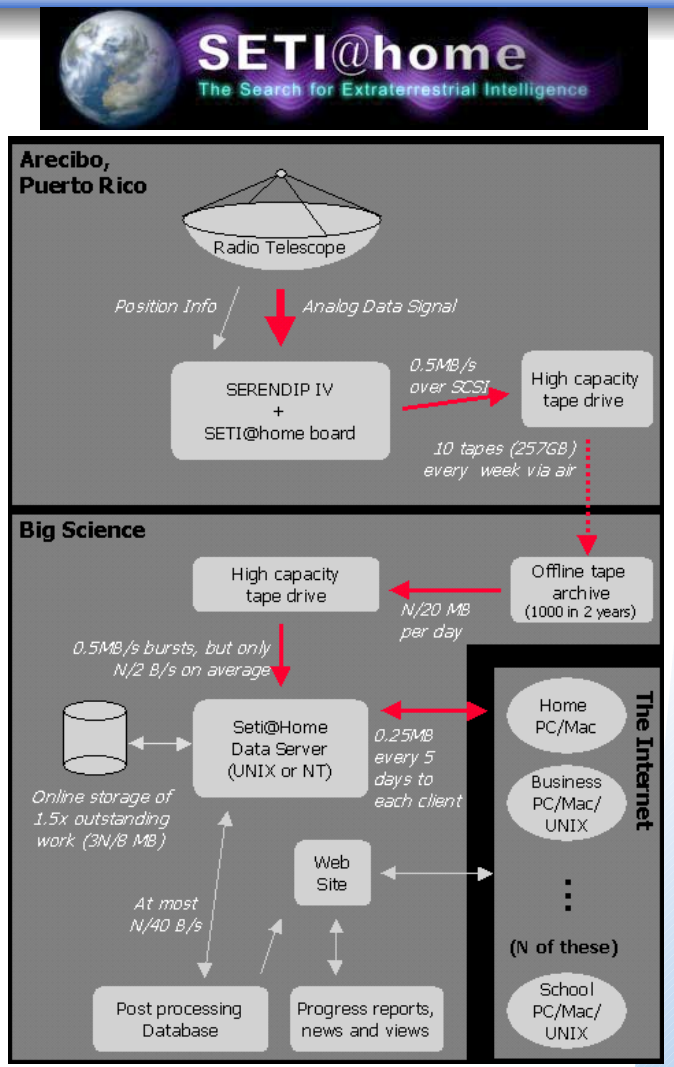
Modèle client/serveur pour les Grilles de données : Le cas NAPSTER

- Entre le client/serveur et le P2P
  - Accès à des données via un site unique contenant un index
- Stockage de données
- Partage des données
  - Données « inaltérables »
  - Copies multiples sans aucun contrôle
- Limites de l'approche
  - Plutôt du client/serveur que réellement P2P
  - Serveur « attaquable »
    - Par les tribunaux…
    - Ou par d'autres…



Modèle client/serveur pour les Grilles de calcul: l'Internet Computing

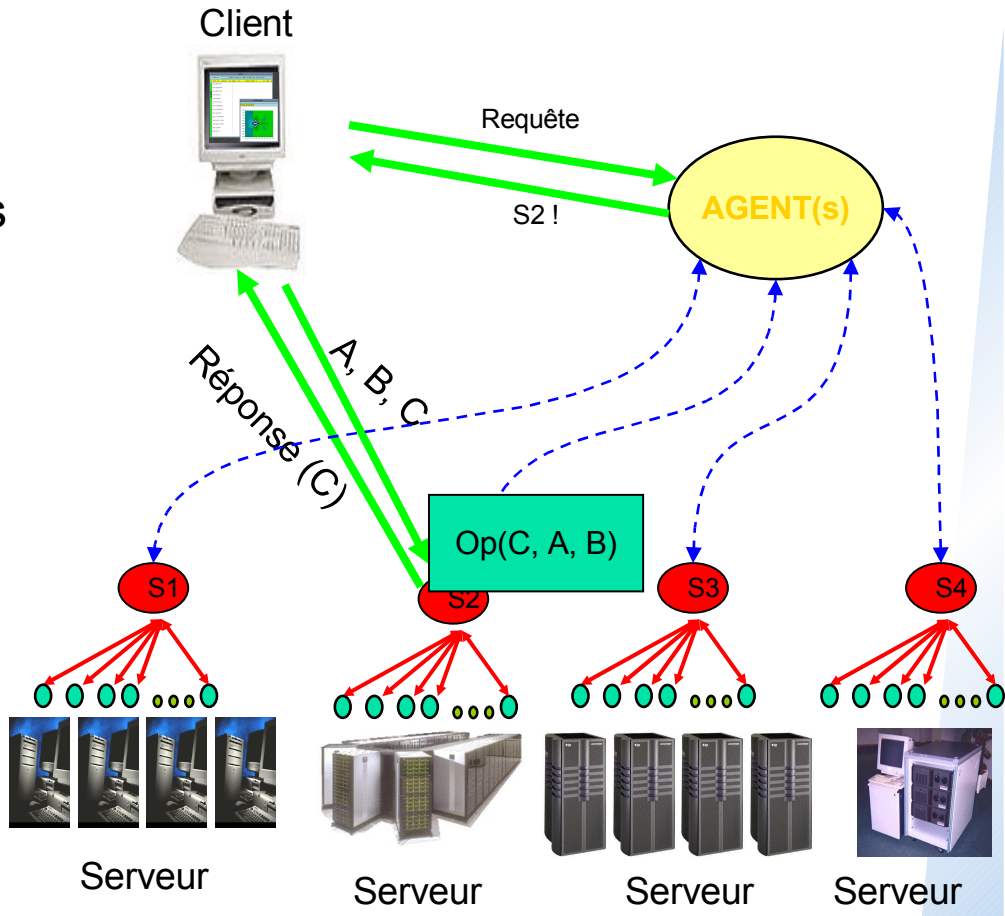
- Principe
  - Des millions de PC en attente…
  - Utilisation des cycles processeurs inutilisés (environ 47% en moyenne dans une entreprise *) via un économiseur d'écran
- Exemples
  - SETI@HOME (ce n'est pas du P2P!)
    - Recherche de signaux extra-terrestres
    - 33.79 Teraflop/s (à comparer aux 12.3 Teraflop/s de l'ordinateur le plus puissant au monde au LLNL !)
  - DECRYPTHON
    - Etablir la carte des 500 000 protéines du vivant
  - RSA-155
    - Casser des codes cryptographiques



** d'après une enquête d'Omni Consulting Group*

Modèle client/serveur pour les Grilles de calcul: Le metacomputing

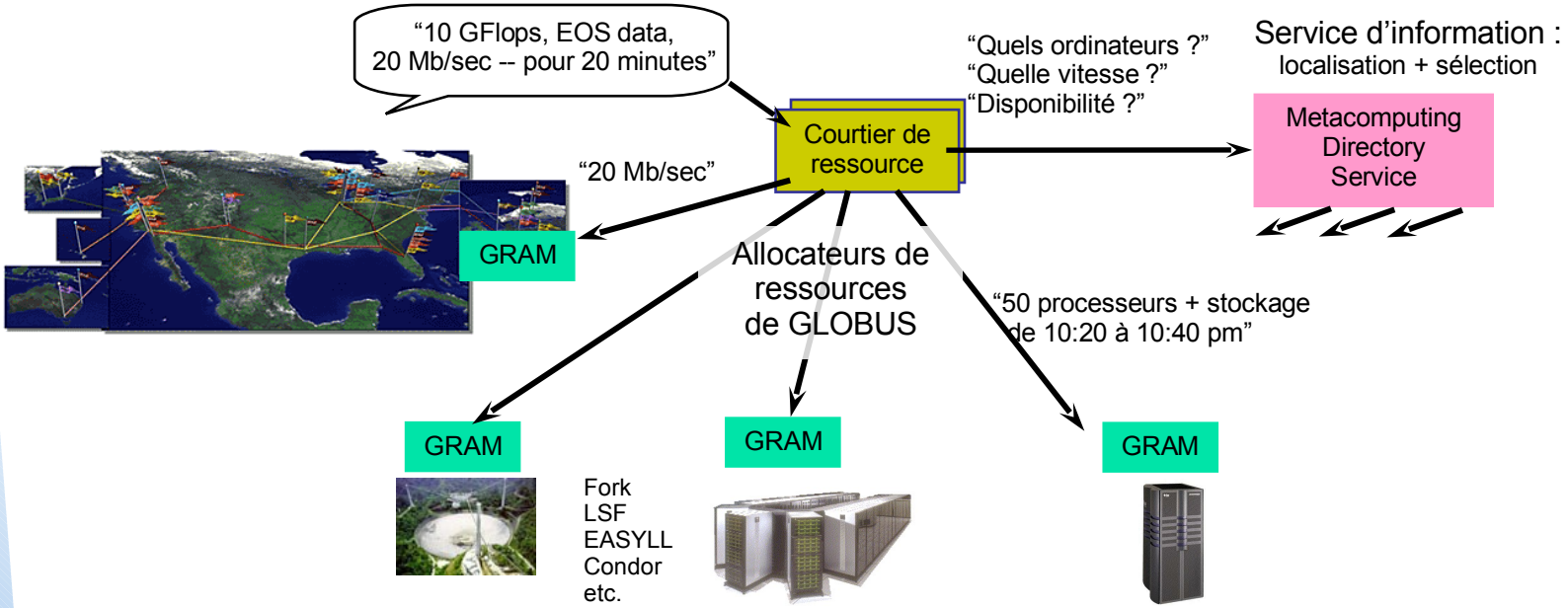
- Principe
  - Acheter du service de calcul sur l'Internet
  - Service = applications préinstallées + calculateurs
- Exemples
  - Netsolve (Univ. Tennessee)
  - NINF (Univ. Tsukuba)
  - DIET (ENS-Lyon/INRIA)



Modèle client/serveur pour les Grilles de calcul: Le grid computing

- Principe
  - Utiliser un supercalculateur parallèle virtuel
  - Faire exécuter ses applications sur des ressources distantes

- Exemples
  - Globus
  - Légion
  - Unicore



Modèles de déploiement des grilles: le modèle distribué Pair-à-Pair

- Grille de données
  - Gnutella, Freenet
- Grille d'information:
  - Recherche décentralisée à la google
- Grille de calcul: CG2P

En résumé

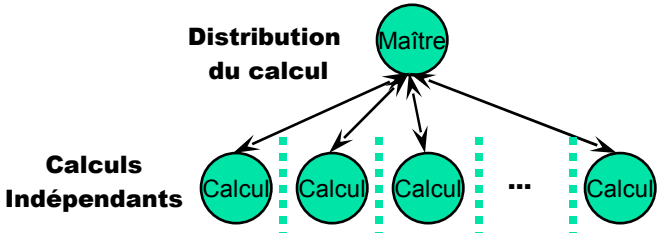
Déploiement / Grilles	Client/serveur	P2P
Grilles de données	Napster Datagrid	Gnutella, Kazaa, Freenet, JXTA
Grilles d'information	Sites Web Moteurs de recherche (Altavista, Google, …)	Web ?
Grilles de calcul	SETI-HOME, Decrypthon Netsolve, Ninf, DIET GLOBUS, Legion, Unicore	CGP2P ?

Quelques grands défis…

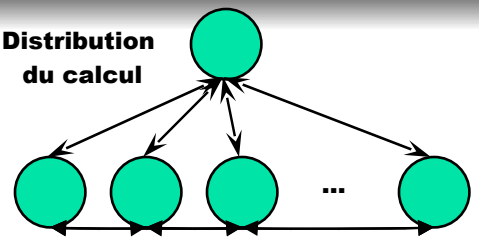
- Middleware et systèmes
  - Internet computing
  - Metacomputing
  - Vers un Grid-aware OS ?
- La programmation des grilles
  - Algorithmique
  - Génie logiciel

Les défis du « Internet computing »

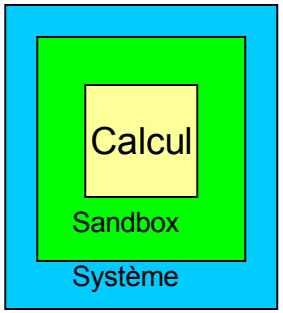
- Cela marche bien pour un spectre étroit d'applications
  - Parallélisme embarrassant
- Usage exclusivement « non-commercial »
  - Casser des codes cryptographiques (défi RSA-155)
  - Recherche de signaux extra-terrestre (SETI@HOME)
  - Décryptage du génome (Decrypton)
- Cette approche est-elle généralisable ?



Quels sont les problèmes?

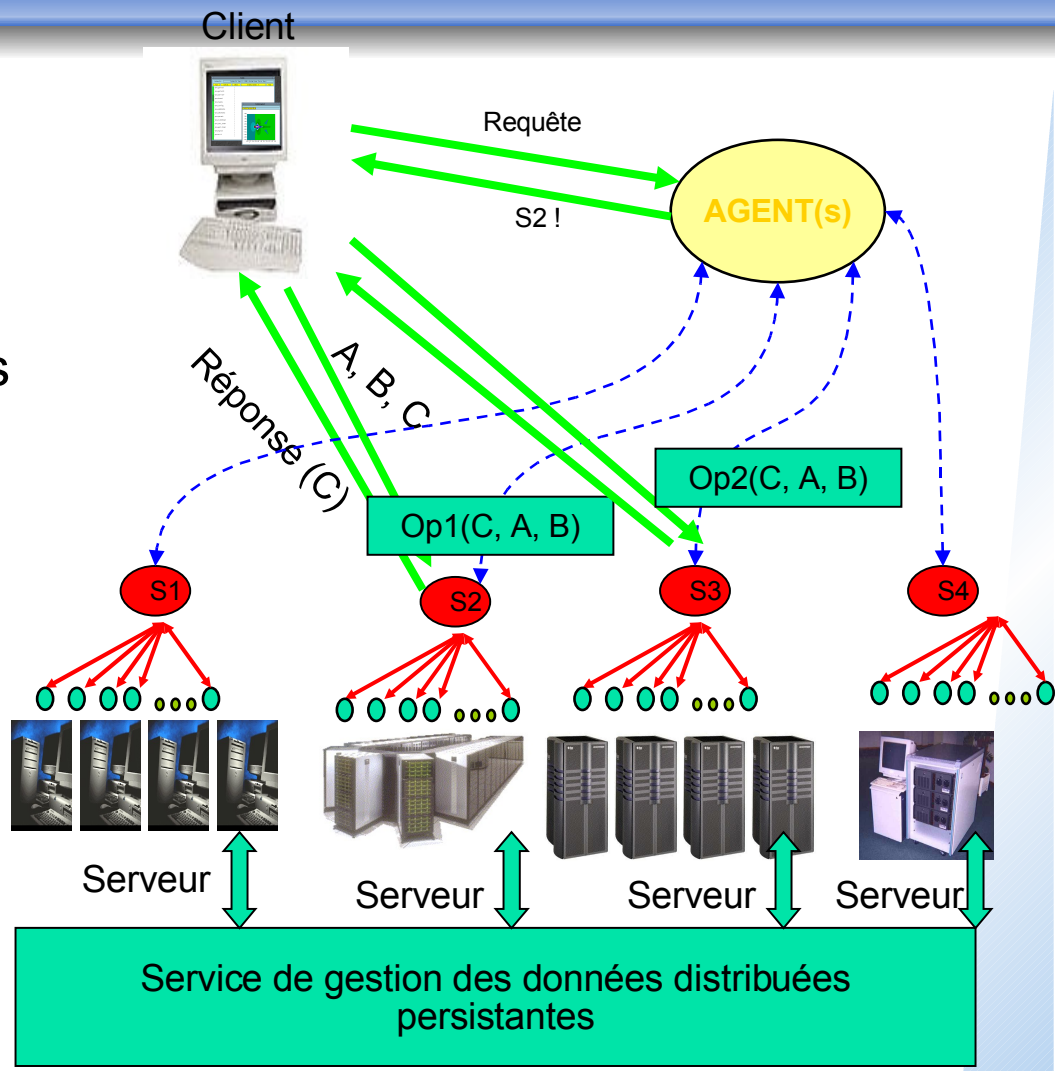


- Elargir le spectre d'applications
  - De vraies applications parallèles
  - Autoriser la communication
- Sécurité
  - Etes vous prêt à laisser exécuter n'importe quoi sur votre PC ?
  - Technique du sandbox (isolation du code de calcul)
  - Comment communiquer avec le monde lorsqu'on est isolé ?
- Parité
  - cela marche si tout le monde joue le même jeux…
- Modèle de déploiement rudimentaire
  - Essentiellement client/serveur
  - A terme, nécessité du P2P



Les défis du metacomputing

- Quels sont les problèmes et défis ?
  - Stockage des données pour éviter les transferts multiples entre client et serveurs
  - Sécurité dans les transferts
  - Modèle de déploiement P2P

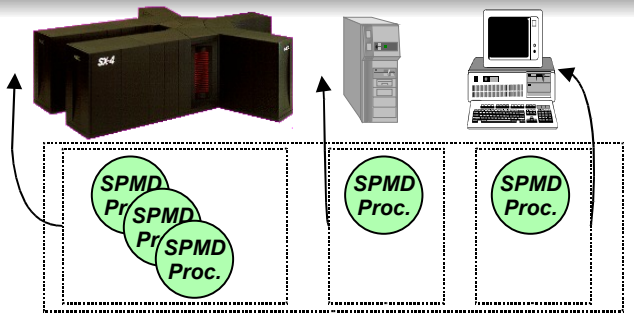


Vers de nouveaux systèmes d'exploitation Grid-aware ?

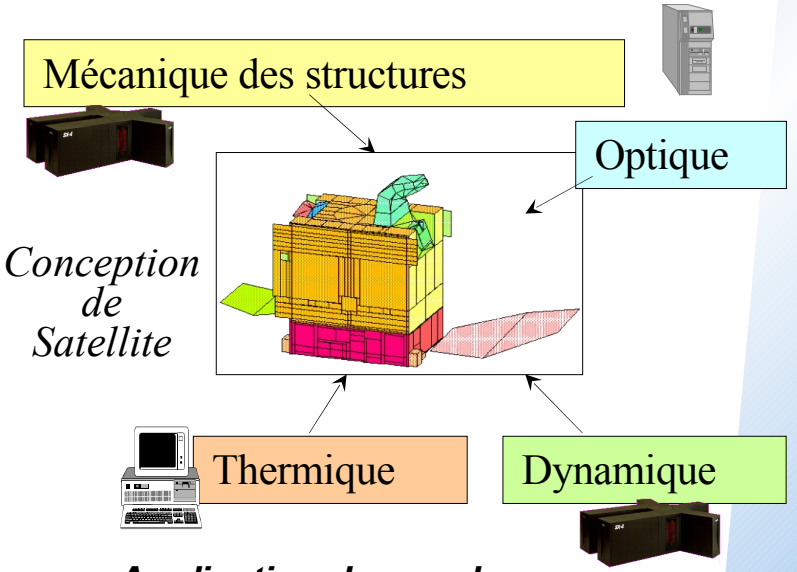
- Conception de systèmes d'exploitation GRID-aware
  - Actuellement: essentiellement middleware
  - Gestion des ressources = rôle de l'OS !

Programmer les grilles de calcul

- Un champ applicatif vaste avec des besoins variés…
- Codes parallèles
  - Une grille de calcul est vue comme un calculateur parallèle virtuel (la genèse du Grid)
- Couplages de codes
  - Une application est un assemblage de plusieurs codes de calcul



Application parallèle



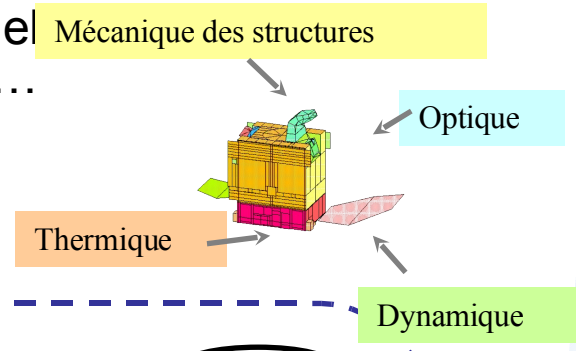
Application de couplage

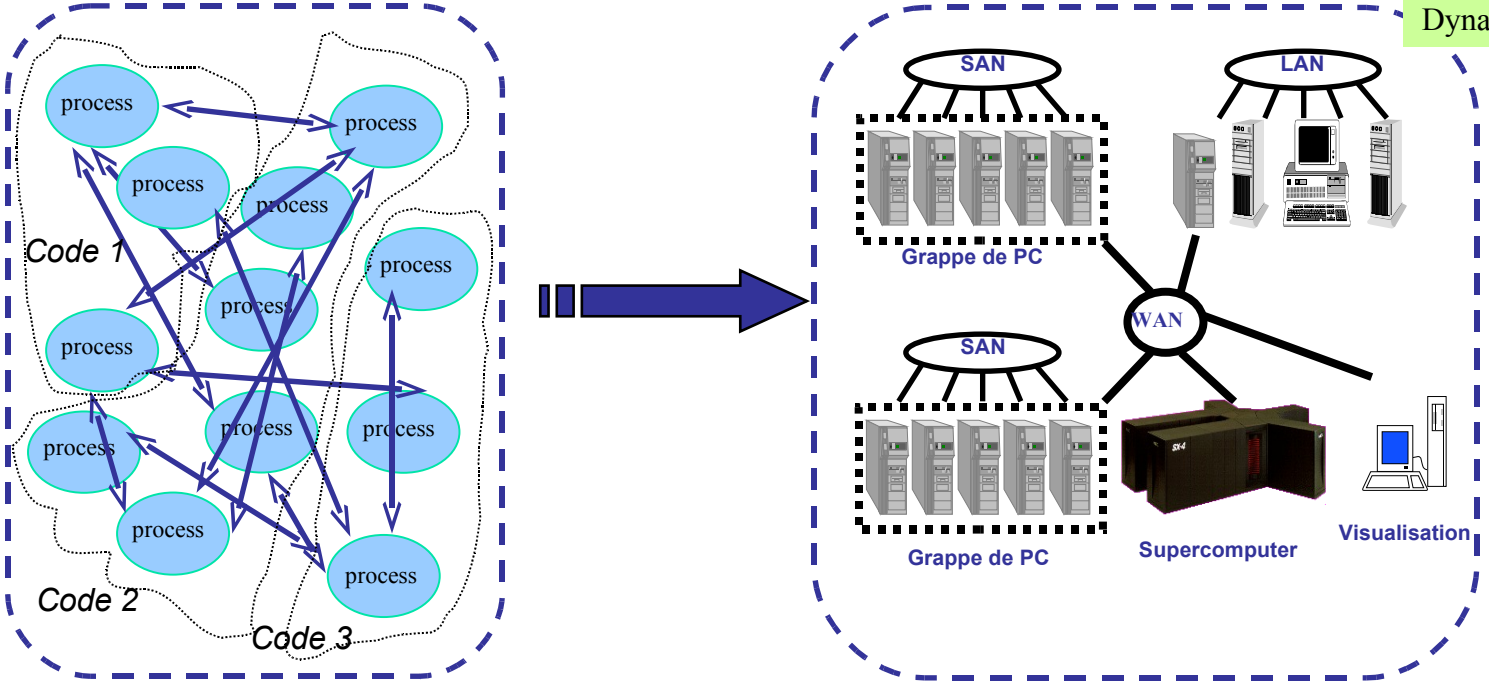
Vers une algorithmique des grilles de calcul

- Programmer une grille de calcul comme un calculateur parallèle virtuel
- Repenser l'algorithmique parallèle
  - Essentiellement conçu pour des architectures parallèles régulières et à configuration statique
- Caractéristiques des grilles informatiques
  - Unité de calcul hétérogène
  - Non-prédictibilité des performances des réseaux
  - Aspect dynamique des ressources de calcul
- Problèmes et défis
  - Politique d'ordonnancement des calculs
  - Equilibrage dynamique par redistribution intensive des données
  - Recouvrement calcul/communication

Couplage de codes

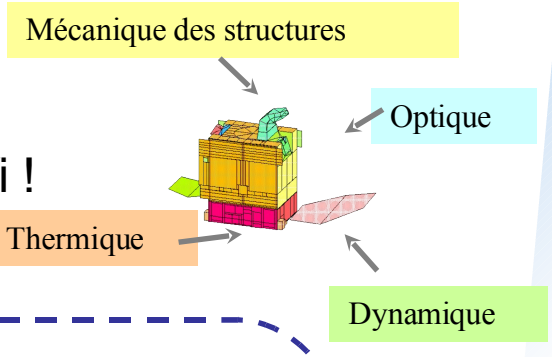
- Utilisation des exécutifs conçus pour la programmation parallèle
  - Une grille de calcul est un calculateur parallèle virtuel programmation par échange de message s'impose…

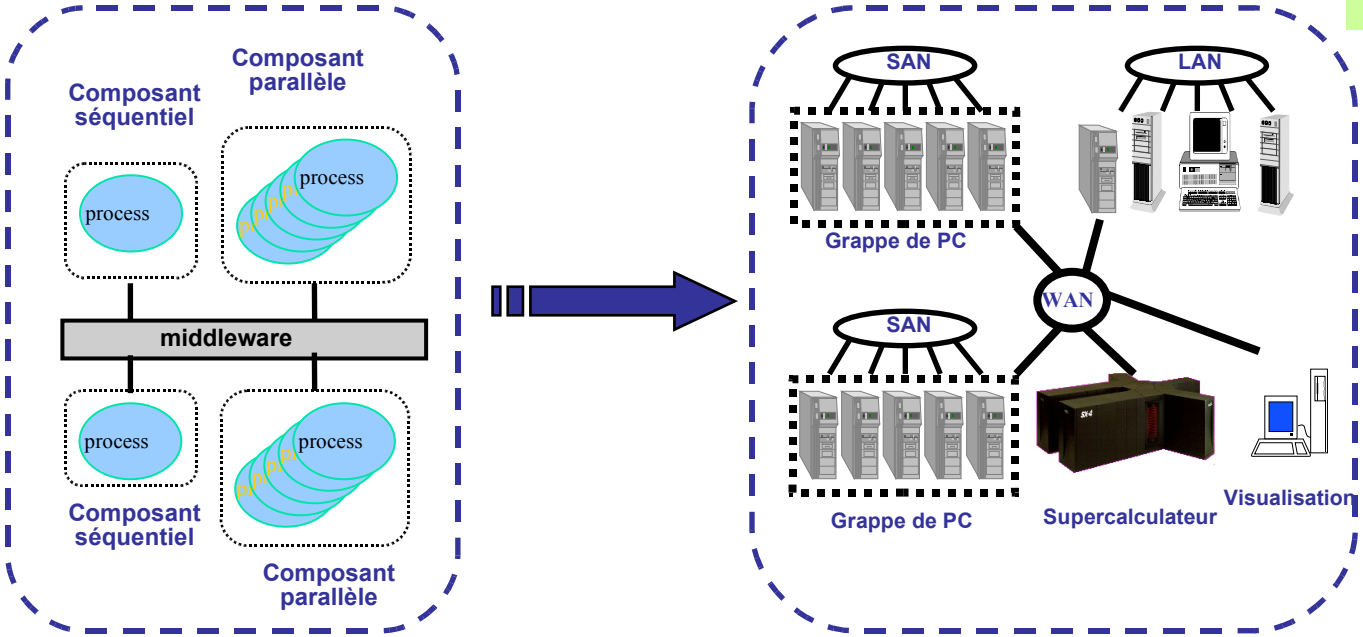




Une approche plus moderne

- Objets distribués / composants
  - Structuration de l'application
  - Encapsulation des codes
- Couplage de codes parallèles
  - Interconnexion des objets / composants -> un réel défi !





ACI GRID

Action Incitative Concertée [ACI]
Globalisation des Ressources Informatiques
et des Données [GRID]

- dynamiser et rendre opérationnelle la contribution des équipes de recherche françaises
- déploiement d'outils logiciels pour la mise en œuvre de " grilles expérimentales " pour des applications diverses
- systèmes et environnements pour le calcul distribué ou pour l'exploitation d'ensembles de données de très grande taille
- modélisation, algorithmique, couplage de codes, visualisation, pré et post-traitements

L'ACI GRID et les défis…

Action Incitative Concertée [ACI]
Globalisation des Ressources Informatiques et des Données [GRID]

- Grilles de calcul
  - Internet Computing
    - CGP2P (F. Cappello, LRI/CNRS)
  - Metacomputing
    - ASP (F. Desprez, ENS-Lyon/INRIA)
- Programmation des Grilles
  - Algorithmique:
    - GRID2 - Thème 3 (J-L. Pazat, IRISA/INSA)
    - TAG (S. Genaud, LSIIT)
    - ANCG (N. Emad, PRISM)
  - Composants:
    - RMI (C. Perez, IRISA/INRIA)
    - CONCERTO (Y. Maheo, VALORIA)

En conclusion

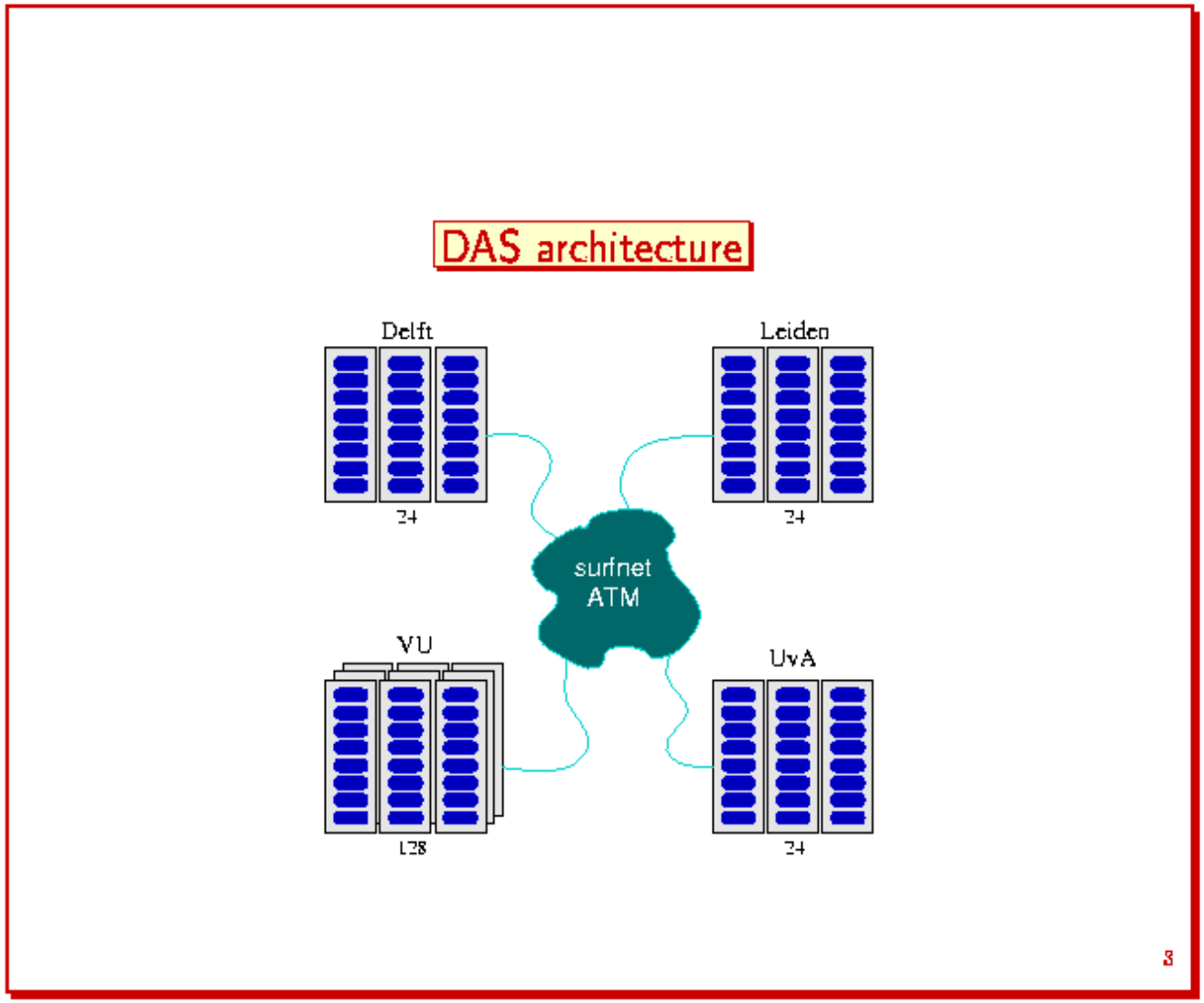
- Une grande variété dans les grilles informatiques
  - Pas de nouvelle thématique
    - Mais plutôt une adaptation de l'existant
    - Une « combinaison » de technologies du parallélisme et du distribué pour les grilles de calcul
  - Une approche nécessairement pluridisciplinaire
    - Système, réseaux, sécurité, applications, …
  - Ne pas seulement encourager les aspects purement calcul…
- Bien identifier ce qui est technologique
  - Les effets de mode...

  De ce qui est plus fondamental
  - Les nouveaux concepts...
- Encourager toutes recherches visant à utiliser des réseaux à grande échelle pour des applications innovantes…
  - Ne pas se concentrer sur une seule approche

Quelques remarques

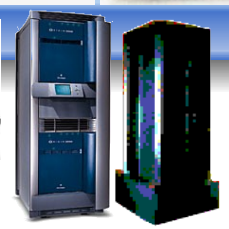
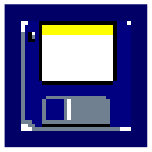
- Il existe d'autres programmes du MR pour l'utilisation des grilles en milieu industriel
- Importance de la coopération européenne et internationale dans ce domaine
- Importance des standards
- Renforcer la recherche dans les domaines du logiciel et dans les nouvelles applications
- http://www-sop.inria.fr/aci/grid

Cluster of Clusters (Netherlands)



What is Grid ?

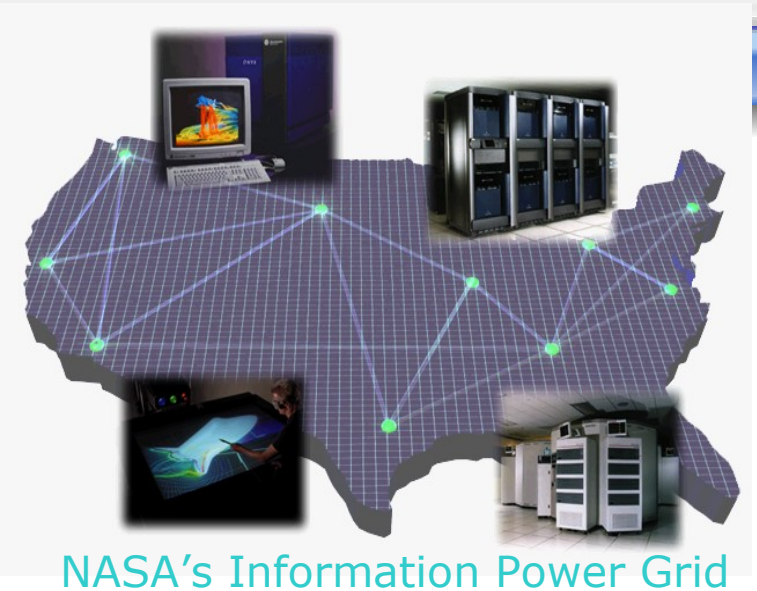


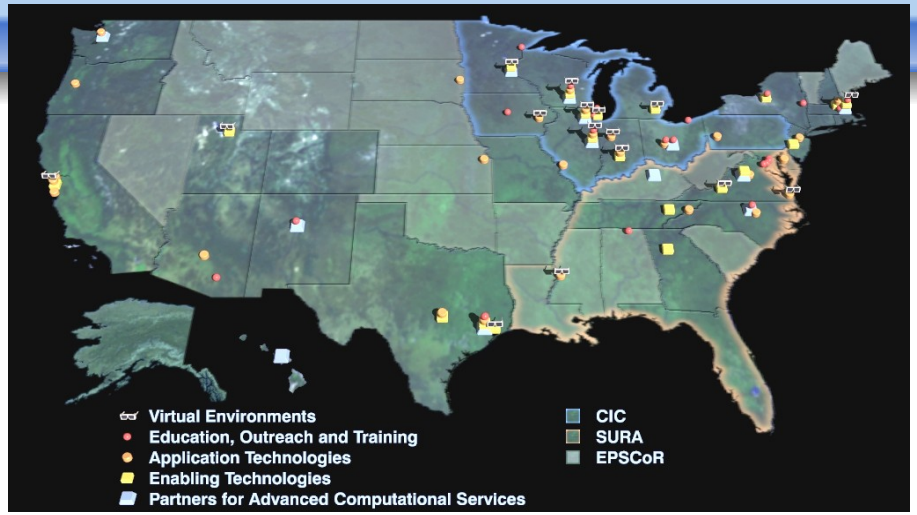



- An infrastructure that couples
  - Computers (PCs, workstations, clusters, traditional supercomputers, and even laptops, notebooks, mobile computers, PDA, and so on)
  - Software
  - Databases (e.g., transparent access to human genome database)
  - Special Instruments (e.g., radio telescope--SETI@Home Searching for Life in galaxy, Austrophysics@Swinburne for pulsars)
  - People (may be even animals who knows ?;-)
- across the local/wide-area networks (enterprise, organisations, or Internet) and presents them as an unified integrated (single) resource.

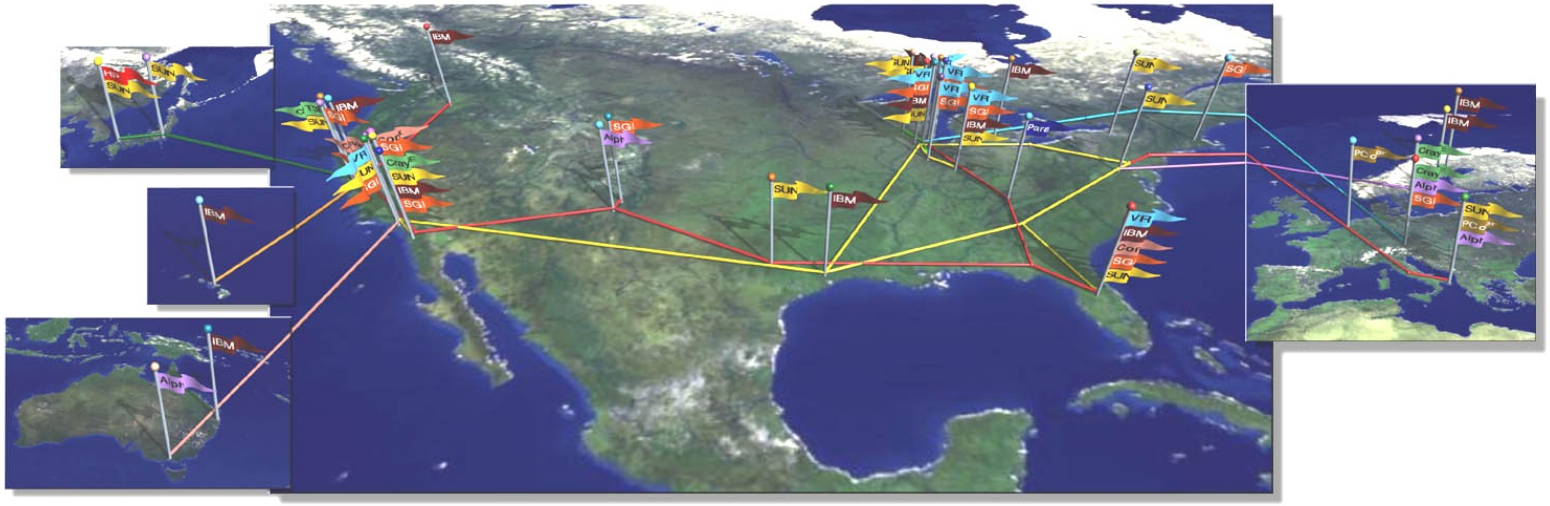
Production Grids & Testbeds



NASA's Information Power Grid



The Alliance National Technology Grid



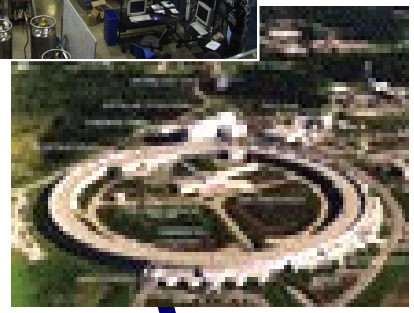
GUSTO Testbed

Why “The Grid”?

- New applications based on high-speed coupling of people, computers, databases, instruments, etc.
  - Online instruments
  - Collaborative engineering
  - Parameter studies
  - Browsing of remote datasets
  - Use of remote software
  - Data-intensive computing
  - Very large-scale simulation

Online Instruments

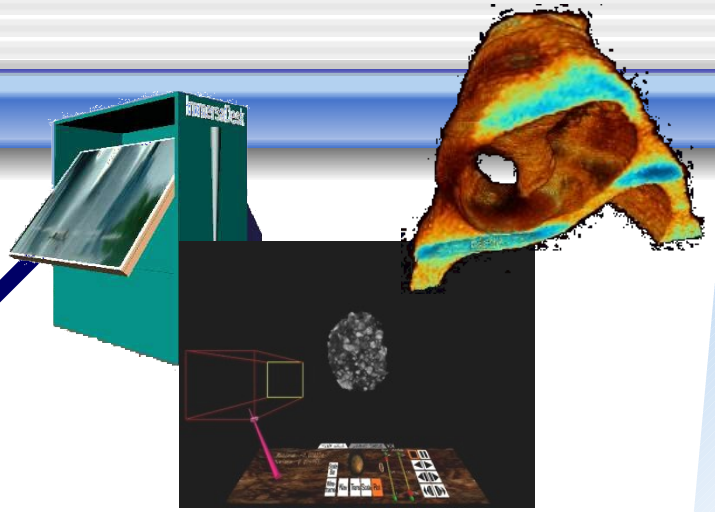
Advanced Photon Source



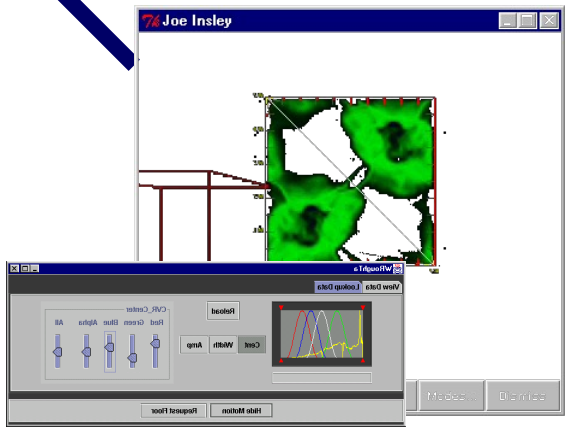
real-time collection

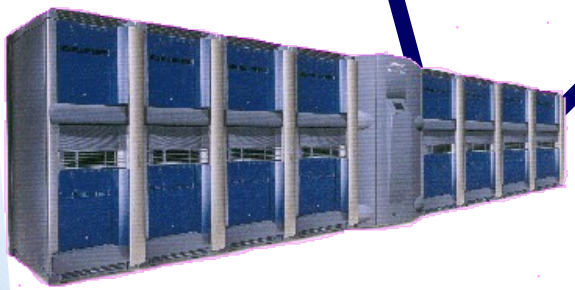
wide-area dissemination

archival storage

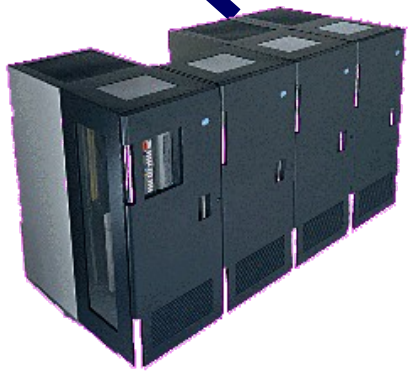


desktop & VR clients with shared controls



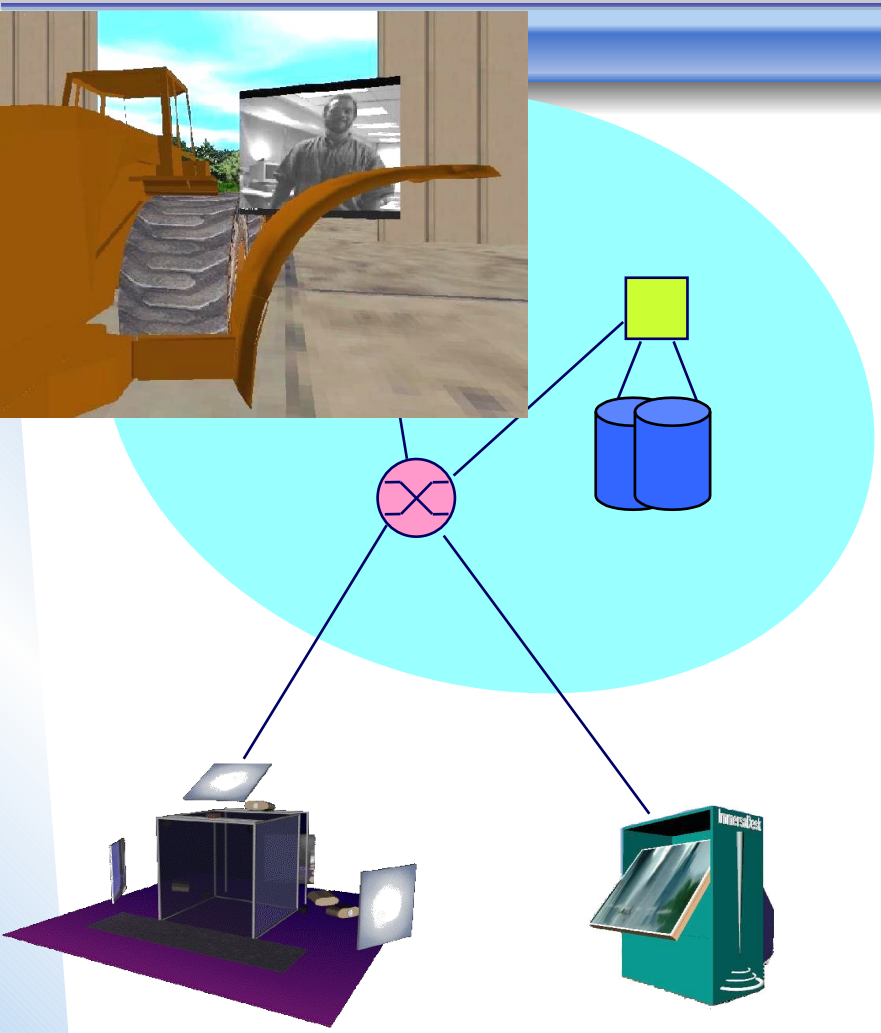


tomographic reconstruction



DOE X-ray source grand challenge: ANL, USC/ISI, NIST, U.Chicago

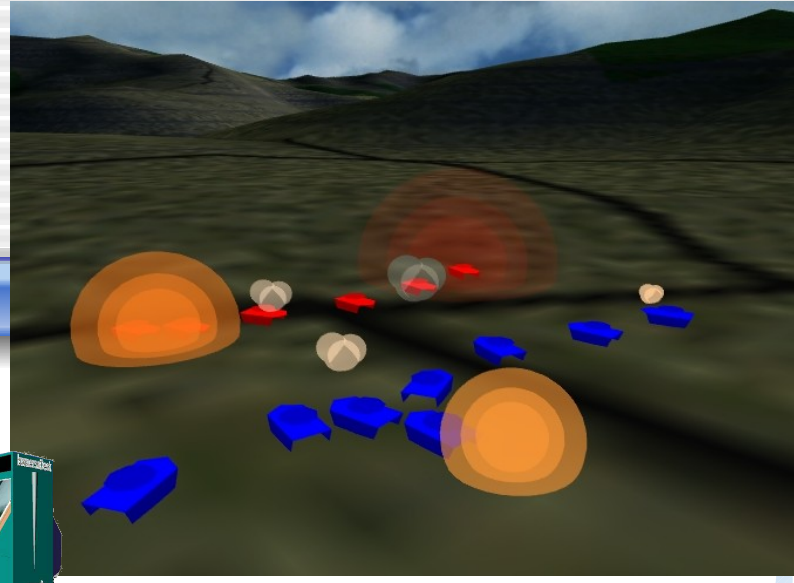
Collaborative Engineering

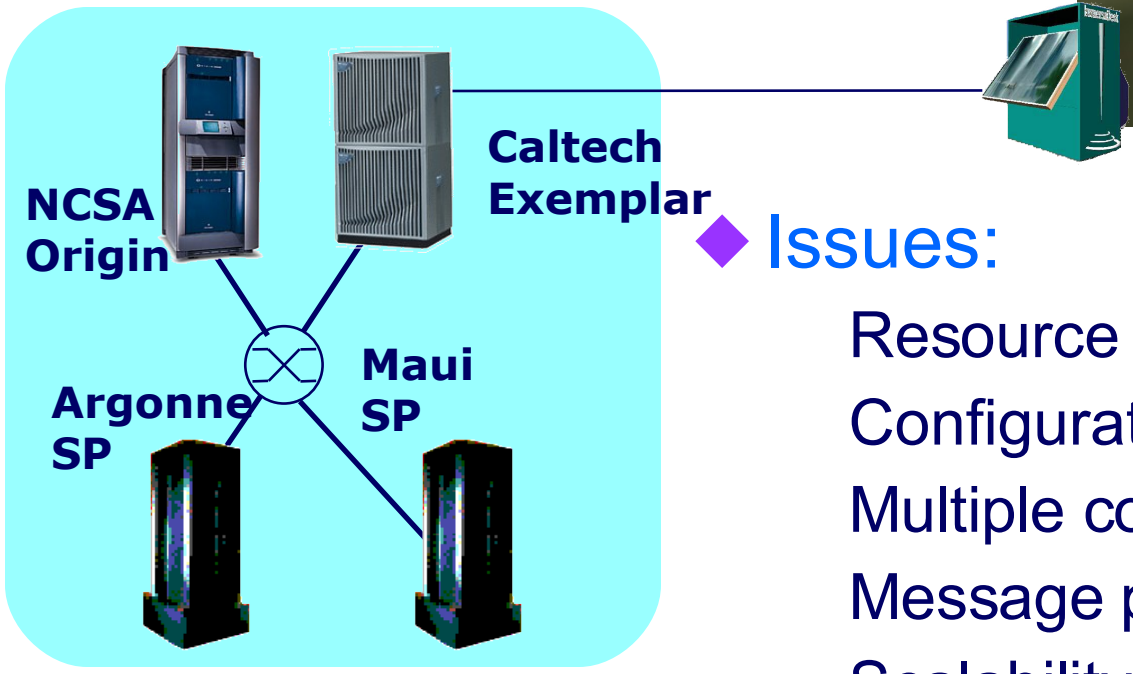


- Manipulate shared virtual space, with
  - Simulation components
  - Multiple flows: Control, Text, Video, Audio, Database, Simulation, Tracking, Haptics, Rendering
- Issues:
  - (un)reliable uni/multicast
  - Security
  - Reservation & QoS

CAVERNsoft: UIC, Electronic Visualization Laboratory

Distributed Supercomputing





- Issues:
  - Resource discovery, scheduling
  - Configuration
  - Multiple comm methods
  - Message passing (MPI)
  - Scalability
  - Fault tolerance

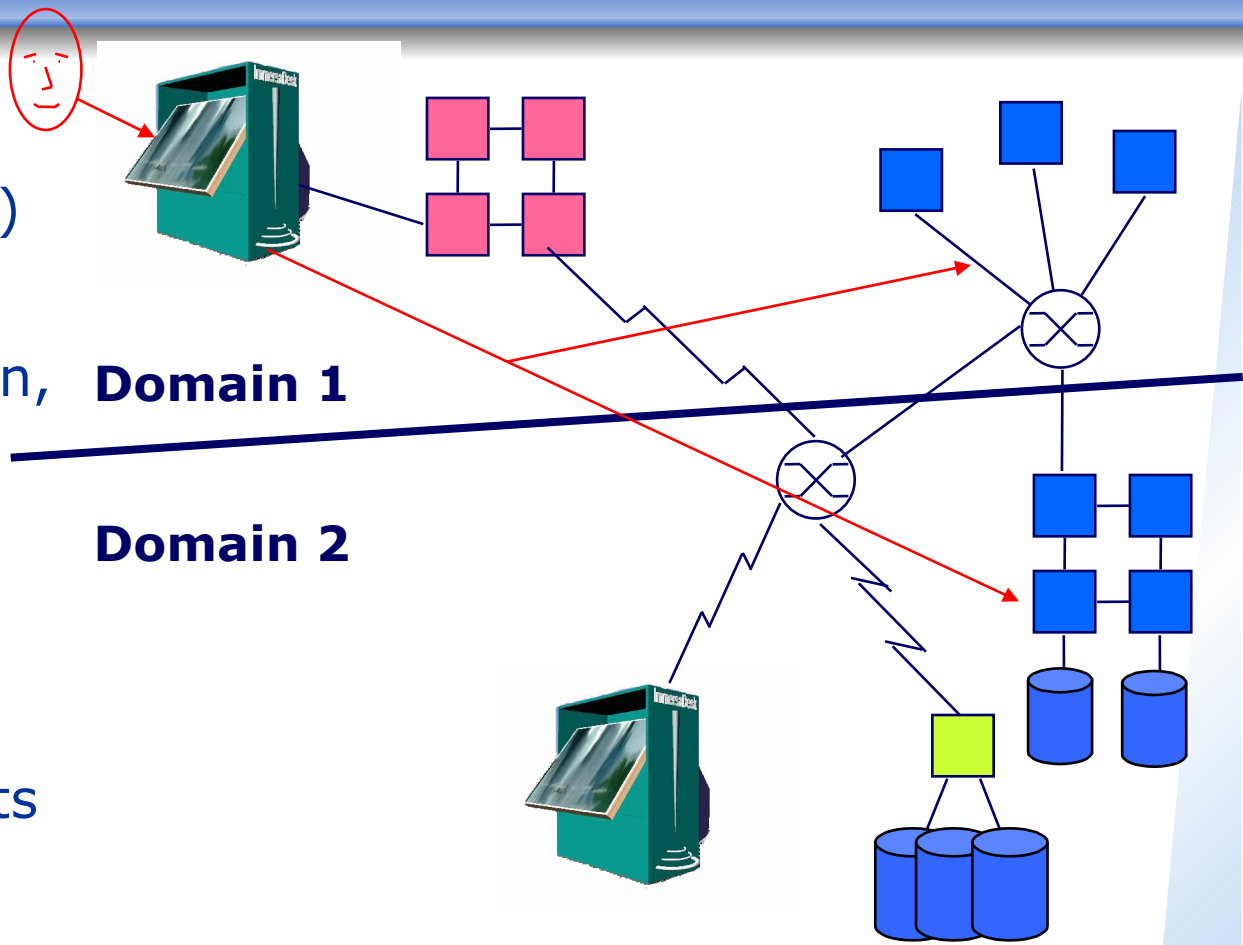
SF-Express Distributed Interactive Simulation: Caltech, USC/ISI

Technical Challenges

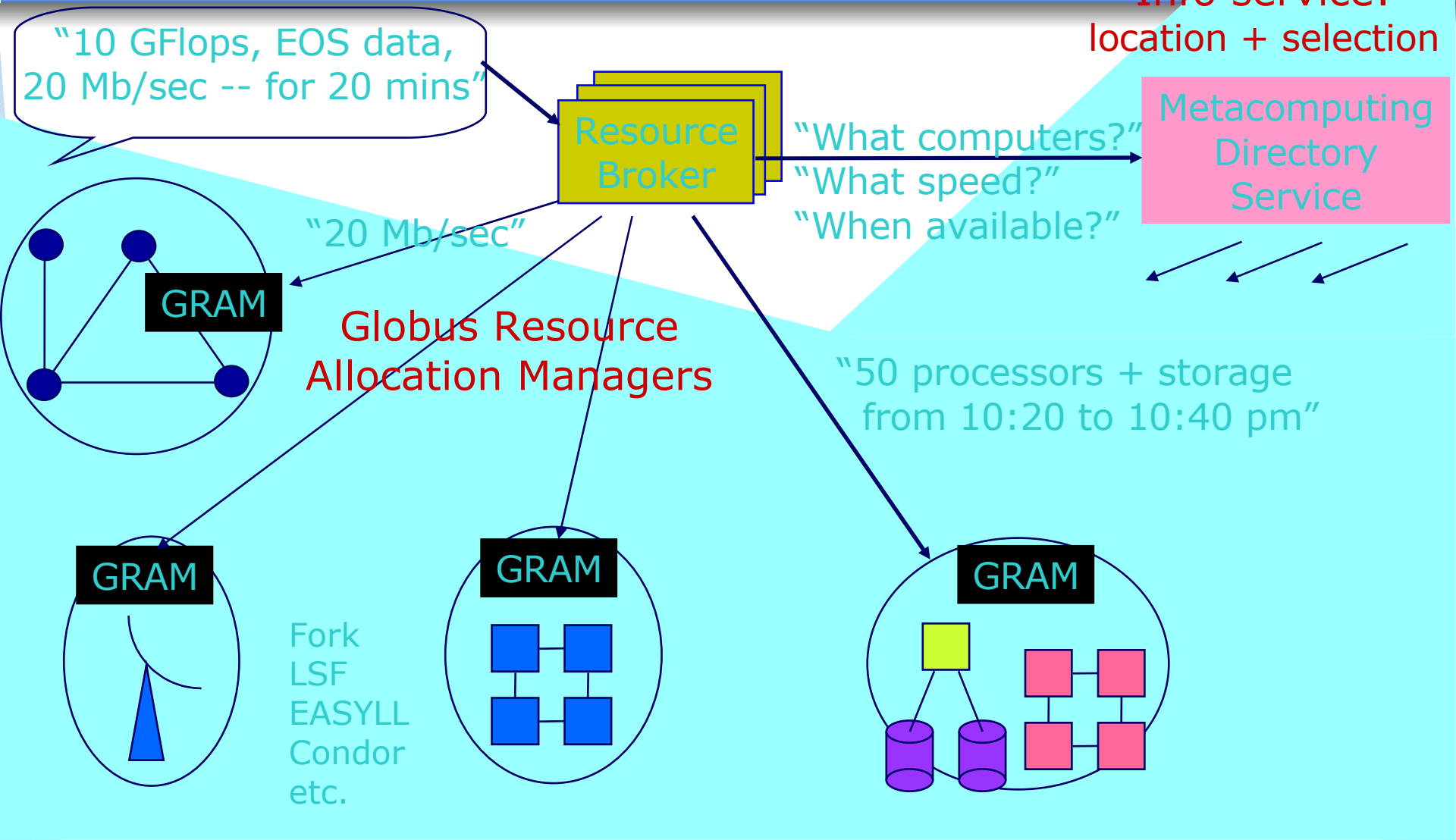
- Complex application structures, combining aspects of parallel, multimedia, distributed, collaborative computing
- Dynamic varying resource characteristics, in time and space
- Need for high & guaranteed “end-to-end” performance, despite heterogeneity and lack of global control
- Inter-domain issues of security, policy, payment

Issues

- Authenticate once
- Specify simulation (code, resources, etc.)
- Locate resources
- Negotiate authorization, acceptable use, etc.
- Acquire resources
- Initiate computation
- Steer computation
- Access remote datasets
- Collaborate on results
- Account for usage



Resource Management Architecture



Architectural Approaches

- Distributed systems: DCE, CORBA, Jini, etc.
  - Rich functionality eases app development
  - Complexity hinders deployment
    - especially in absence of global control
  - Performance difficulties
- Internet/Web Protocols and Tools
  - Simple protocols facilitate deployment
  - Missing functionality hinders app development
  - Performance difficulties

The Globus Project

- Basic research in grid-related technologies
  - Resource & data management, security, QoS, policy, communication, adaptation, etc.
- Development of Globus Toolkit
  - Core services for grid-enabled tools & apps
- Construction of production grids & testbeds
  - Multiple deployments to distributed organizations for production & prototyping
- Application experiments
  - Distributed applications, tele-immersion, etc.

Globus Project Participants

- Globus Project is a large community effort
  - Globus Toolkit core development
    - Argonne, USC/ISI, NCSA, SDSC
  - Globus Toolkit contributors
    - NASA, DOE ASCI DRM (SNL, LBNL, LLNL), Raytheon, and numerous others
  - Collaborators
    - University, lab, industrial, and international partners spanning many scientific and engineering disciplines
- Active in Grid Forum
  - http://www.gridforum.org

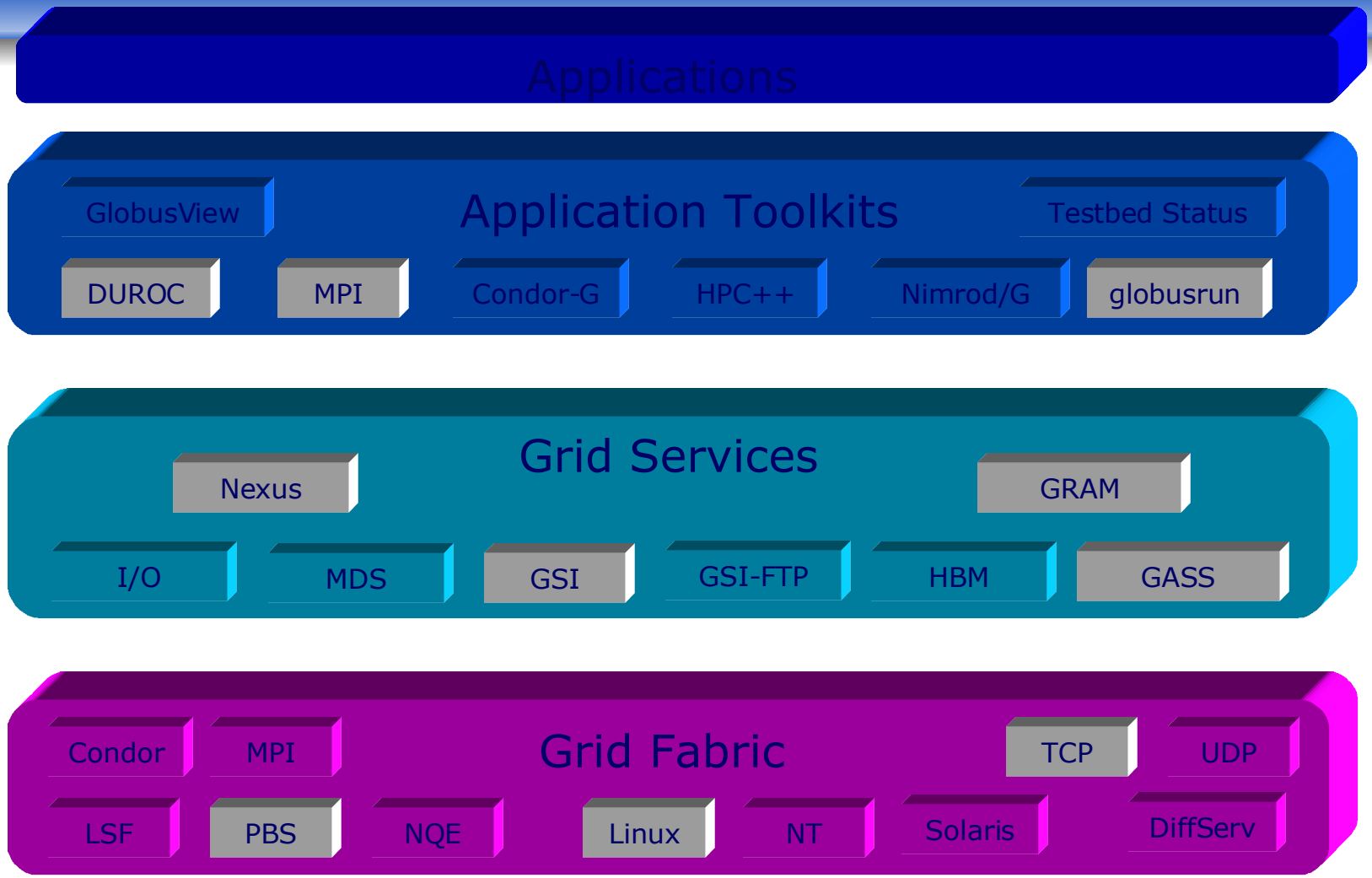
Globus Approach

- A toolkit and collection of services addressing key technical problems
  - Modular “bag of services” model
  - Not a vertically integrated solution
  - General infrastructure tools (aka middleware) that can be applied to many application domains
- Inter-domain issues, rather than clustering
  - Integration of intra-domain solutions
- Distinguish between local and global services

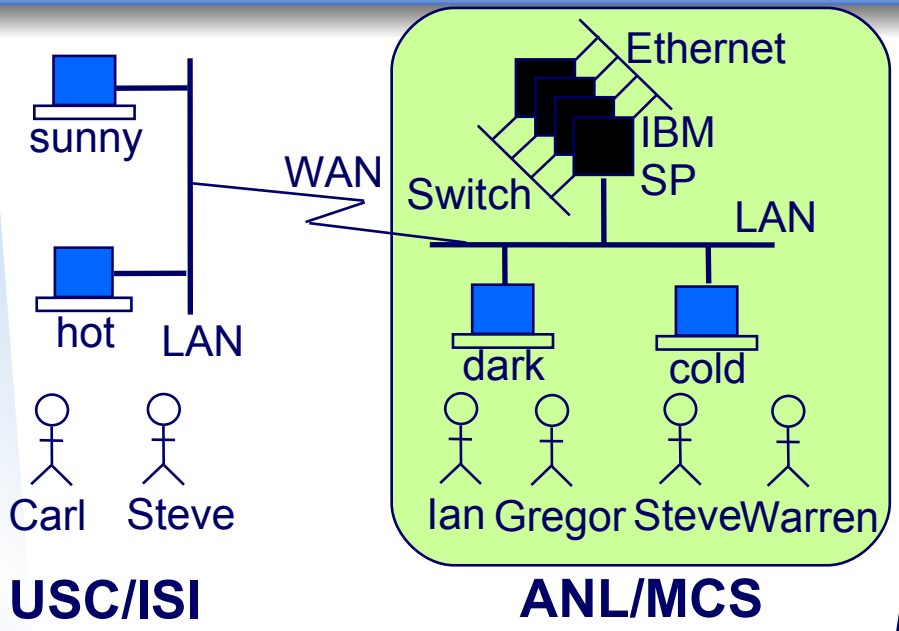
Globus Toolkit Grid Services

- Security (GSI)
- Resource management (GRAM)
- Information services (MDS)
- Remote file management (GASS)
- Communication (I/O, Nexus)
- Process monitoring (HBM)

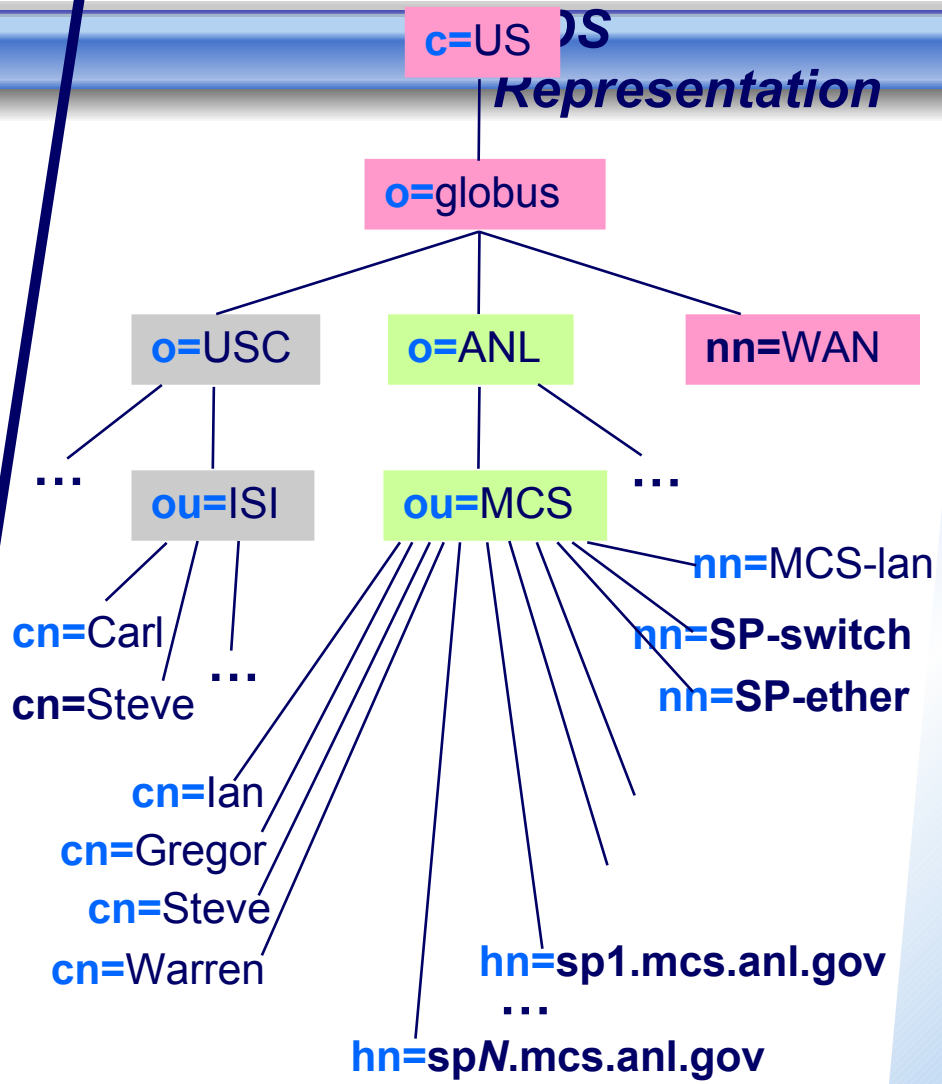
Layered Architecture

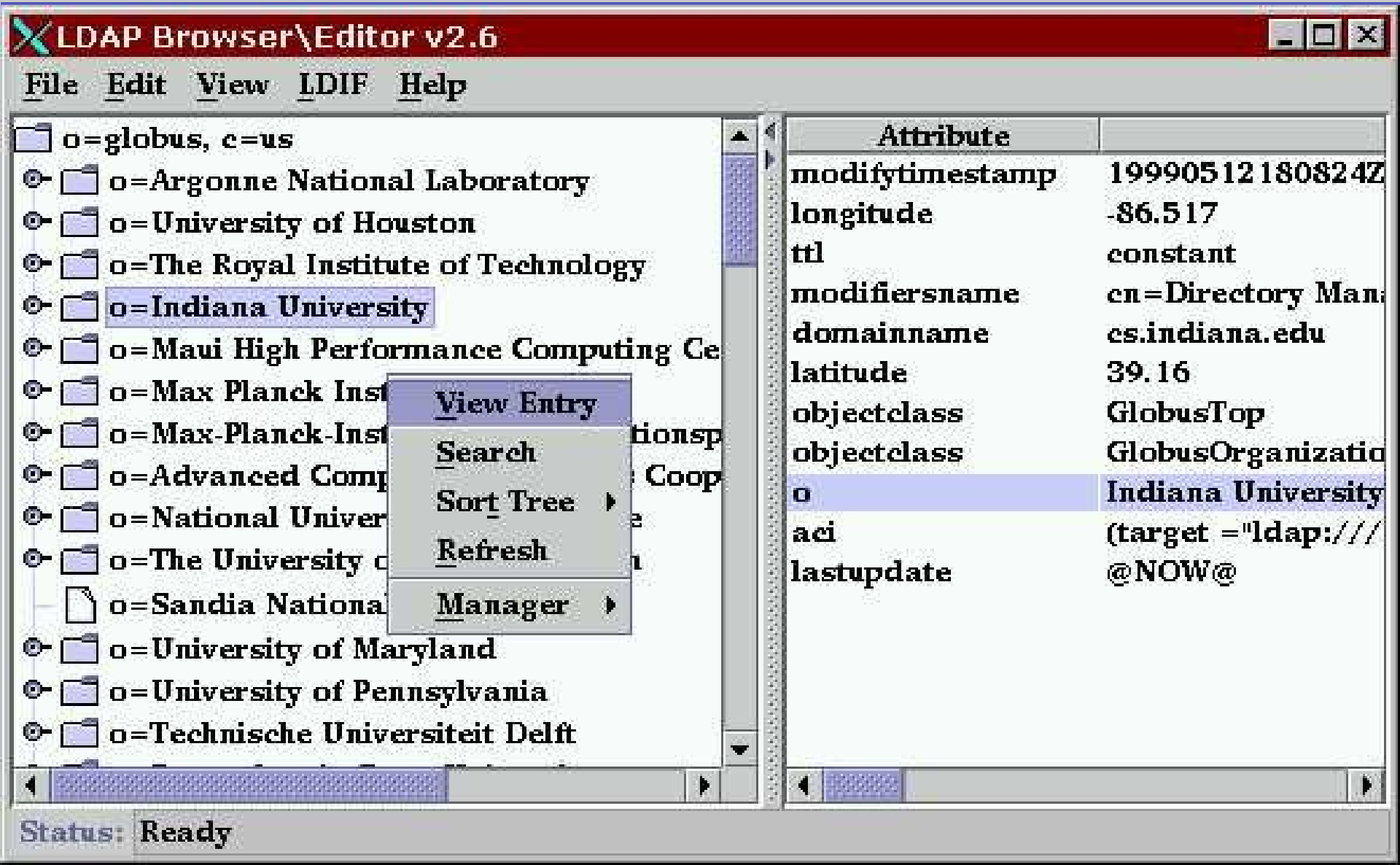


Metacomputing directory service



Physical Structure




LDAP Browser\Editor v2.6
File Edit View LDIF Help
o=globus, c=us
o=Argonne National Laboratory
o=University of Houston
o=The Royal Institute of Technology
o=Indiana University
o=Maui High Performance Computing Ce
o=Max Planck Inst
o=Max-Planck-Inst
o=Advanced Comp
o=National Univer
o=The University c
o=Sandia Nationa
o=University of Maryland
o=University of Pennsylvania
o=Technische Universiteit Delft
View Entry
Search
Sort Tree
Refresh
Manager
Attribute
modifytimestamp 19990512180824Z
longitude -86.517
ttl constant
modifiersname cn=Directory Mana
domainname cs.indiana.edu
latitude 39.16
objectclass GlobusTop
objectclass GlobusOrganizatio
o Indiana University
aci (target ="ldap:///
lastupdate @NOW@
Status: Ready