Propos sur les gestionnaires de tâches et de ressources (*Batch Scheduler*)

Olivier Richard (Mdc)

Laboratoire d'Informatique de Grenoble (LIG)
Equipe-Projet INRIA Mescal

6 octobre 2009

Notre expérience

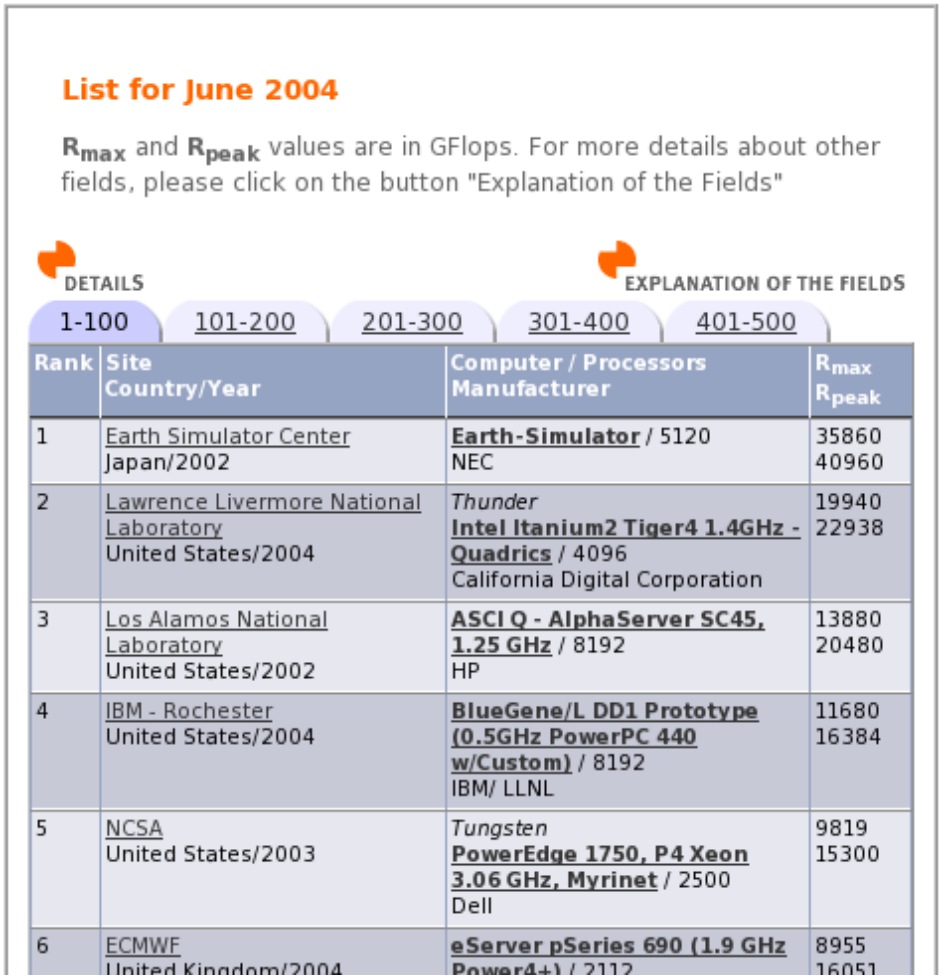
- OAR : Gestionnaire de ressource
- Kadeploy : Outils de déploiment
- Cigri : Gestionnaire pour grille légère

- CIMENT : Grappe de production
- Grid'5000 : Plate-forme distribuée dédiée à l'expérimentation

Sommaire

Top 500 (www.top500.org)

- **1** BlueGene/L, 280.6 TFlop/s, 131072 processeurs (juin 2007)
- **500** 896 processeurs, 4 TFlop/s
- La majorité des grappes possèdent plus de 1024 processeurs



List for June 2004

R_{max} and R_{peak} values are in GFlops. For more details about other fields, please click on the button "Explanation of the Fields"

DETAILS EXPLANATION OF THE FIELDS

1-100 | 101-200 | 201-300 | 301-400 | 401-500

Rank	Site Country/Year	Computer / Processors Manufacturer	R_{max} R_{peak}
1	Earth Simulator Center Japan/2002	**Earth-Simulator** / 5120 NEC	35860 40960
2	Lawrence Livermore National Laboratory United States/2004	*Thunder* **Intel Itanium2 Tiger4 1.4GHz - Quadrics** / 4096 California Digital Corporation	19940 22938
3	Los Alamos National Laboratory United States/2002	**ASCI Q - AlphaServer SC45, 1.25 GHz** / 8192 HP	13880 20480
4	IBM - Rochester United States/2004	**BlueGene/L DD1 Prototype (0.5GHz PowerPC 440 w/Custom)** / 8192 IBM/ LLNL	11680 16384
5	NCSA United States/2003	*Tungsten* **PowerEdge 1750, P4 Xeon 3.06 GHz, Myrinet** / 2500 Dell	9819 15300
6	ECMWF United Kingdom/2004	**eServer pSeries 690 (1.9 GHz Power4+)** / 2112	8955 16051

Evolutions des grappes (clusters)

- Démocratisation
- Densification
  - Nombre de processeurs en augmentation
  - Nombre de coeurs (bi-processeurs / bi-coeurs) x4, x8 ...
  - **la puissance électrique**
- Consommation électrique

Les grappes au quotidien

Des utilisateurs et des programmes :

- Utilisateurs avec une connaissance très variable des aspects systèmes / gestion des ressources
- Les tâches à exécuter sont variées (nombre, taille, durée...)

Les ressources reste(ro)nt limitées Rôles de l'administrateur :

- Aider les utilisateurs à exploiter les ressources de calcul (et de stockage)
- Maintenir un bon niveau d'utilisation de(s) grappe(s)

Nécessité d'un gestionnaire de tâches et de ressources

Organiser/répartir manuellement les ressources entre les utilisateurs et leurs tâches à traiter est réaliste qu'à **petite échelle**, moins de 10 utilisateurs et peu de tâches en concurrence (*agenda partagé*, *mailing-list*).

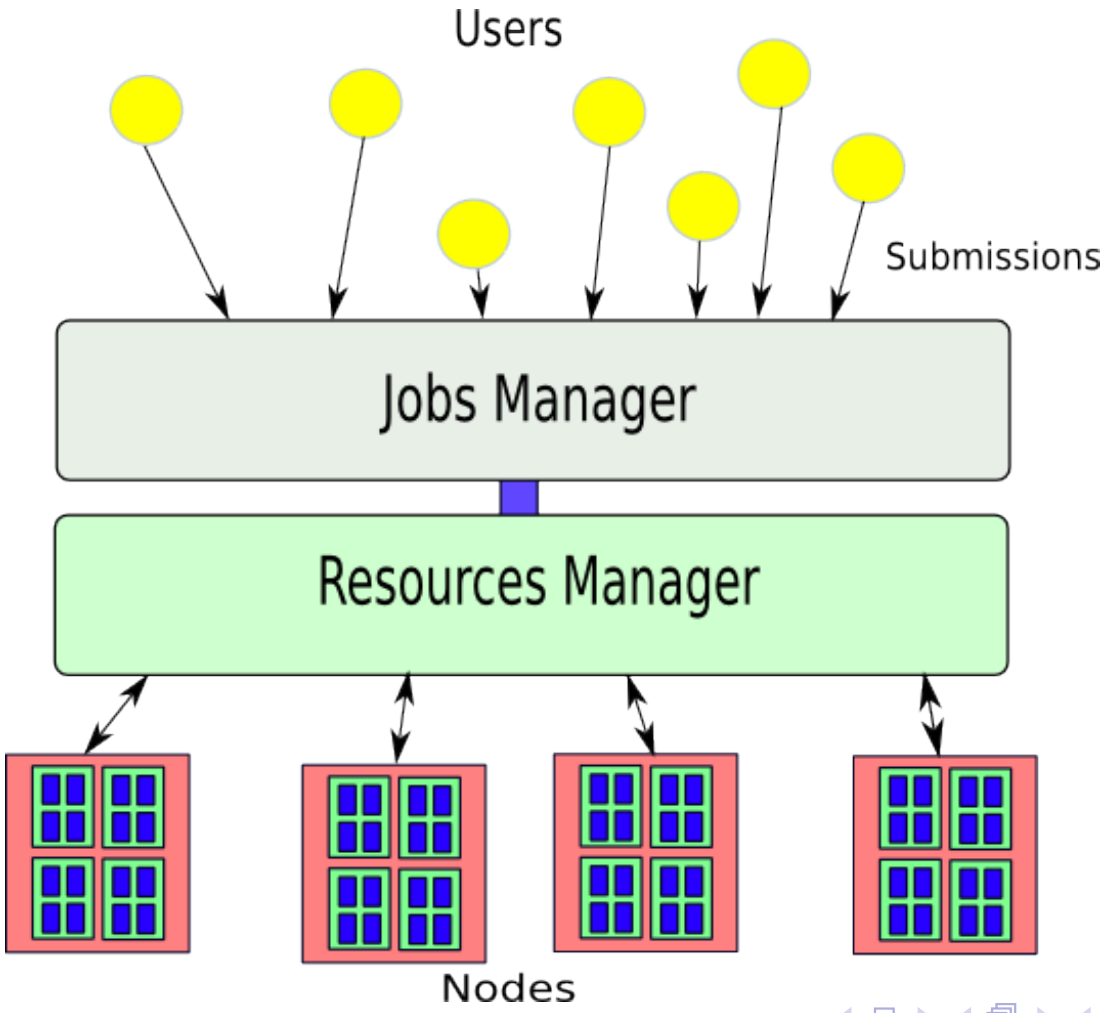
A **moyenne et grande échelle** on utilise un gestionnaire de ressource

- gére l'attribution des ressources aux tâches suivant une politique préétablie
- fait le suivi de l'exécution des tâches
- surveille l'état des ressources

Attention : l'administrateur est toujours nécessaire ! !

Principe général

Dans leur version simple, séparation en 2 couches (parfois une 3éme Workload Managment) :



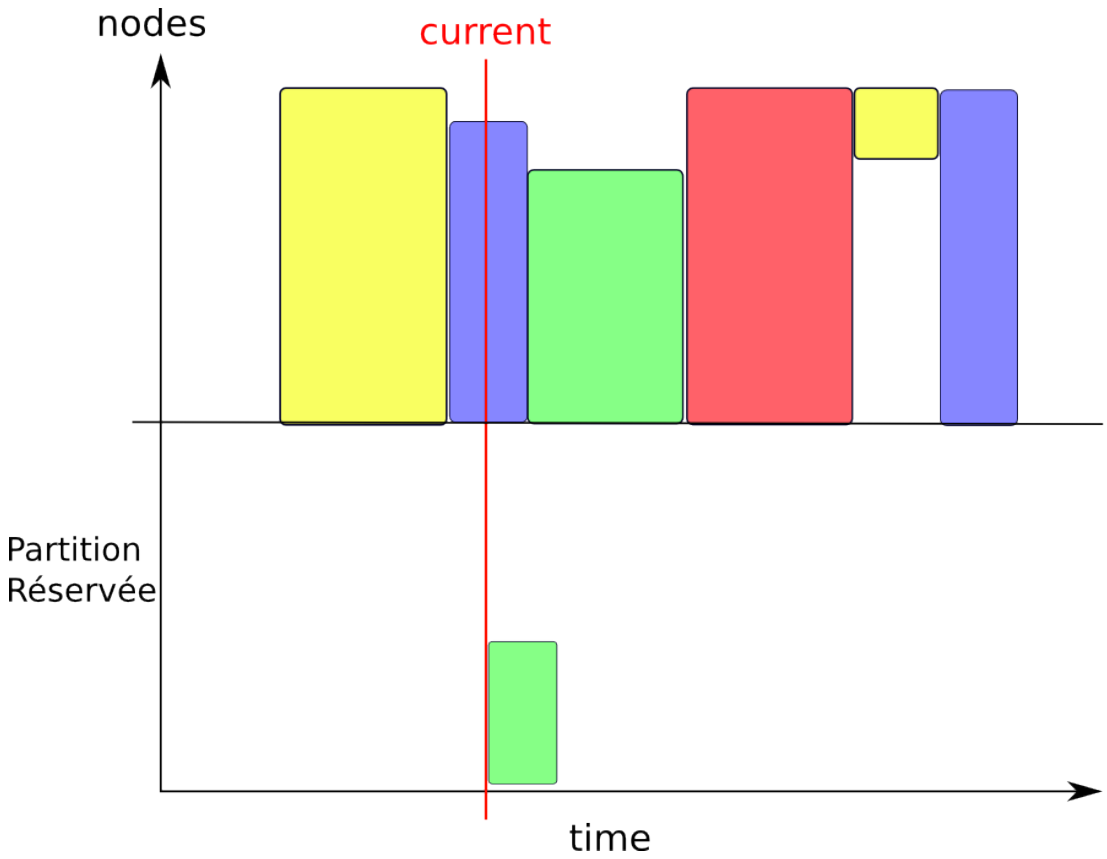
Mise en place

- Lors de l'installation de la machines par la société la fournissant.
- Les paramétrages initiaux peuvent convenir sur la durée de vie la machine
- Reparamétrages si :
  - La population d'utilisateur change
  - Les tâches à exécuter évoluent en nature
  - Mise-à-jour / ajout de matériel (exemple nouvelle tranche)

Important

L'installation et le paramétrage d'un gestionnaire suppose des échanges avec les utilisateurs et les adminstrateurs (réunion, information, formation, support). Il peut y avoir des compromis à déterminer (rendement/niveau de service)

Illustration du compromis rendement / temps de réponse



Les Gestionnaires de tâches et de ressources

Aussi appelés *Batch Scheduler*
Existent en très grand-nombre :

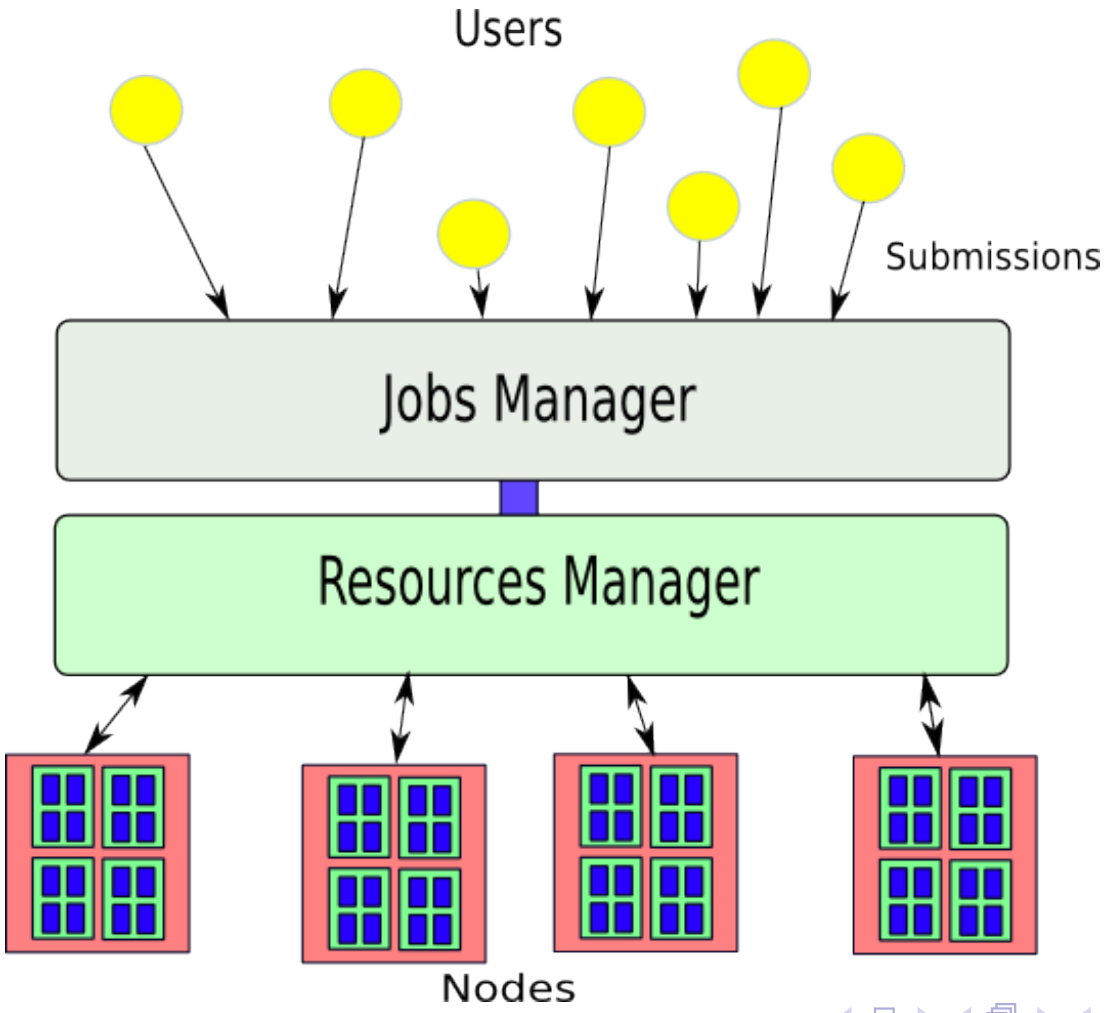
- **Condor**
- **Sun Grid Engine (SGE)**
- **MAUI/Torque**
- **Slurm**
- **OAR** (LIG/INRIA) :)
- LSF (Platform)
- PBS Pro
- Moab (Cluster Resources)
- Autres : **BQS (CC-IN2P3)**, **Lava**, Loadleveler, CCS...

- `http://en.wikipedia.org/wiki/Job_scheduler`

Note : Ici calcul haute-performance/grappe, mais utilisés dans d'autre domaine gestion/finance/rendu de film (enchainement de tâches).

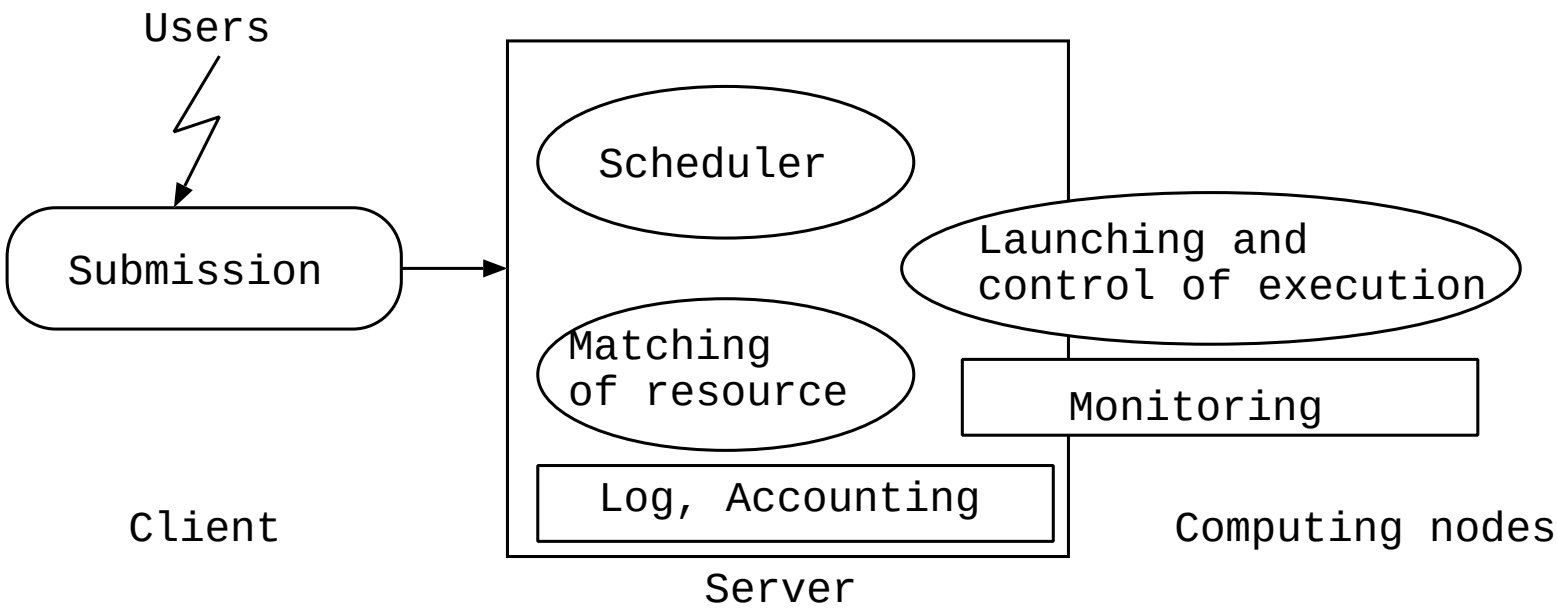
Principe général

Dans leur version simple, séparation en 2 couches (parfois une 3éme Workload Managment) :



Organisation générale

- Un serveur central
- Des programmes clients (en ligne de commandes) pour l'interaction avec les utilisateurs
- Une grande latitude dans le paramétrage



Fonctionnalités (1/2)

liste non-exhaustive

- Tâche (*soumission*) Interactive (shell) / Batch
- Tâche séquentielle et parallèle
- Walltime (temps limite). (**important pour l'ordonnancement)**
- Accès exéclusif / non-exclusif aux ressources
- Appariement de ressources
- Scripts Epilogue/Prologue (exécuter avant/après les tâches)
- Suivi (*monitoring* des tâches (consommation des ressources)
- Dépendance entre tâches (*workflow*)
- Logging et accounting
- Suspension/reprise des tâches

Fonctionnalités (2/2)

liste non-exhaustive

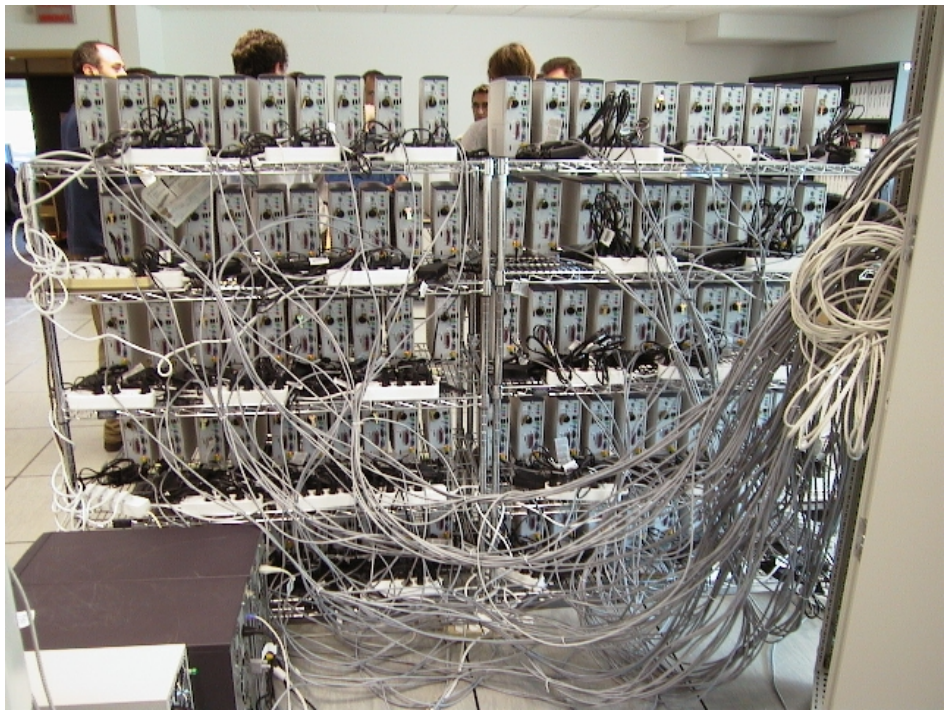
- Dépendance entre job
- Tableaux de tâches
- **Advance Reservation**
- **Expression des hiérarchies dans les requêtes**
- **Support de ressources de type différent (ex licence, capacité de stockage, capacité réseaux...)**
- **Tâche container** (*soumettre dans une tâche*)
- **tâche besteffort**
- **Type multiple de tâches** (besteffort, powersaving, deploy, timesharing, idempotent, **power**, cosystem ...) (élément important pour l'extension/l'adaptation)
- Tâches moldables
- First-Fit (Conservative Backfilling,)
- Fairsharing

OAR : un gestionnaire de taches et de ressources **polyvalent**

http ://oar.imag.fr/

OAR : Historique

- Début 2003 : Une machine dans le Top500 (225 noeuds), OpenPBS(Torque) est instable et difficile à faire évoluer
- PBSpro se comporte mieux (passage à l'échelle imparfait)
- Regle des 80/20 (20% des fonctionnalités utilisées dans 80 % des situations)



Objectifs

Un gestionnaire de tâches et de ressources **polyvalent** et **personnalisable**.

- Suivre l'évolution technologique (machine et infrastructure de plus en plus complexe)
- Adaptation aux différents contextes (cluster, cluster-on-demand, cluster virtuel, plate-forme pour l'expérimentation à la Grid'5000, *grand cluster*, besoin spécifique).

Sous-estimation

Regle des 80/20 : les 20% des fonctionnalités **ne sont pas les mêmes pour tous ! ! !**

OAR : principes de conception

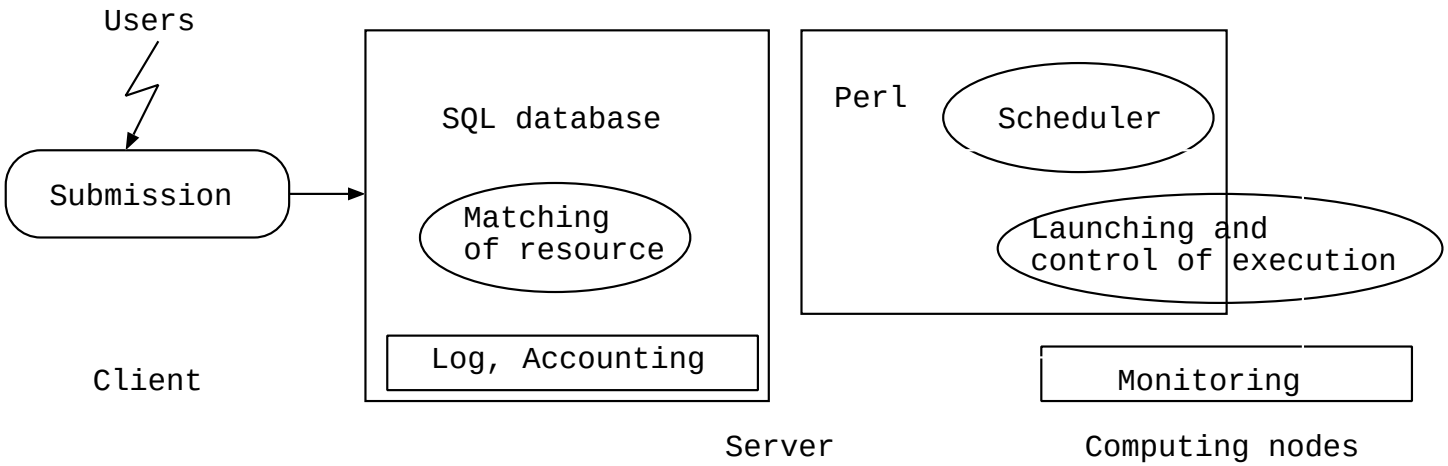
Utilisation de composants logiciels de haut niveau

- **Base de donnée relationnelle** (MySql/PostgreSQL) pour stocker et échanger :
  - Information sur les ressources et les tâches
  - L'état interne du système
- **Language(s) de script** (majoritairement Perl) pour le moteur d'exécution
  - Bien adapté pour les parties systèmes
  - Structures de haut niveau (listes, tables associatives, tris...)
  - Cycles de développement court
- **Autres composants** (Perl, Ruby, Caml)
  - **SSH, CPUSET** (confinement, nettoyage)
  - **Taktuk** lanceur lui aussi très polyvalent

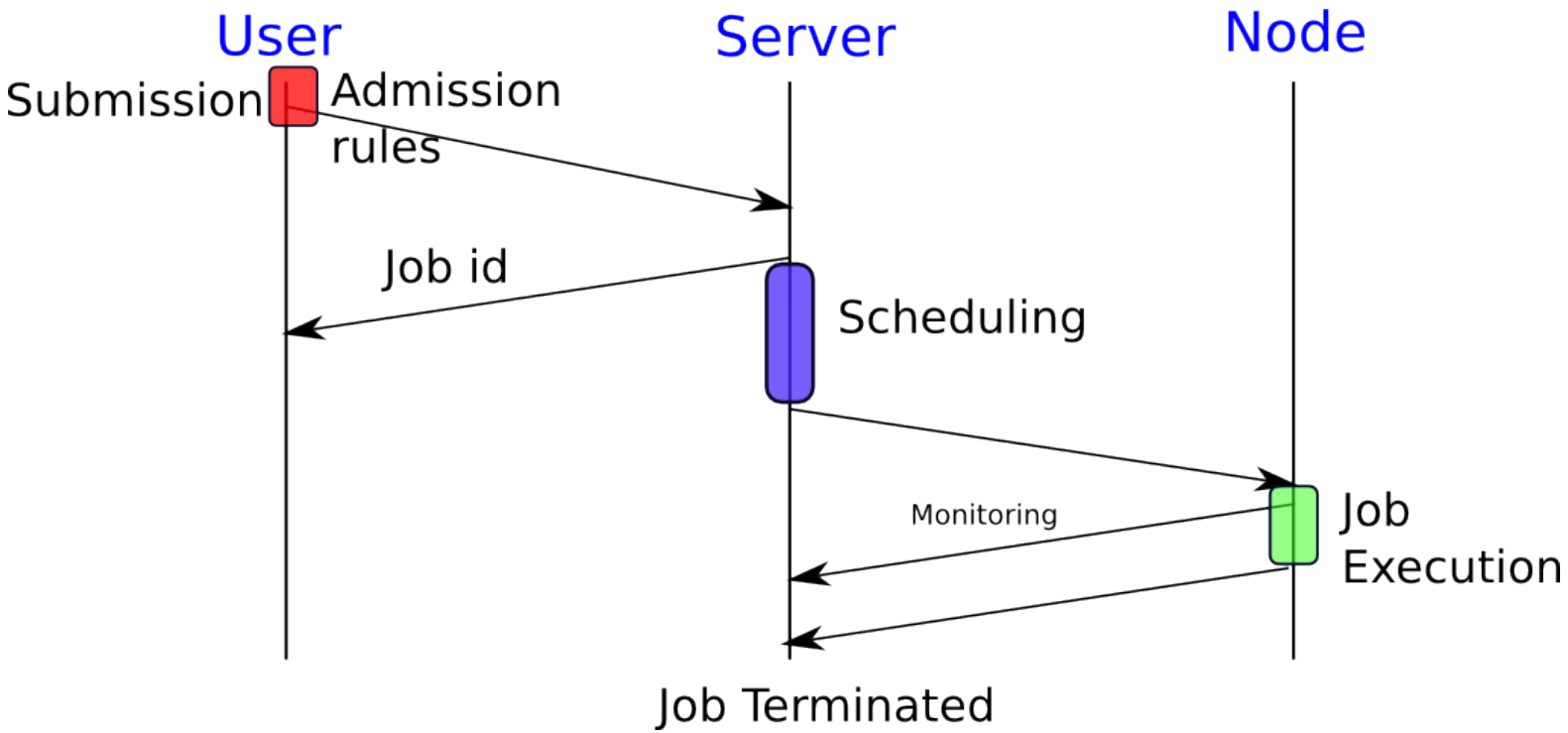
OAR : organisation générale

La base de donnée a un rôle central

- **l'état interne simplement accessible**
- le moteur est composé de **petit modules Perl**
- chaque module (= un script) peut-être facilement remplacé



Cycle de général



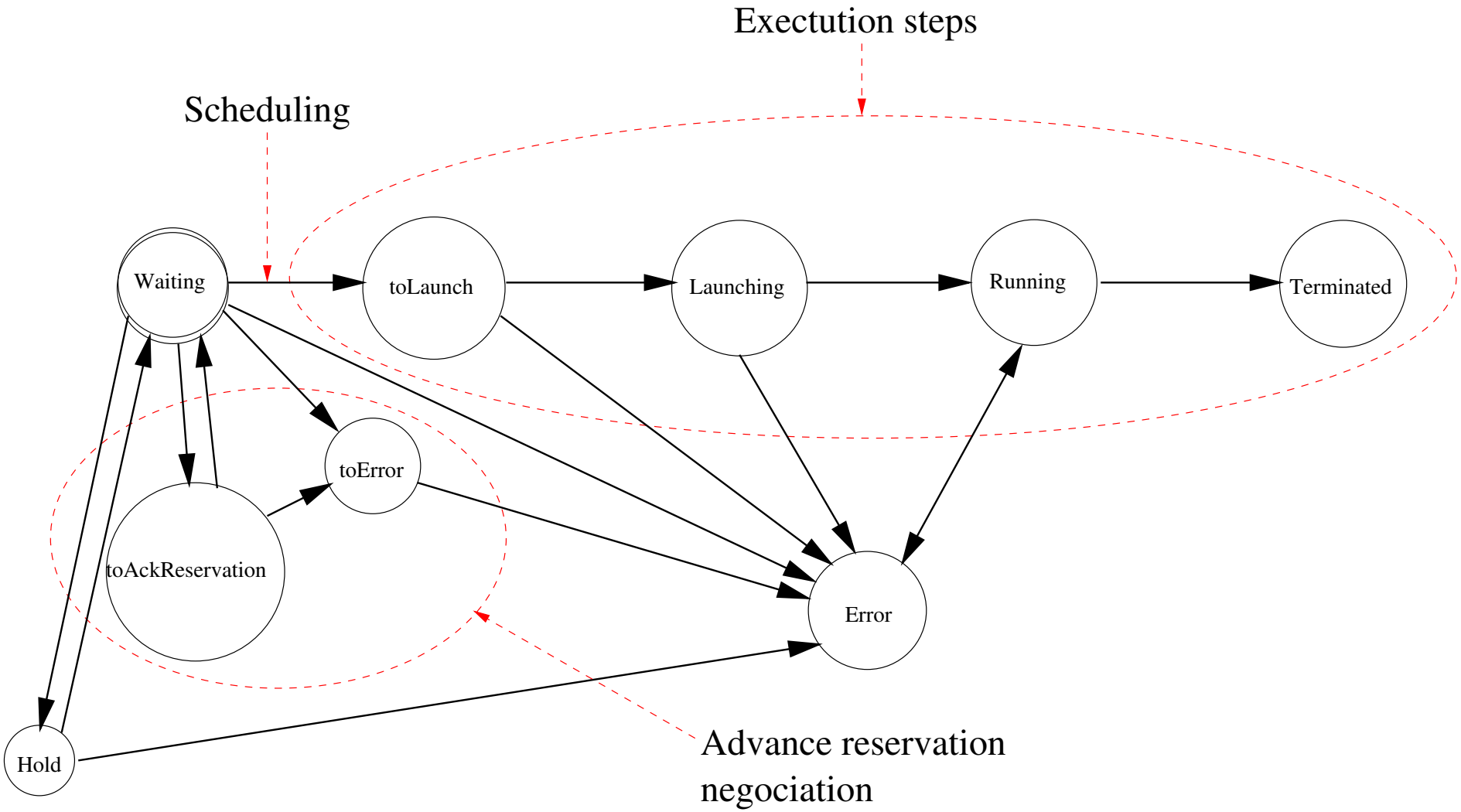
Régles d'admissions

Un point de paramétrage important

- **Cadrage** des réquêtes
- fixe des valeurs par défaut : walltime, queue, nombre de ressources demandées,
- contrôle d'accès (utilisateur, groupe, plage horaire...)
- point de *personnalisation* (au même titre que les scripts de prologue et d'épilogue)

Diagramme d'état d'une tâche

Exemple du système OAR (version 1.6)



Examples de soumission : OAR

Soumission pour tâche interactive : [1]

- **oarsub -I nodes=4 -i**

Soumission en *batch* (avec *walltime* et choix de queue) :

- **oarsub -q default -l walltime=2 :00,nodes=10 /home/toto/script**

Soumission d'une réservation :

- **oarsub -r "2008-04-27 11 :00" -l nodes=12**

Connection à une réservation (utilise le numéro de tâche) :

- **oarsub -C 154**

[1] **Note :** Chacune des commandes de soumission renseigne un numéro de tâche.

Ordonnancement

L'ordonnancement est l'étape [2] où le système choisi les **ressources à attribuées** aux tâches et **les dates de lancement**.

L'ordonnancement est défini suivant une **politique** qui se traduit par l'utilisation **d'algorithmes d'ordonnancement**.

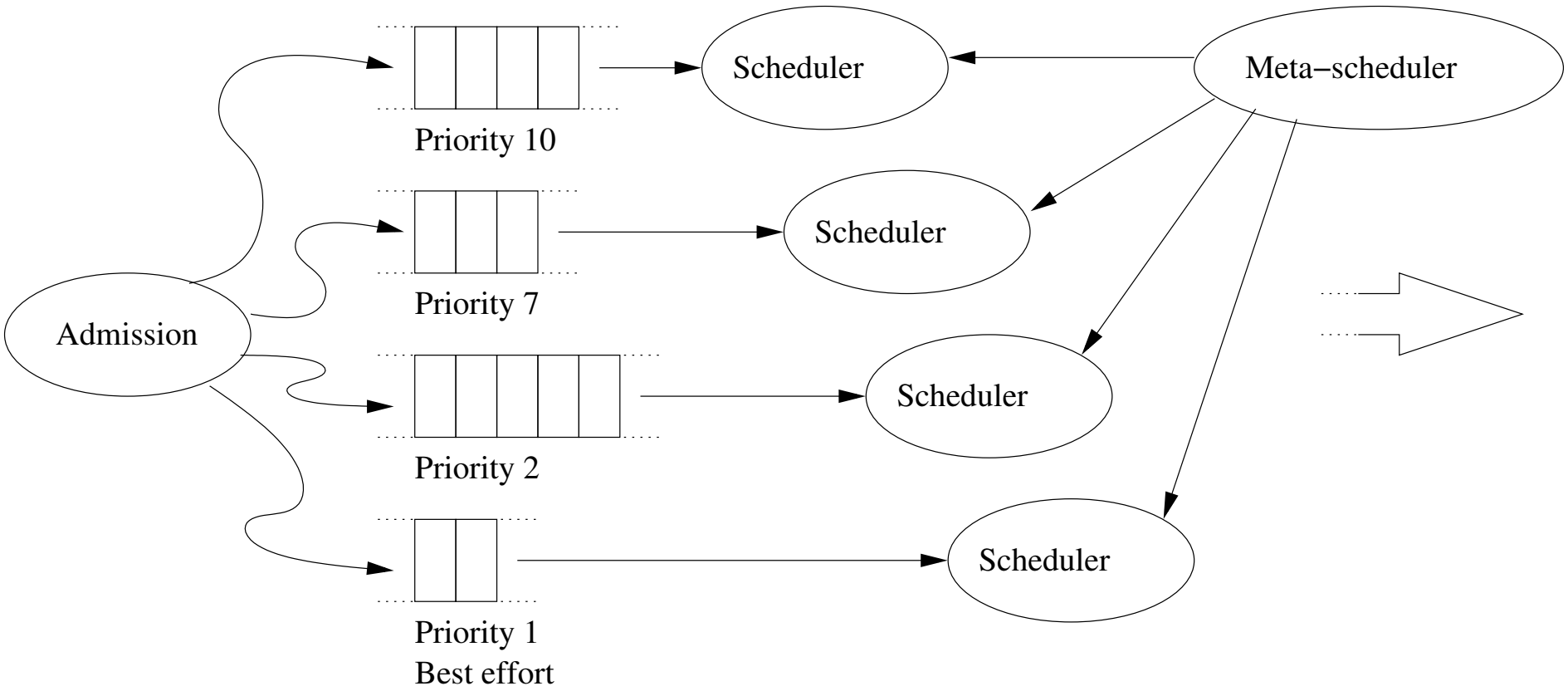
De plus de nombreux **critères et paramètres** sont utilisés pour guider et cadrer les allocations et les priorités.

[2] **Note :** l'ordonnancement est recalculé à chaque changement d'état (majeur) d'une tâche.

Organisation de l'ordonnancement

Gestion des tâches par file (queues)

- chaque file a une priorité
- chaque file a sa propre politique d'ordonnancement



Appariement de ressource / ressource matching

Une étape préliminaire à l'ordonnancement

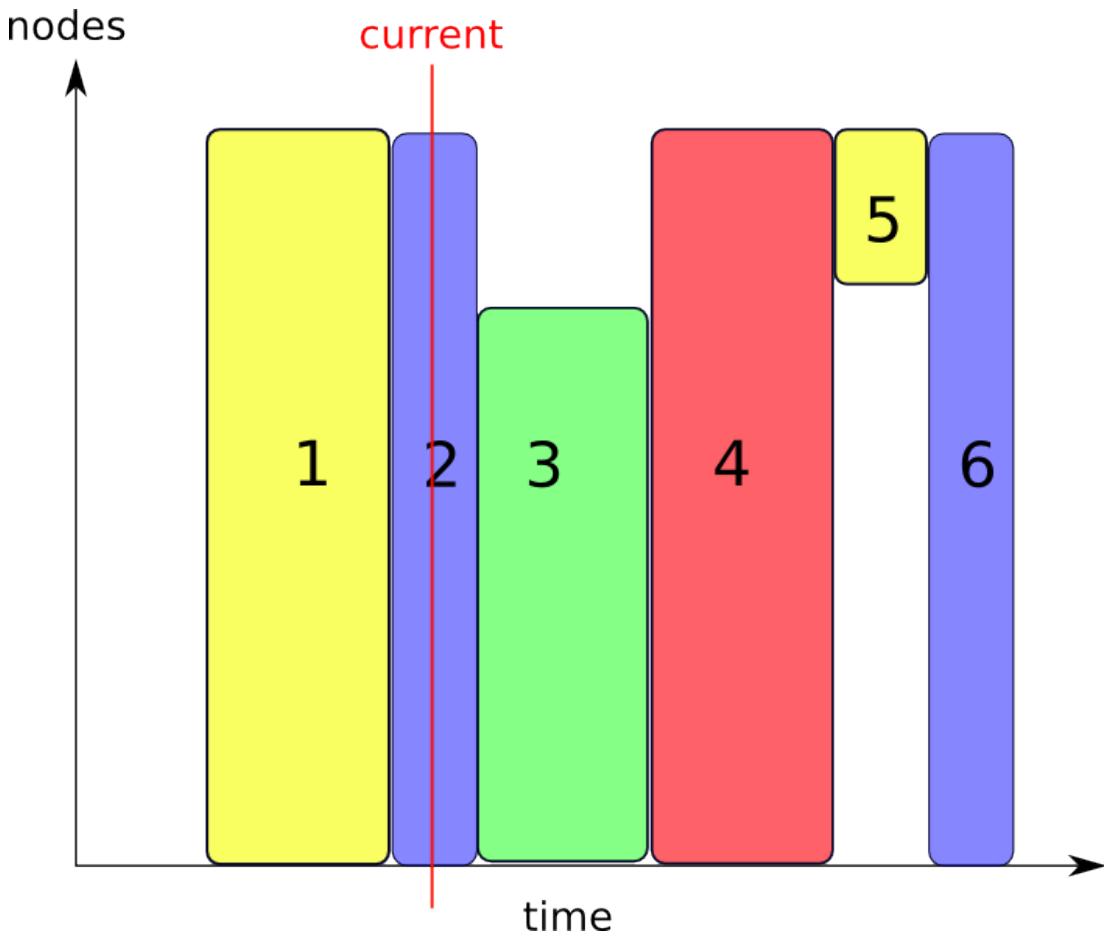
- **Filtrage** de resources
- **Classement** de ressource dans Condor
- Permet de spécifier des besoins particuliers
- mémoire, architecture, machine particulières, OS, niveau de charge...

Condor / ClassAds : Syntaxe, Attributs, Opérateurs, Classement (Ranking)

Politiques d'ordonnancement

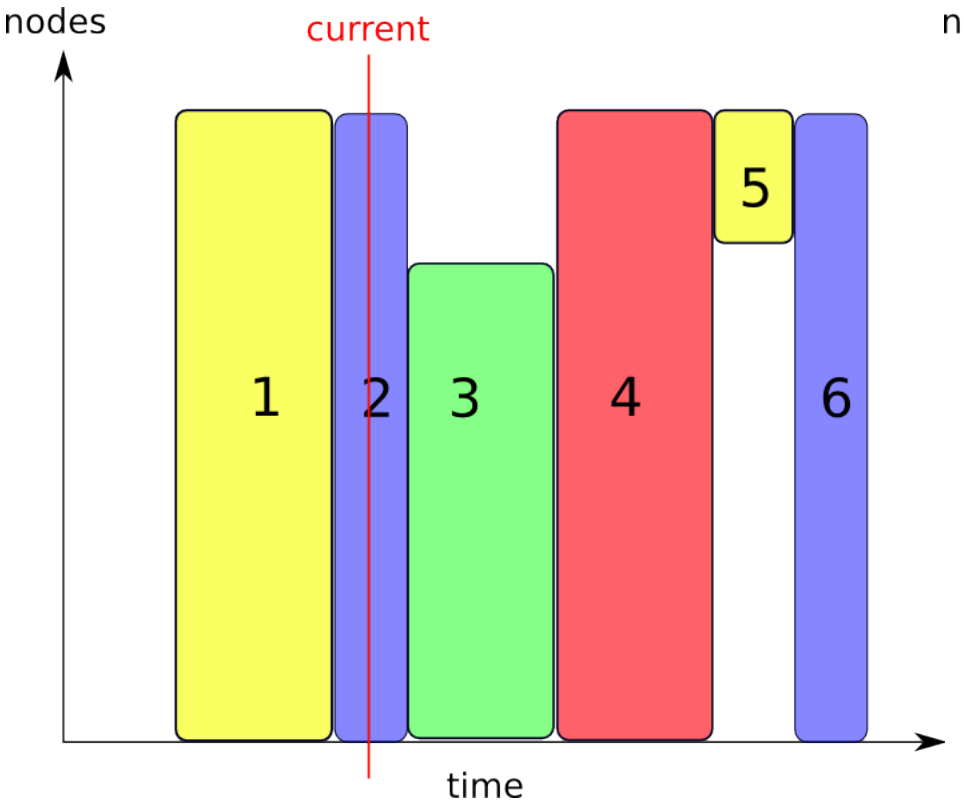
- FIFO (First-In First-Out)
- First-Fit (Backfilling)
- FairSharing
- Equilibrage de charge
- Récursivité
- SLA (Service Level Agrement)(Qualité de Sercice)

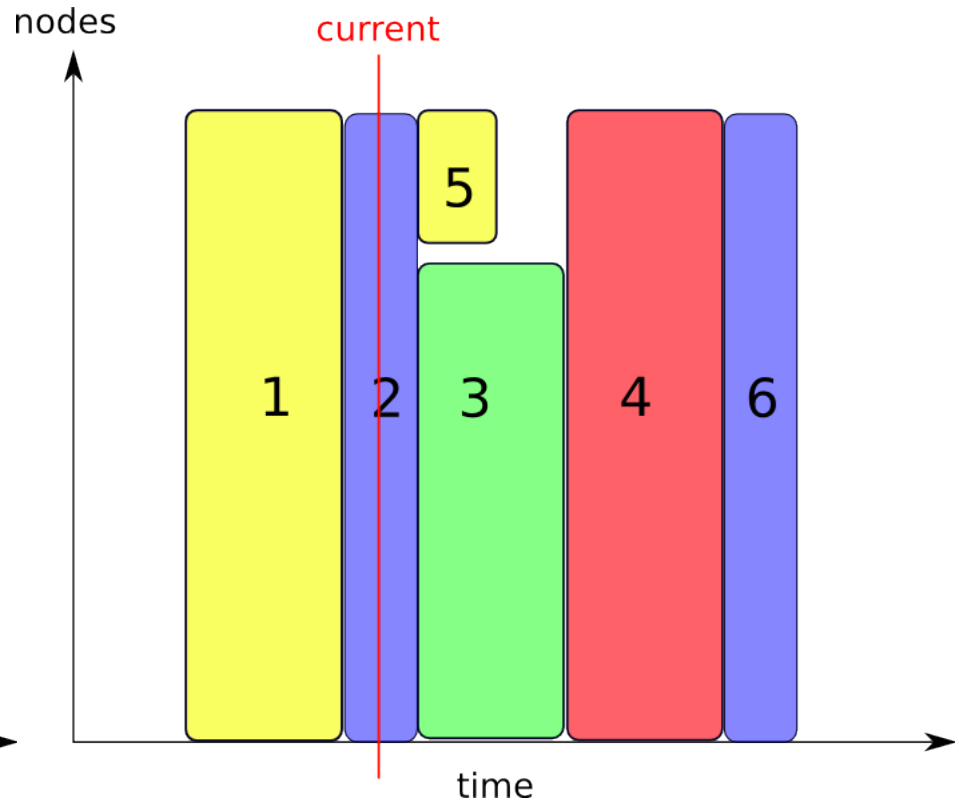
FIFO : Fisrt-In First-Out



First-Fit (Backfilling)

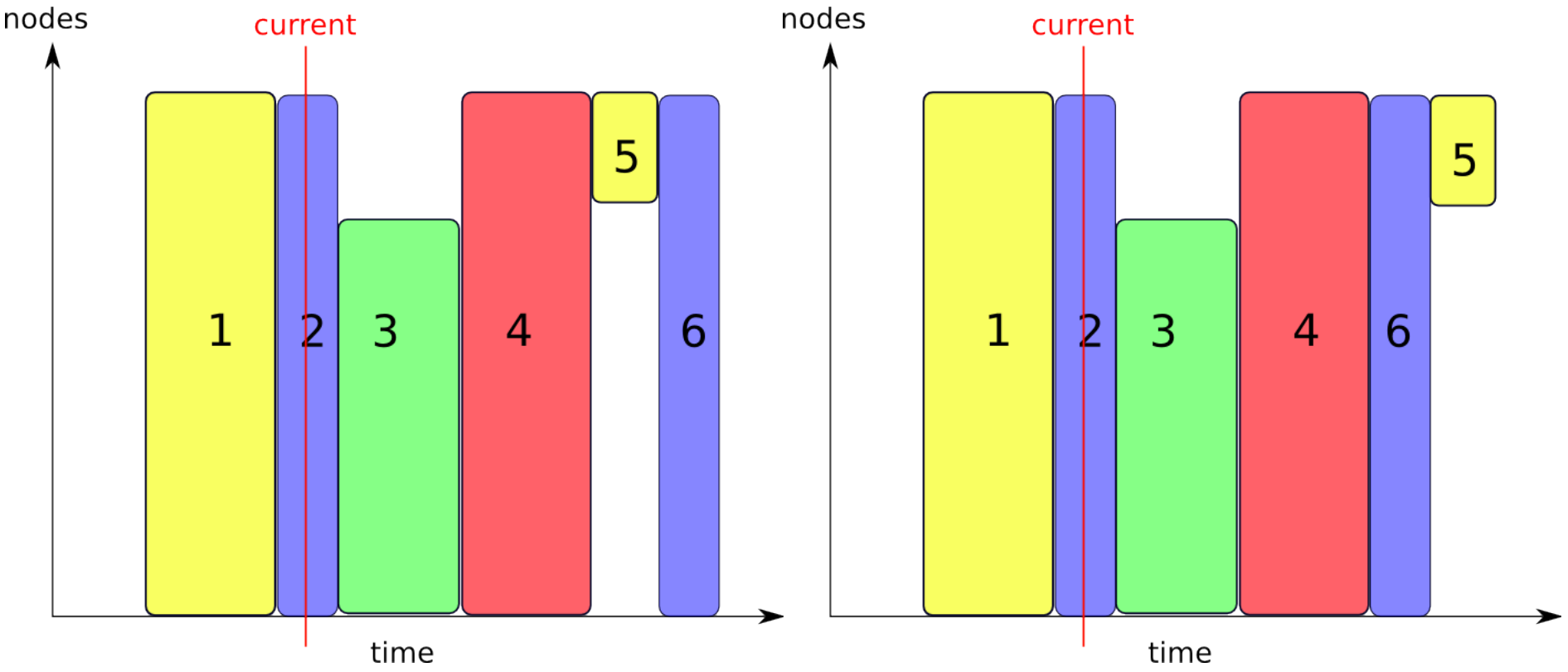
Remplissage des trous si l'ordre des tâches précédentes ne sont pas modifiées





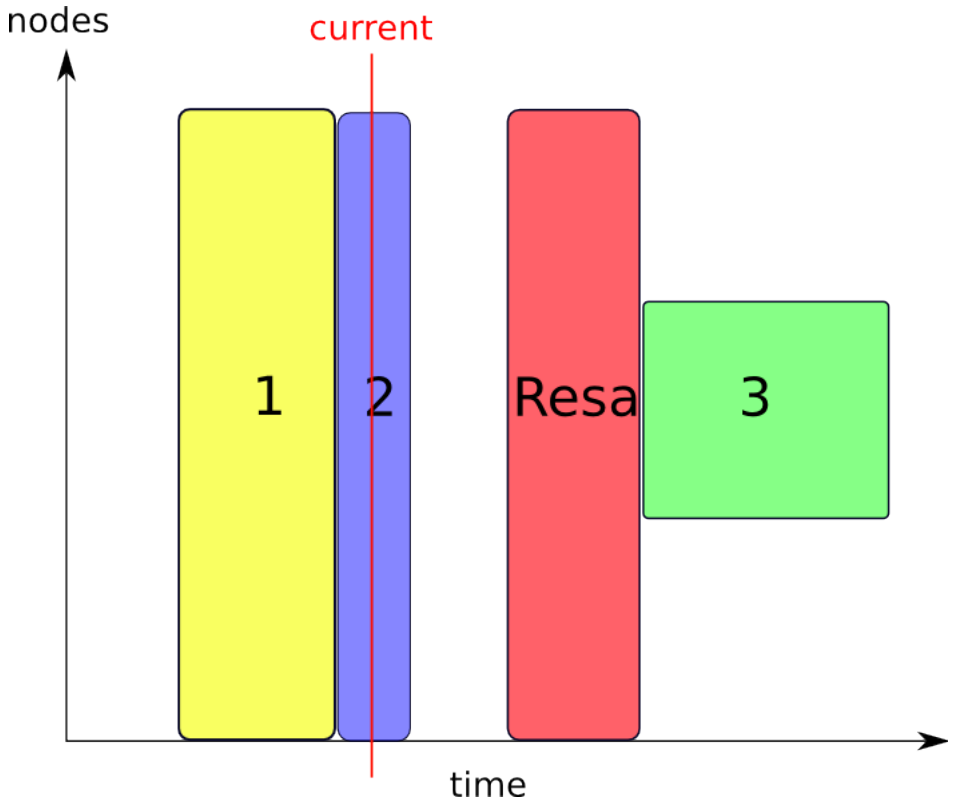
FairSharing (partage équitable)

L'ordre est calculé suivant ce qui a été consommé (on favorise les utilisateurs peu gourmands). Définition d'une fenêtre et paramètres de pondération.



Réservation (*Advance Reservation*)

- **Très pratique** pour démo, planification, tâche de type grille...
- **Mais**
  - Contraignant pour l'ordonnancement (attention au niveau d'utilisation)
  - Les ressources sont rarement utilisée sur toute la durée (gaspillage)

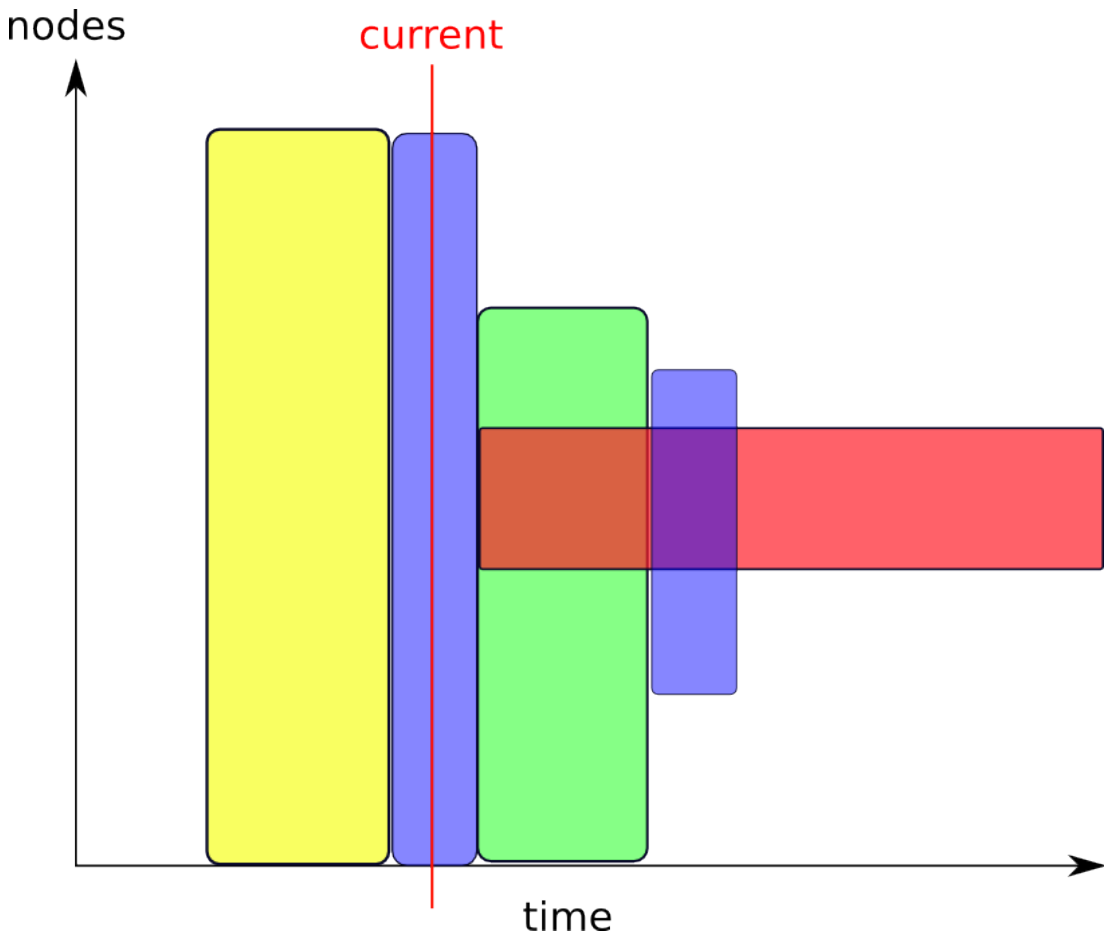


oarsub -r "2008-04-27 11 :00" -l nodes=12

Equilibrage de Charge

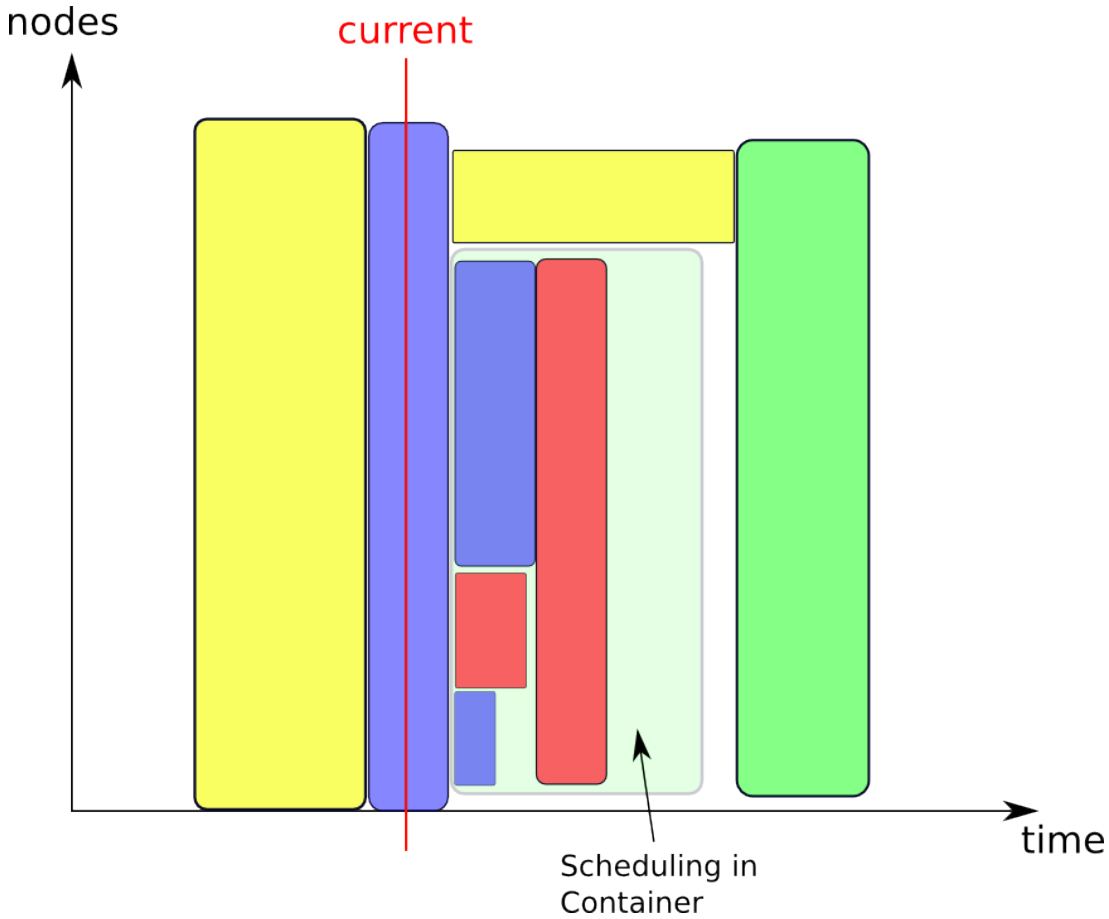
Une solution relativement simple :maintenir des indicateurs de charge et faire un tri en ordre croissant avant affectation. Attention peut interférer ou ne pas être possibles avec certains ordonnanceurs

TimeSharing



Récursivité

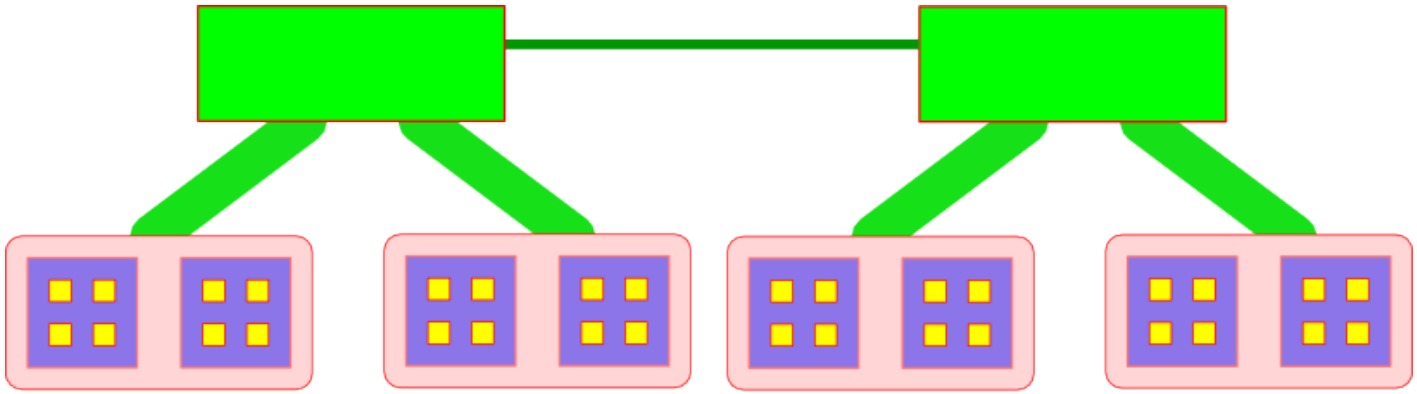
Faire de l'ordonnancement dans une allocation/réservation. Intéressant pour formation, démo, partage de ressource plus flexible par groupe d'utilisateurs / projet. Tâche de type container.



Contraintes Topologiques

Evolution du matériel

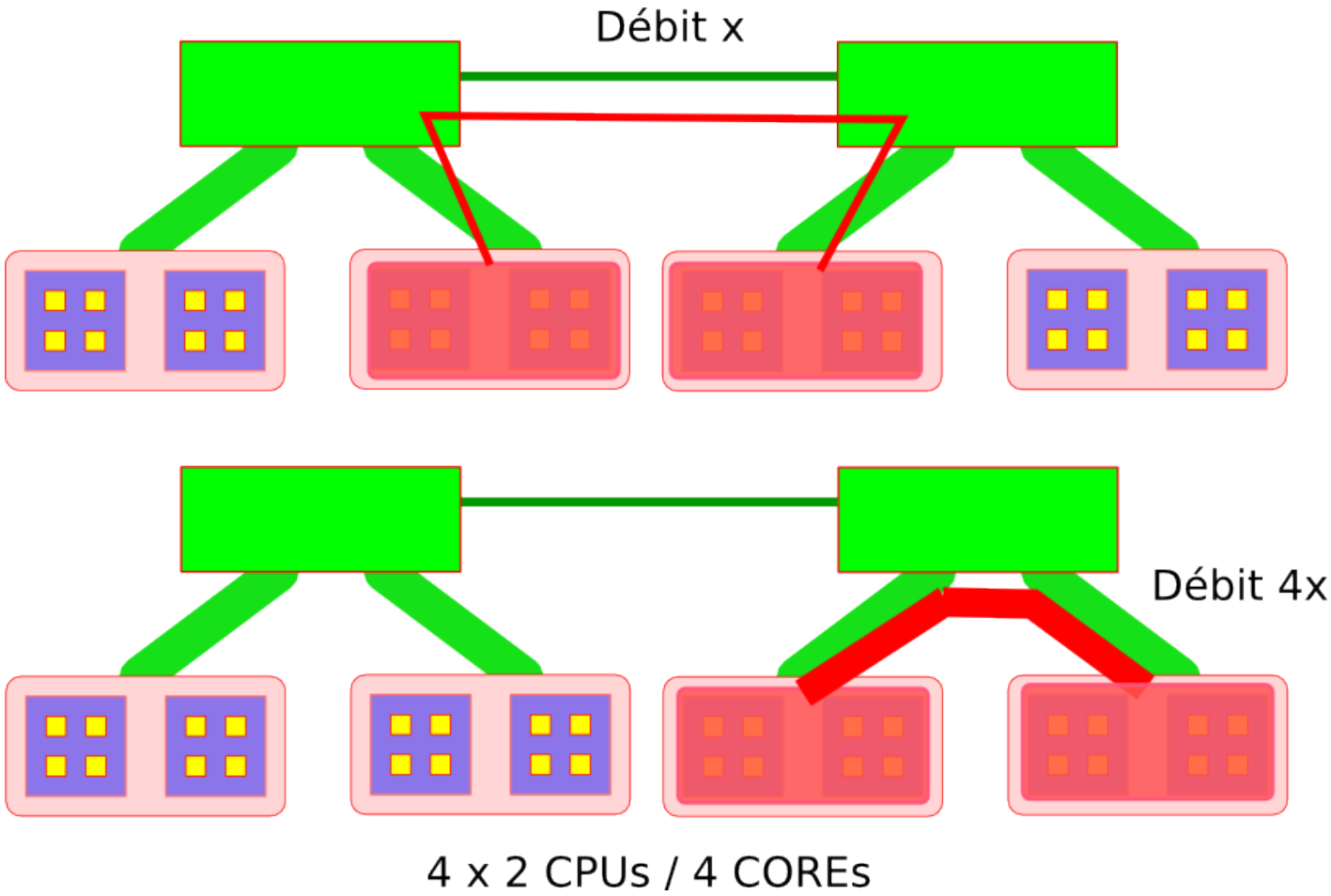
- switch/noeud/cpu/core : Architecture Hierarchique
- machine NUMA / machine BlueGene grille 2D, 3D



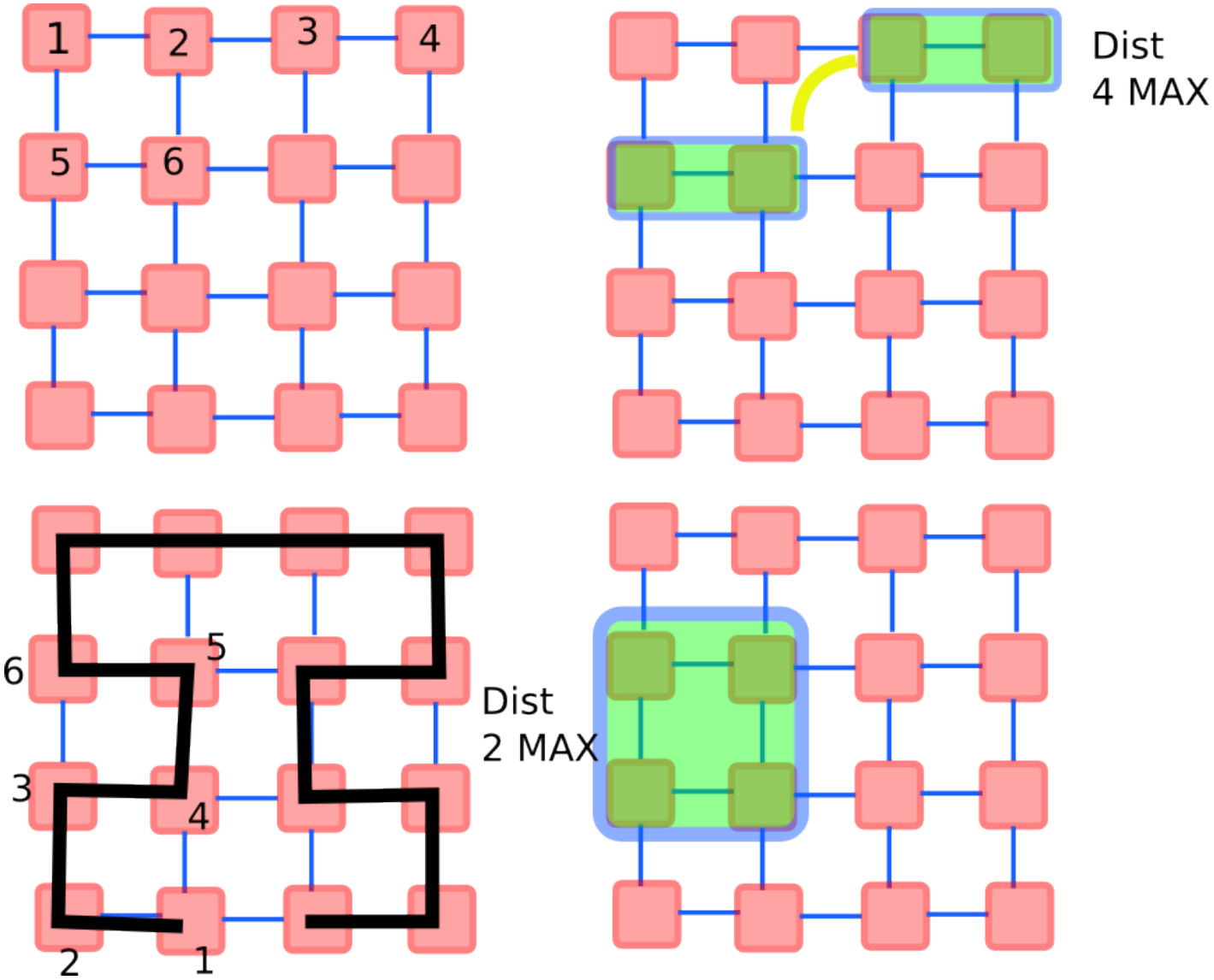
4 x 2 CPUs / 4 COREs

Contraintes Topologiques : hiérarchique

Problème avec les applications parallèles sensible au débit communication.

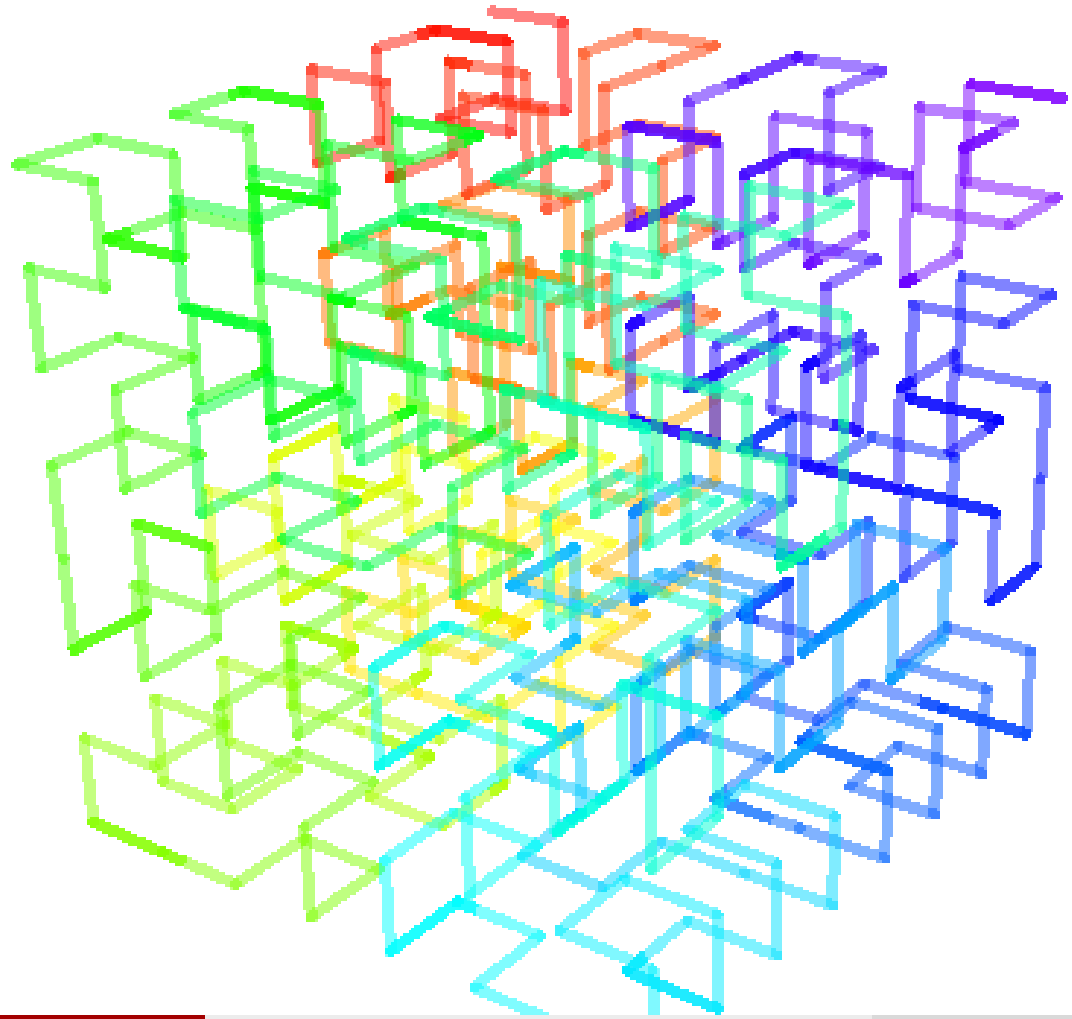


Contraintes Topologiques : grille/tore 2D



Contraintes Topologiques : grille/tore 3D

- Courbe de Hilbert (Slurm / topology)
- Wikipedia / *Hilbert_curve*



Contraintes Topologies :

En résumé

- Les contraintes topologiques complexifient l'ordonnancement, problème d'optimisation
- L'ordonnanceur doit supporter la notion de hiérarchie
- Une bonne numérotation peut faciliter le travail de l'ordonnanceur pour les grilles/tores 2D/3D et **allocation de ressources contiguës**

- **oarsub -l switch=1/nodes=2/cpu=2/core=2 mon-appli-parallèle**
- 1x2x2x2 = 8 coeurs

Application parallèle et affinité processeur

Note : CPUSET ensemble de coeurs et/ou CPU sur un noeud.

1. L'attribution CPUSET/core pour application parallèle peut ne pas suffire
2. Problème de l'ordonnanceur de l'OS (ici souvent Linux), le processus change de coeur à l'intérieur des des CPUSET
3. Il faut utilisé les capacités de vérouillage sur coeur (*Processor Affinity*)

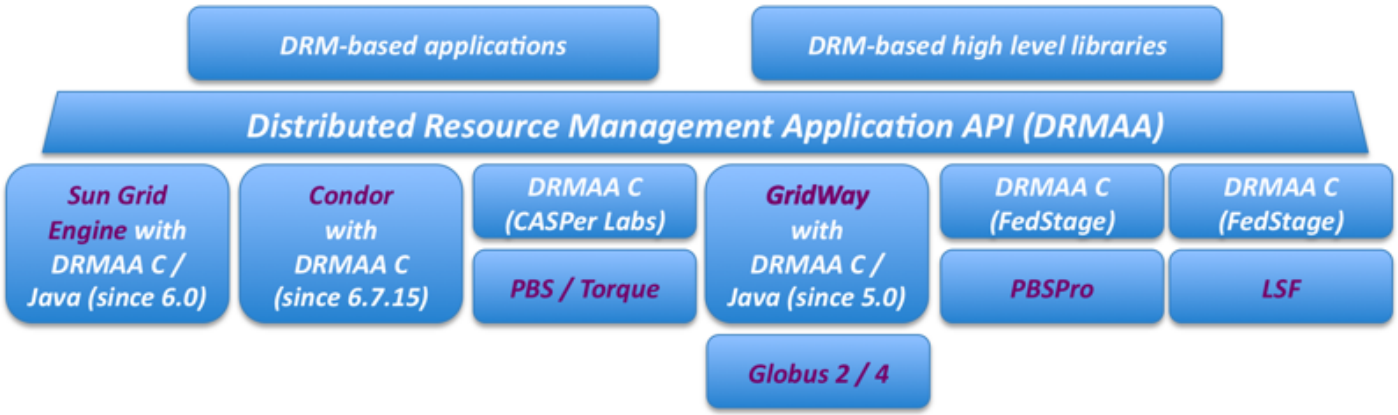
Eco-systéme

Un gestionnaire fait partie d'une infrastructure qui peut être complexe

- Multi-grappe, grille légére, grille type Globus, EGEE
- Outils de déploiement, infrastructure de calcul virtuelle (*Cloud Computing*)
- Outils de monitoring, d'accouting, reporting
- Outils pour la gestion d'érnegie
- Politique de sécurité, outil de confinement réseau
- Partage / couplage de ressource avec un autre gestionnaire de ressources (notion de co-système)

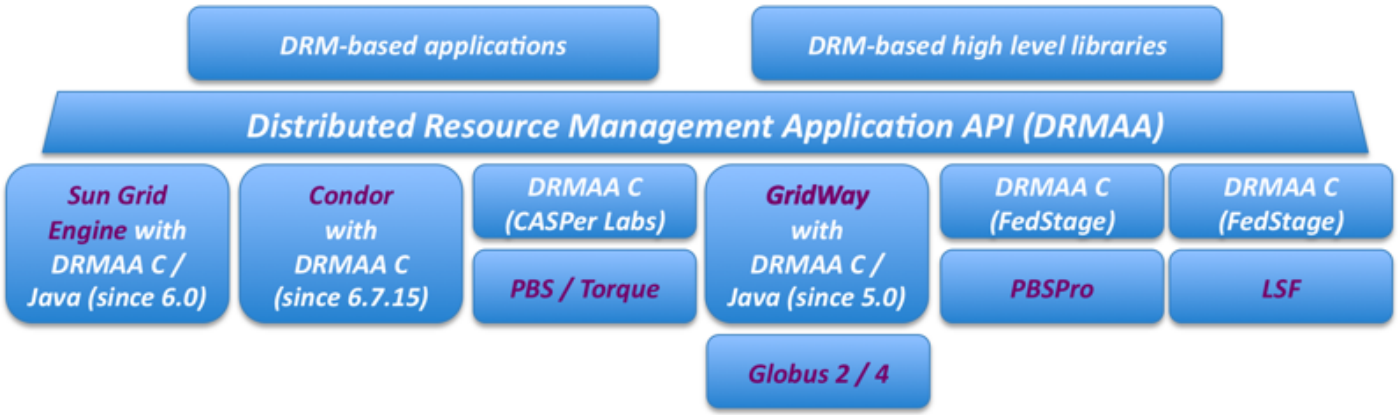
Interfaces

- Interface commande en ligne (CLI)
- Application exemple DRMAA (v1, v2)
- Grille : Globus GT2, GT4/ OGSA-BESS, G-Lite - BLAHp, SAGA
- Interface web REST
- avec des jolies variantes



Interfaces

- Interface commande en ligne (CLI)
- Application exemple DRMAA (v1, v2)
- Grille : Globus GT2, GT4/ OGSA-BESS, G-Lite - BLAHp, SAGA
- Interface web REST
- avec des jolies variantes



Interface web : REST

- REST = protocole HTTP PUT/GET/POST/DELETE sur des ressources
- `http://fr.wikipedia.org/wiki/Representational_State_Transfer`
- interface simplifiée
- présent dans OAR (apparitions dans d'autre gestionnaire LAVA, SGE ? ? ?)

wget -O - `http://mydomain.org/oarapi/resources.json?structure=simple`
Donne la liste de toutes les ressources de la grappe au format json

Energie

The Green500 List

Machines du Top500 triées suivant les Mflops/Watt

File Edit View History Bookmarks Tools Help

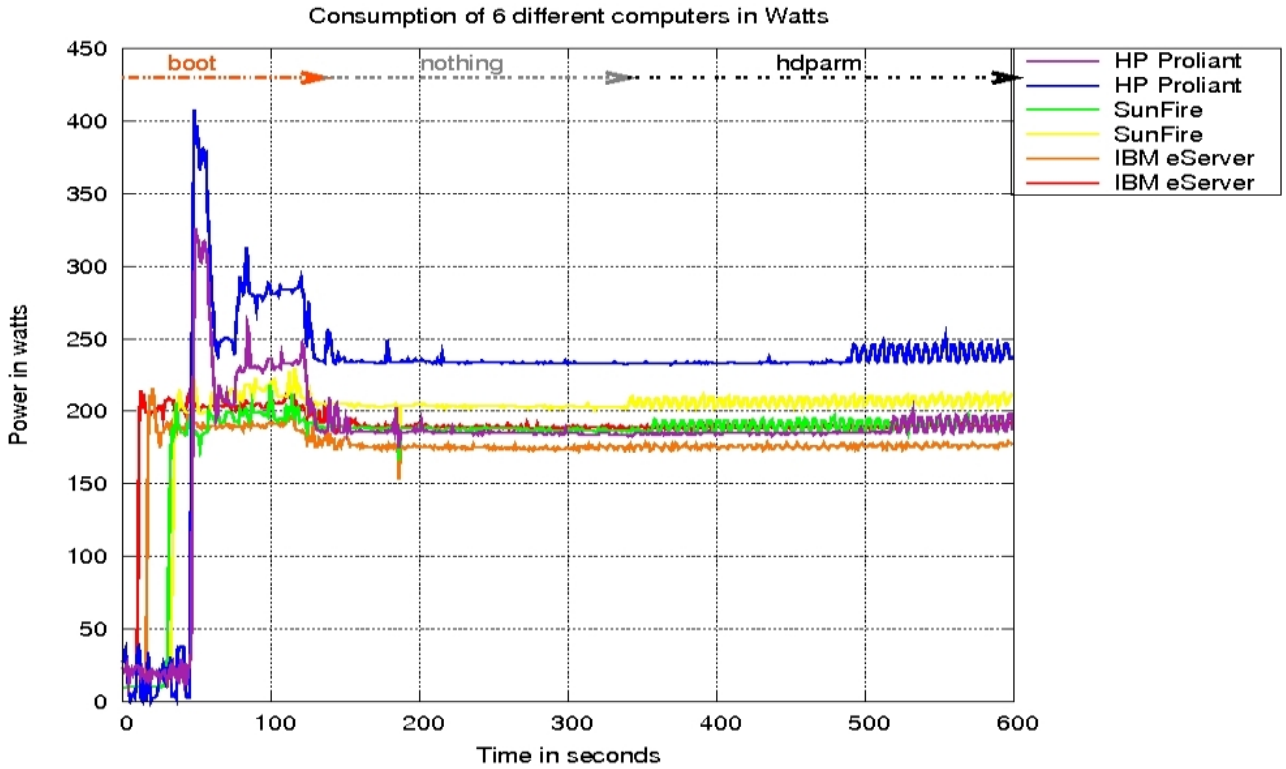
Green500 Rank	MFLOPS/W	Site*	Computer*	Total Power (kW)	TOP500 Rank*
1	536.24	Interdisciplinary Centre for Mathematical and Computational Modelling, University of Warsaw	BladeCenter QS22 Cluster, PowerXCell 8i 4.0 Ghz, Infiniband	34.63	220
2	530.33	Repsol YPF	BladeCenter QS22 Cluster, PowerXCell 8i 3.2 Ghz, Infiniband	26.38	429
2	530.33	Repsol YPF	BladeCenter QS22 Cluster, PowerXCell 8i 3.2 Ghz, Infiniband	26.38	430
2	530.33	Repsol YPF	BladeCenter QS22 Cluster, PowerXCell 8i 3.2 Ghz, Infiniband	26.38	431
5	458.33	DOE/NNSA/LANL	BladeCenter QS22/LS21 Cluster, PowerXCell 8i 3.2 Ghz / Opteron DC 1.8 GHz , Infiniband	138	41
5	458.33	IBM Poughkeepsie Benchmarking Center	BladeCenter QS22/LS21 Cluster, PowerXCell 8i 3.2 Ghz / Opteron DC 1.8 GHz , Infiniband	138	42
7	444.94	DOE/NNSA/LANL	BladeCenter QS22/LS21 Cluster, PowerXCell 8i 3.2 Ghz / Opteron DC 1.8 GHz , Voltaire Infiniband	2483.47	1
8	371.67	ASTRON/University Groningen	Blue Gene/P Solution	94.5	75
9	371.67	IBM - Rochester	Blue Gene/P Solution	126	56
9	371.67	RZG/Max-Planck-Gesellschaft MPI/IPP	Blue Gene/P Solution	126	57

The Green500 List

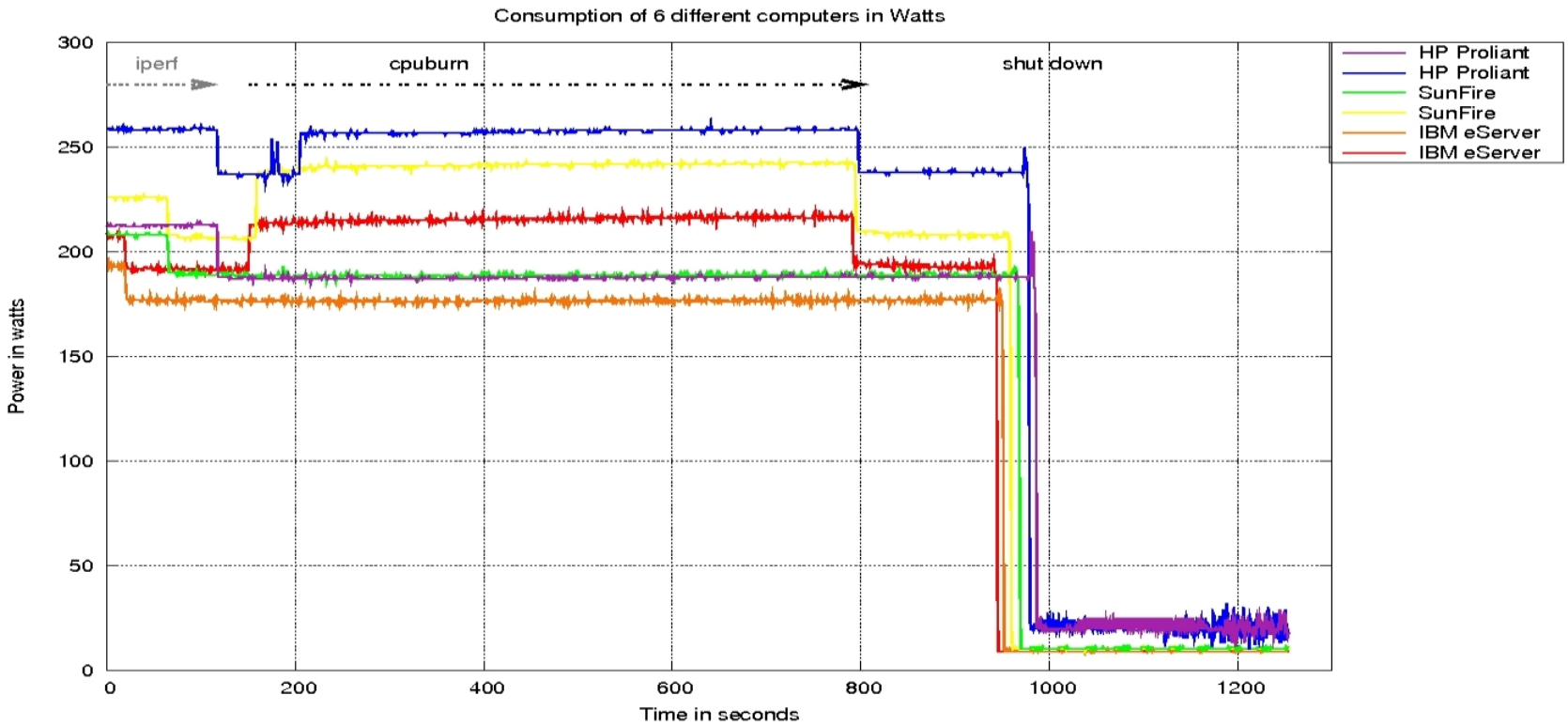
- Les architectures spécialisées occupent les 19 premières places.
- Machine *classique* : Blade Center Xeon QC 2.5 Ghz (**265.80 MFlops/Watt**).
- Le benchmark utilisé (Linpack) est bien connu et bien maîtrisé !
- Pas de données pour des benchmarks plus variés.
- Les informations sur la puissance consommée font leur apparition dans le Top500.

Quelques puissances consommées

Green-Net Projet INRIA sur le suivi de la consommation et l'étude des logiciels pour sa maîtrise dans le HPC.

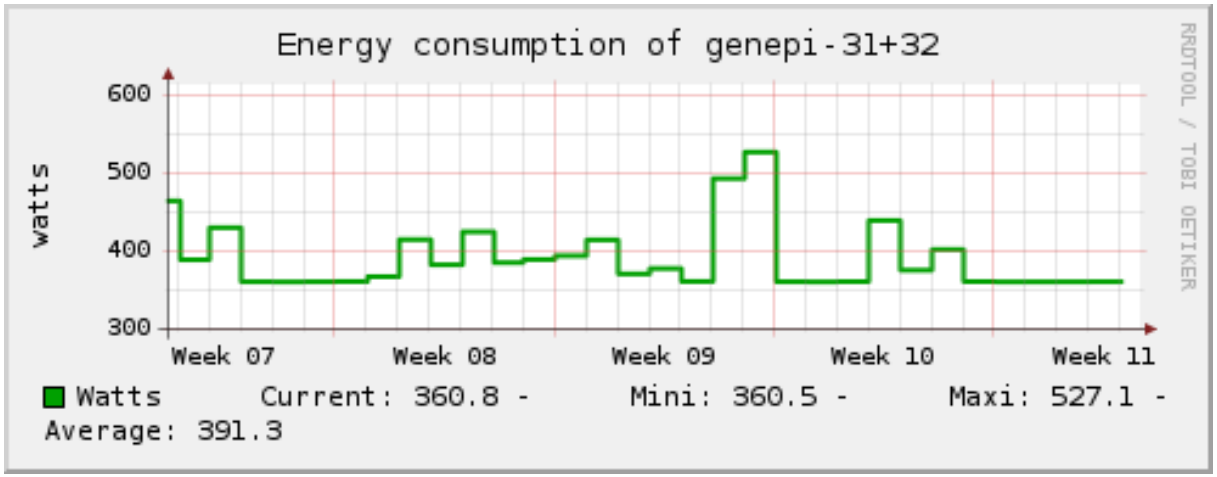


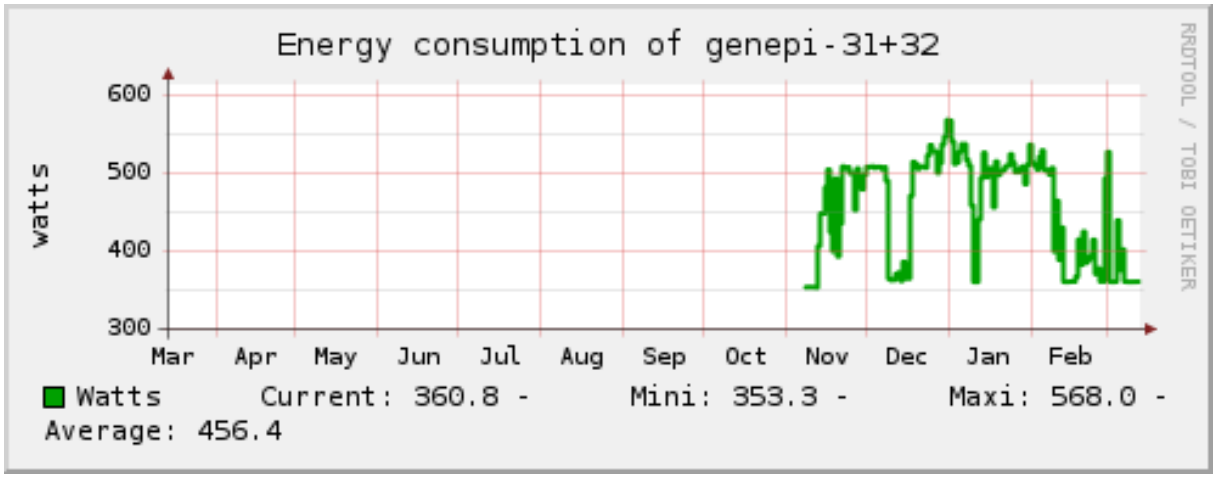
Quelques consommations



Autres consommations

2 machines bi-quad-core Xeon (BULL)





SGI Molecule : Concept Computer

Présentée à SC'08.

- Intel Atom N330
- Rack 3U, 90 noeuds / rack , 5-10 Watts / noeud

- Autre société, Sicortex : 5,832 cpu (64bits MIPS 1,4 Gflops), 20KWatt
- ARM processeur dual-core Cortex A-9 / 2Ghz / 0.5 Watt (FPU ? ? ?)

Centre de calcul, mésocentre, grappes labo, grappes pour l'expérimentation

- Des rôles très variés
  - Régles d'usages, durée des jobs, type de jobs...
- Des taux d'utilisation différents / consommations énergiques
  - 90% – 100% pour les centres de calcul (?).
  - Plus variable pour les méso-centres.
  - Très irrégulier pour les grappes de labo et les plate-formes pour l'expérimentation comme Grid'5000 (25% – 50%).
  - Utilisation des ressources inutilisées pour les applications paramétriques (généralement en mode **BestEffort**), mais il reste de large périodes d'inactivité.

Centre de calcul et Energie

- Maximiser le rendement énergetique (pas forcément la priorité).
- Le matériel est-il bien adapté, efficace... ?
- Quel est le rendement des applications (accélération, gaspillage) ? (rarement connu, ou peu surveillé)
- La gestion globale des ressources permet-elle une bonne maîtrise de la consommation d'énergie ? (les détails de la consommation ne sont que rarement connus)

Quelques études de cas liées à la consommation d'énergie.

Seuil de température

- Cas d'une climatisation limite.
- Lors d'un pic de température nécessité d'arrêter ou de mettre en veille des noeuds.
- La sonde de température alerte le gestionnaire de ressource (puis IPMI ou script de mise en veille).
- Arrêt de noeud libre, noeud avec job besteffort, checkpoint avant retrait du job et arrêt du noeud ou arrêt du noeud et perte du job.
- *Simple à mettre en place dans un gestionnaire de ressource.*

Cluster Virtuel - ComputeMode

- Création d'un cluster virtuel avec les ressources inutilisées
- Exemple salle de TP la nuit (UFRIMA - Université Joseph Fourier)
  - PXE
  - Wake-On-Lan
  - Diskless systems
  - OAR comme gestionnaire de ressources, **réveil à la demande, zone indisponible**
- **Usage : cluster d'appoint intégré dans la grille du Méso-centre CIMENT**
- **Heure creuse, pas de climatisation, disques inutilisés ! :)**

DSLlab

- Plateforme pour l'expérimentation sur Internet/ réseau ADSL.
- Machine fanless chez les particuliers.
- Les machines sont en veille lorsqu'elles sont inutilisées (pas de Wake-On-Lan possible)
- Fonction d'**heure de réveil** par les carte-mères (géré via par le gestionnaire de ressource)

Arrêt / Mise Veille / Réveil

- Arrêt / Mise Veille des noeuds lorsqu'ils sont inutilisés
- Réveil lors de l'arrivée de nouveau job
- Limiter les cycle d'arrêts/réveil (réactivité) $\longrightarrow$ prédire la charge.
- **Note :** Arrêt/allumage de machines fatiguent peu le matériel (15000 cycles arrêt brutal/allumage pas de souci particulier).
- *Assez simple à mettre en place dans un gestionnaire de ressource.*

Tarifications heures pleines/creuses - Tâche Priotaire

- Les tâches prioritaires passent en journée en heures pleines.
- Toutes les tâches peuvent passer la nuit en heures creuses.
- Variantes : des noeuds sont éteints en journée ou bloqués à vitesse réduite (consommation limitée, **attention, par forcément le plus efficace en énergie consommée, durée/efficacité**)
- *Assez simple à mettre en place dans un gestionnaire de ressource.*

Slurm

Approche simple

- **SuspendTime :** nombre de seconde à partir duquel un noeud peut être mise en veille / éteint
- **SuspendRate, ResumeRate :** nombre de noeud par minute pouvant changer d'état (important pour les grosses installation)
- **SuspendProgram, ResumeProgram :** programme à exécuter pour contrôler les noeuds
- **SuspendExcNodes, SuspendExcParts :** noeuds et/ou partition à exclure du contrôle

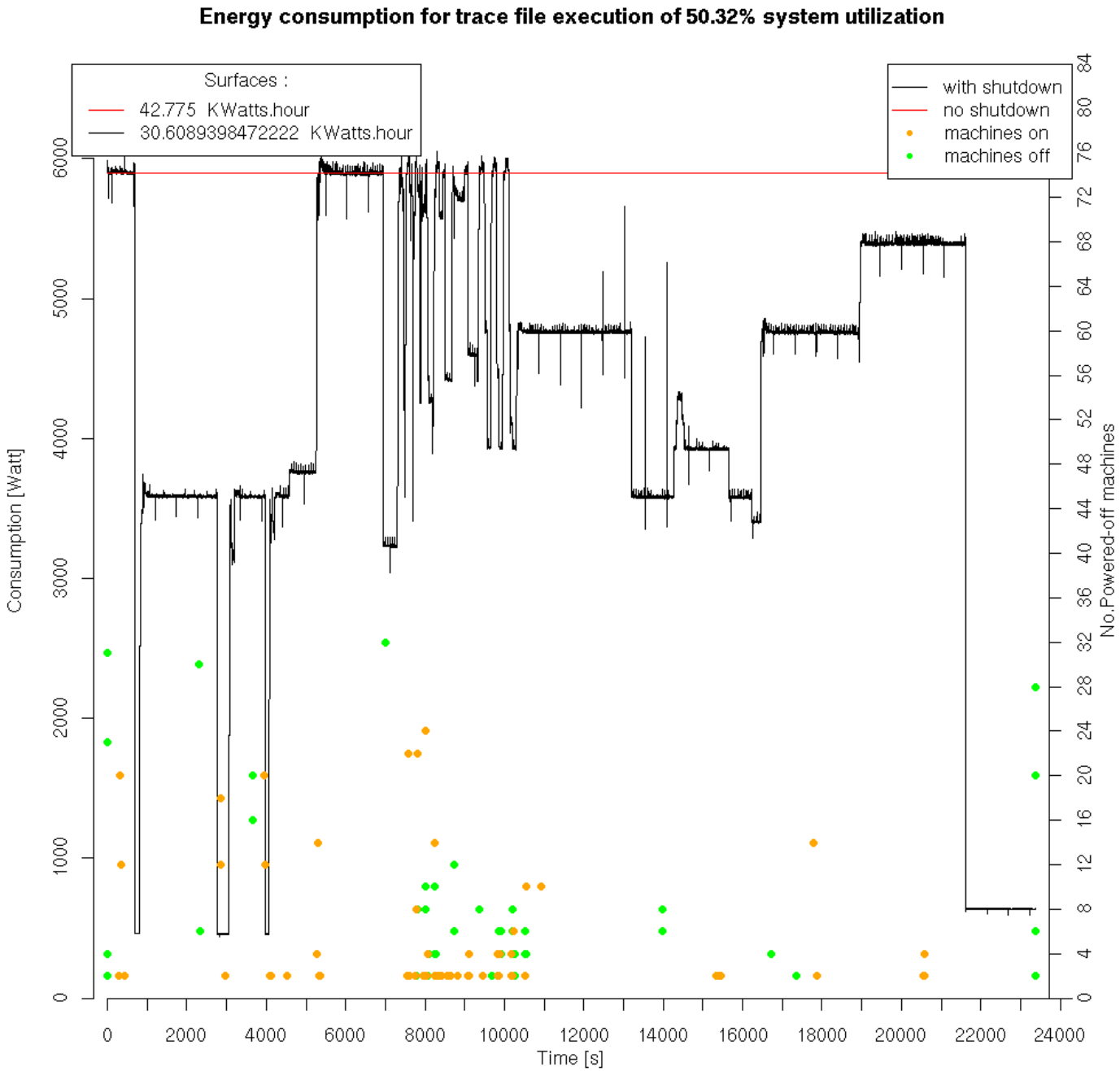
LSF, Moab (Cluster Resources)

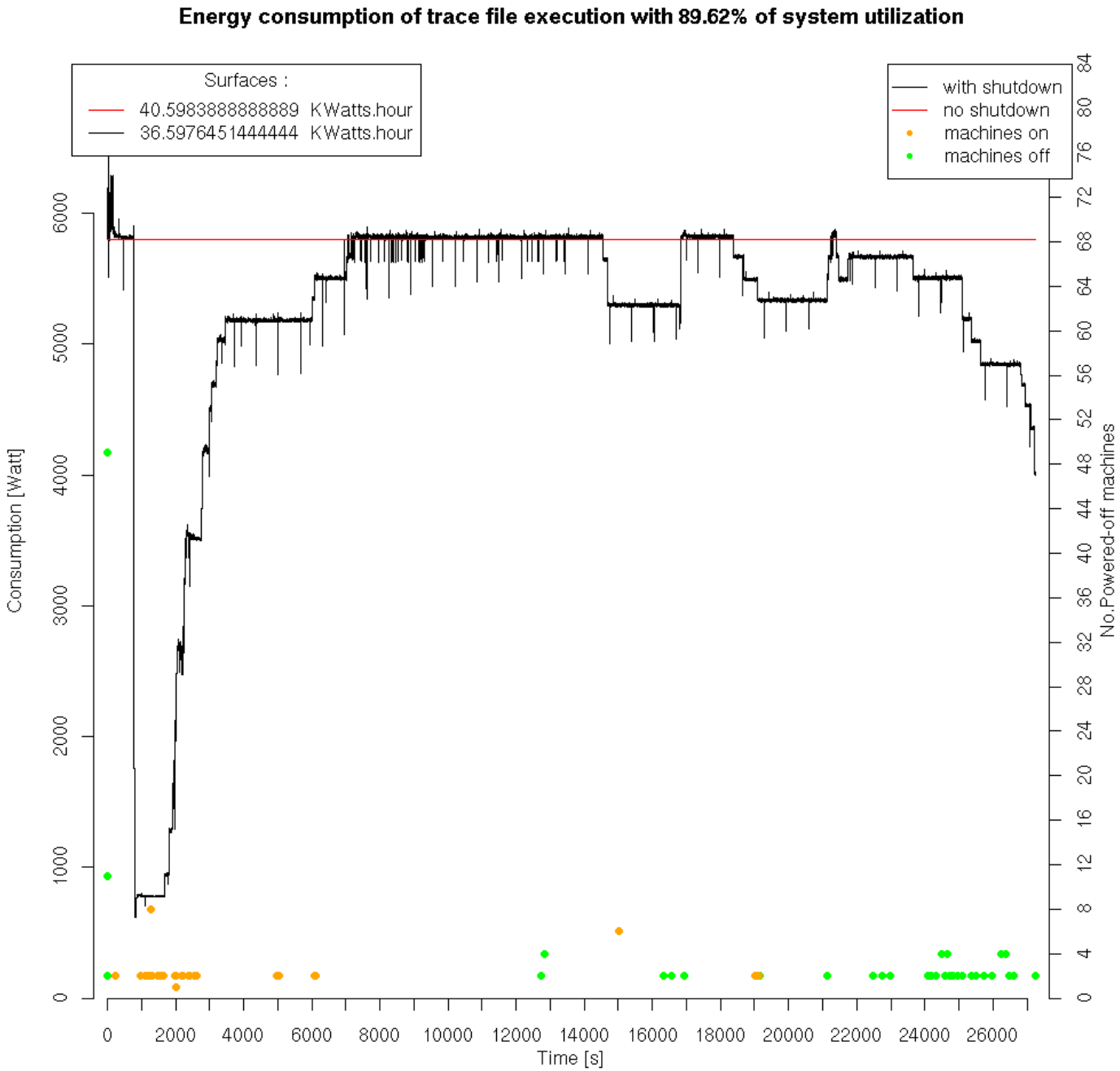
Attention pas testé, document très commercial pour Moab, factuel pour LSF

- Suivi de consommation, température
- Usage de consommation par utilisateur, projet, job (?)
- Gestion/contrôle d'énergie
  - arrêt/mise en veille de noeud
  - priorité heures creuses/ heures pleines

OAR et gestion de l'énergie

- Priorité heures creuses/pleines par paramétrage
- Développées lors du *Google Summer Of Code 2008* (Gsoc'08)
  - Module de prédiction de charge.
  - Un nouveau type de job paramétrique : **powersaving** + options (cpufreq, arrêt sélectif de périphérique disque, video ..., politique spécifique)
  - Ex Job BestEffort $\longrightarrow$ fréquence CPU la plus faible.


Energy consumption for trace file execution of 50.32% system utilization
Surfaces :
42.775 KWatts.hour
30.6089398472222 KWatts.hour
with shutdown
no shutdown
machines on
machines off
Consumption [Watt]
No.Powered-off machines
Time [s]
0
1000
2000
3000
4000
5000
6000
0
2000
4000
6000
8000
10000
12000
14000
16000
18000
20000
22000
24000
0
4
8
12
16
20
24
28
32
36
40
44
48
52
56
60
64
68
72
76
80
84


Energy consumption of trace file execution with 89.62% of system utilization
Surfaces :
40.5983888888889 KWatts.hour
36.5976451444444 KWatts.hour
with shutdown
no shutdown
machines on
machines off
Consumption [Watt]
No.Powered-off machines
Time [s]
0
1000
2000
3000
4000
5000
6000
0
4
8
12
16
20
24
28
32
36
40
44
48
52
56
60
64
68
72
76
80
84
0
2000
4000
6000
8000
10000
12000
14000
16000
18000
20000
22000
24000
26000

Du coté des applications et du système

Des travaux de recherches ;

- Contention mémoire, concurrence et consommation.
- Application MPI et contention (10% conso en moins, 1% de temps en plus).
- DVFS et opérations I/O.
- Consommation et machines virtuelles (vision intégré).
- Répartition de charge au niveau des grilles.

En pratique

- La sélection du matériel, monitoring précis de la consommation.
- Bien connaître les applications (bon rendement énergétique).
- Discussion avec les utilisateurs (pour la maîtrise du gaspillage, qualité du code)
- Politique *de gestion d'énergie* : arrêt/mise en veille, priorité, heures pleines/ heures creuses,
- Veille technologique...

Divers

Divers cas d'exploitation

- Applications Multiparamétriques
  - Utilisation des ressources non-utilisées
- Déploiement/Virtualisation
  - Des ressources plus simples à exploiter pour les utilisateurs
- Ressources hétérogénes
  - mémoire
  - réseaux
  - licence
- Tolérance aux pannes
- Haute-disponibilité
- Multi-grappes

Haute-disponibilité

Assurer la continuité de service est important pour les grandes infrastructure

Pannes d'un noeud de calcul :

- Arrêt en erreur de la tâche (nettoyage des autres noeuds)
  - re-soumission automatique (si option positionnée)
  - reprise depuis un point de reprise si disponible (*checkpointing*)

Pannes du seveur :

1. maintient d'un second serveur (synchronisation d'état), bascule auto
2. élection d'un nouveau serveur parmi les noeuds de calcul (LSF)

Note : Suppose la HA sur les autres services critiques comme l'authentification (ex Ldap), le système de fichier distribué (ex NFS) (exemple SGE), de nommage (ex DNS), BD (OAR)...

Multi-grappe

Le cas des multi-grappes est très courant :

1. achat d'une nouvelle grappe et conservation de l'ancienne
2. achat par tranche

Deux approches distinctes :

1. un gestionnaire par grappe
   - file de routage vers les autres gestionnaire de tâches/ressources
2. un seul gestionnaire pour l'ensemble des grappes [3]
   - chaque grappe est vue comme une partition homogéne dans l'ensemble des ressources
   - suppose (*pousse pour*) que les services soient commun à chaque grappe (ex : système de fichier, authentification,...)
   - simplifie énormément l'administration

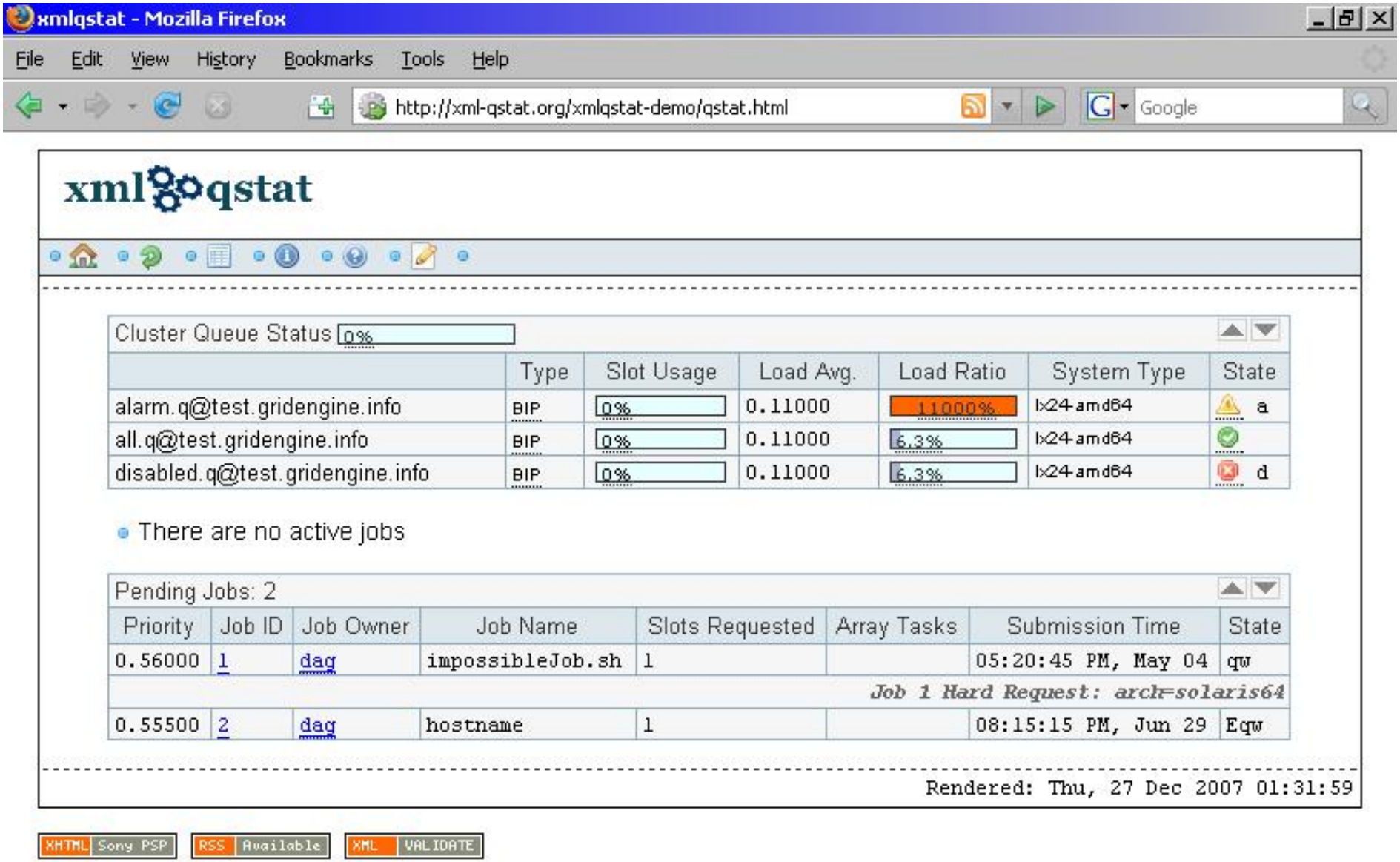
[3] C'est le cas pour Grid'5000, 3 à 5 grappes par site

Cas des longues tâches

1. Dédier des noeuds
2. Suspendre en journée / relancer la nuit ou le week-end
3. Checkpoint (point de reprise)
   - applicatif (la solution la plus sûre)
   - système (contraintes, limitations)

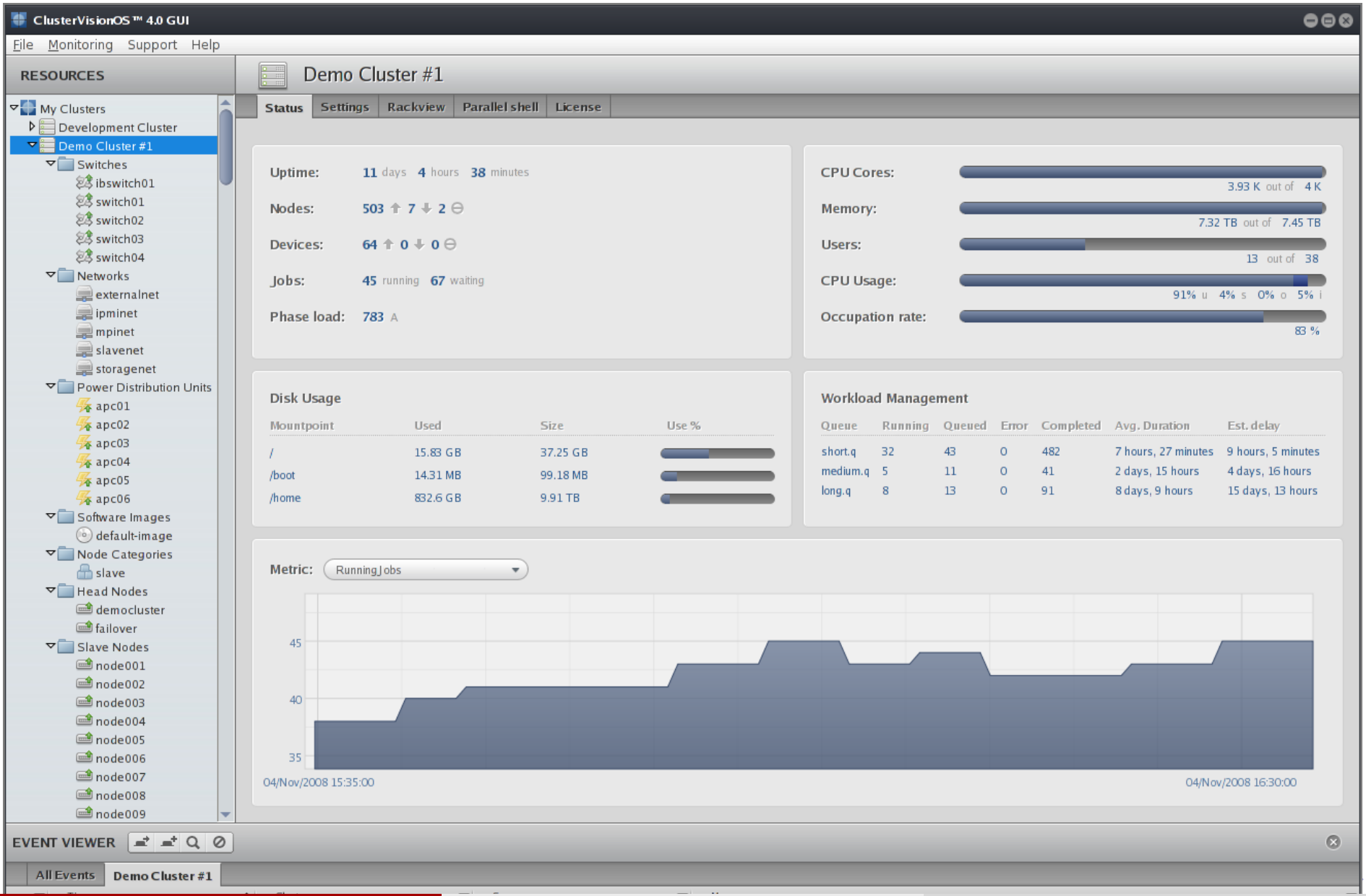
GUI

SGE : Xml-qstat



OAR : Diagramme de Gantt

ClusterVisionOS : une vision intégrée



Eléments de comparaison (Forcément biasé ! ! !)

- **Condor** référence académique (High-Throughput Computing)
- **Sun Grid Engine (SGE)** vieillissant / vraiment libre ?
- **MAUI/Torque** vieillissant / vraiment libre ?
- **Slurm très grandes machines**
- **OAR Challenger** :)
- LSF (Platform) (pour le support)
- PBS Pro (pour le support)

C'est aussi une affaire de goût ?

- Différence dans la philosophie : exemple OAR définit ressources exemple les cores, les licences, SGE dédinit des queue, des hosts auxquels sont rattachés des ressources

Conclusion

Ce qu'il faut retenir :

- Les grappes sont quasi-ominiprésentes dans le domaine des sciences appliquées.
- Leur taille augmente
- Les gestionnaires de tâches et de ressources sont nécessaires
- **Fixer une politique de partage et d'accès**
- **Dialoguer/Former/Informer les utilisateurs (réunion d'information,documentation, chartre, tutoriaux...)**
- Des gestionnaires de ressources pour tout les goûts (logiciels libres et propriétaires)
- Le réglage fin reste **complexe** (les infrastructures sont complexes, et les demandes aussi). Beacoup de compromis.

Des questions ?

Liens


Condor
http ://www.cs.wisc.edu/condor/


Sun Grid Engine (SGE)
http ://gridengine.sunsource.net


TORQUE/MAUI
http ://www.clusterresources.com/


SLURM
www.llnl.gov/linux/slurm/


LSF
http ://www.platform.com


OAR
http ://oar.imag.fr