OAR

Un gestionnaire de tâches et de ressource pour grandes grappes de calcul

Olivier Richard Bruno Bzeznik Romain Cavagna Joseph Emeras

LIG / CIMENT / ALADDIN-G5K

JRES - 3 decembre 2009

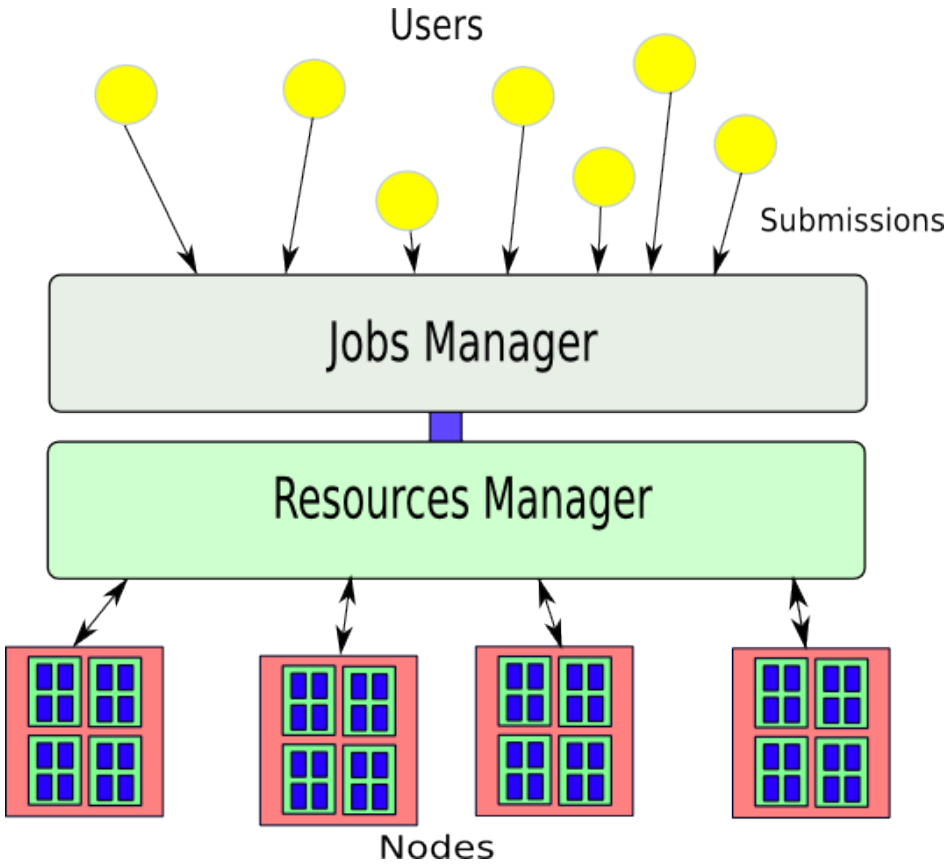
Sommaire

Les Gestionnaires de tâches et de ressources

- Aussi appelés *Batch Scheduler*
- Contexte : **High Performance Computing** (Supercalculateurs, Grappes de calcul, Clusters)
- Existent en très grand-nombre : **Condor** , **Sun Grid Engine (SGE)**, **MAUI/Torque**, **Slurm**, **OAR**, **Catalina**, *LSF* , **Lava**, *PBS Pro*, *Moab*, Loadleveler, CCS...
- `http://en.wikipedia.org/wiki/Job_scheduler`

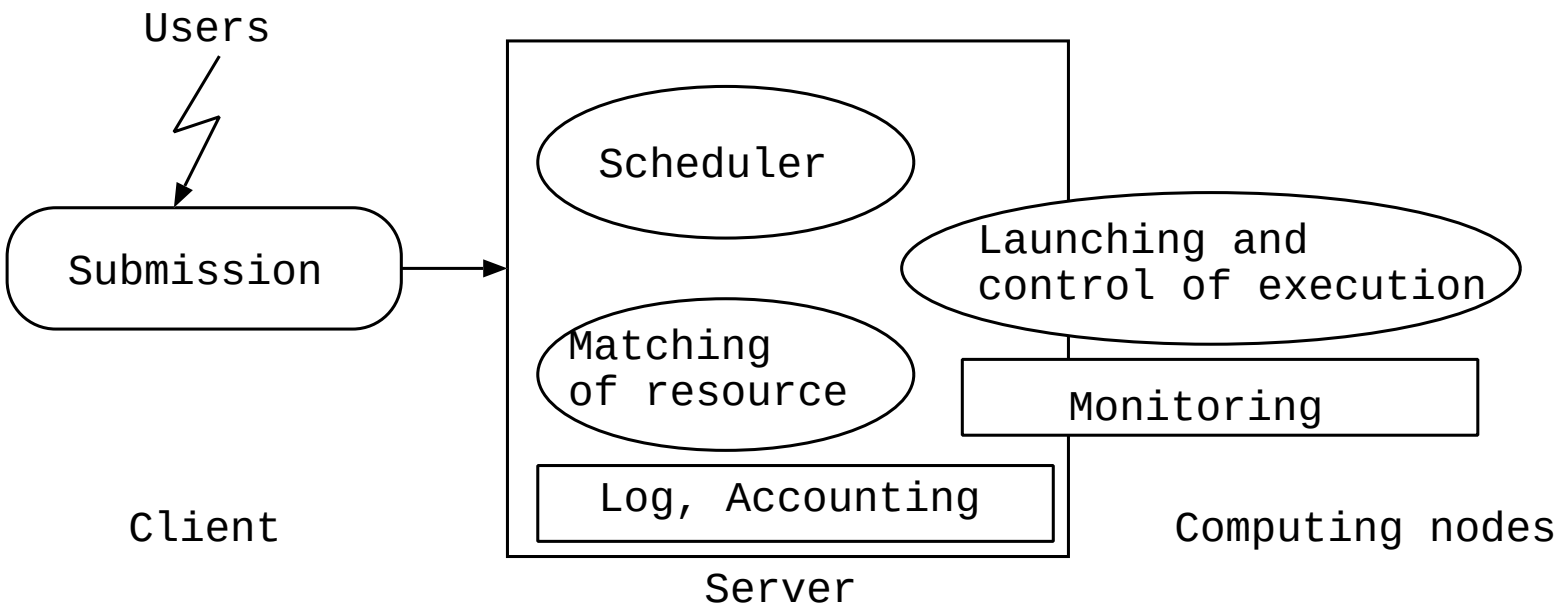
Principe général

Dans leur version simple, les gestionnaires de tâches et de ressources sont séparés 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 exclusif / 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

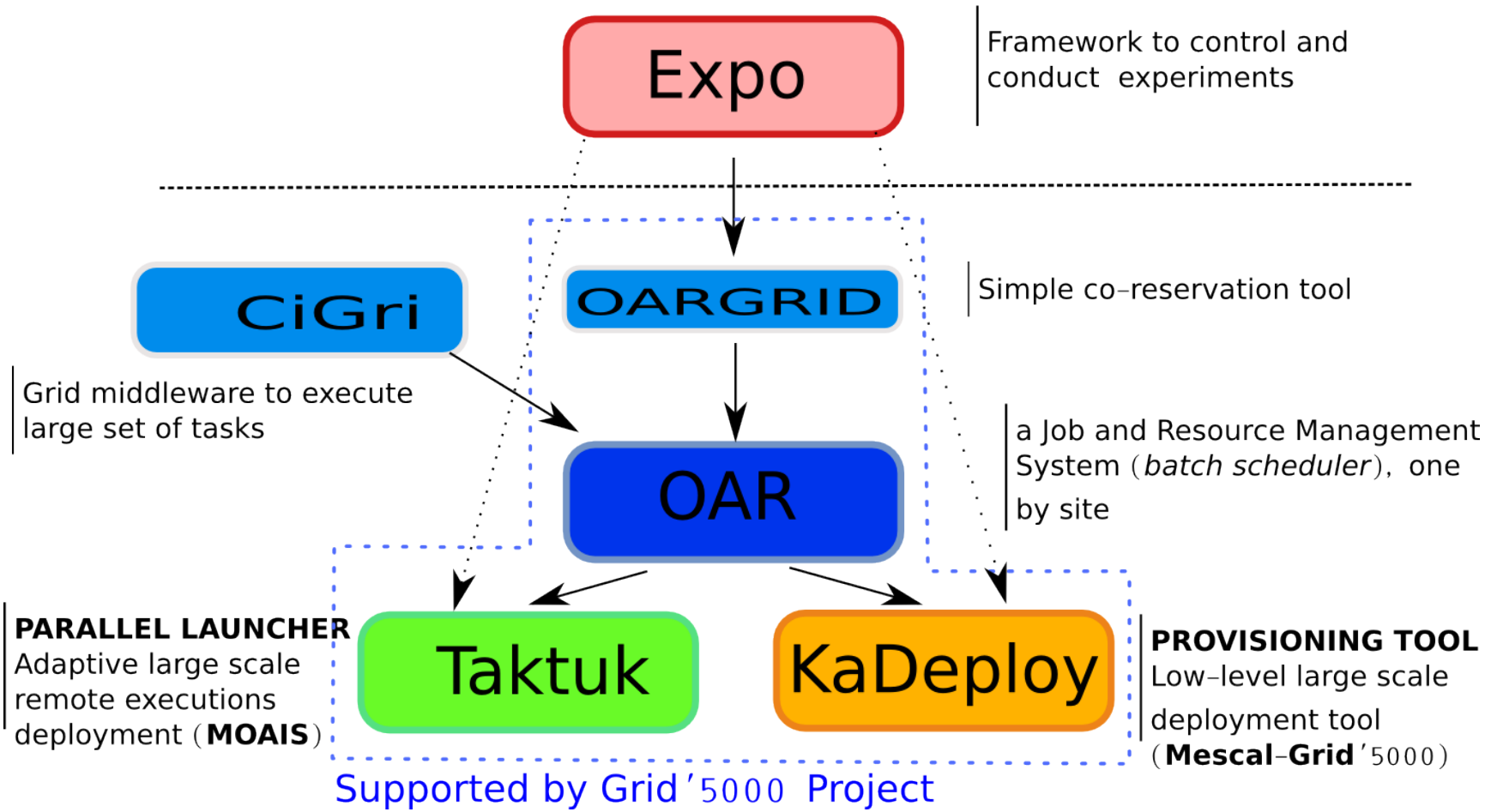
- Tableaux de tâches
- First-Fit (Conservative Backfilling,)
- Fairsharing
- ...

Objectifs

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

- Suivre l'évolution technologique (machine et infrastructure de plus en plus complexe)
- Contexte initial : CIMENT ▸ ... et Grid5000 ▸ ...
- Adaptation aux différents contextes (cluster, cluster-on-demand, cluster virtuel, multi-cluster, grille légère, plate-forme pour l'expérimentation à la Grid'5000, *grand cluster*, besoin spécifique).

Eco-système


Expo
Framework to control and conduct experiments
CiGri
OARGRID
Simple co-reservation tool
Grid middleware to execute large set of tasks
OAR
a Job and Resource Management System (batch scheduler), one by site
PARALLEL LAUNCHER
Adaptive large scale remote executions deployment (MOAIS)
Taktuk
KaDeploy
PROVISIONING TOOL
Low-level large scale deployment tool (Mescal-Grid'5000)
Supported by Grid'5000 Project

Fonctionnalités et spécifités de OAR

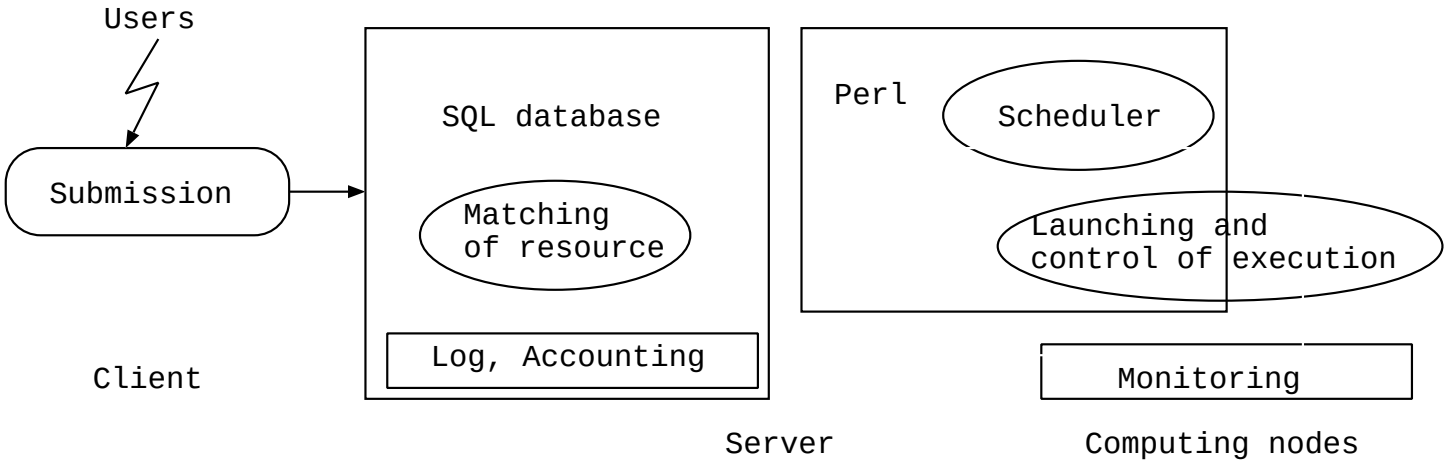
liste non-exhaustive

- Fonctionnalités classiques +
- Advance Reservation
- **Expression des hiérarchies dans les requêtes**
- **Support de ressources de type différent (ex licence, capacité de stockage, capacité réseaux...)**
- **Tâche container** (*récursivité, soumettre dans une tâche*)
- **Tâche besteffort** (tâche à priorité nulle, très utilisé par *CiGri*)
- **Type multiple de tâches** (besteffort, deploy, timesharing, idempotent, power, cosystem ...) (personnalisable)
- **Tâches moldables**
- **Economie d'énergie**

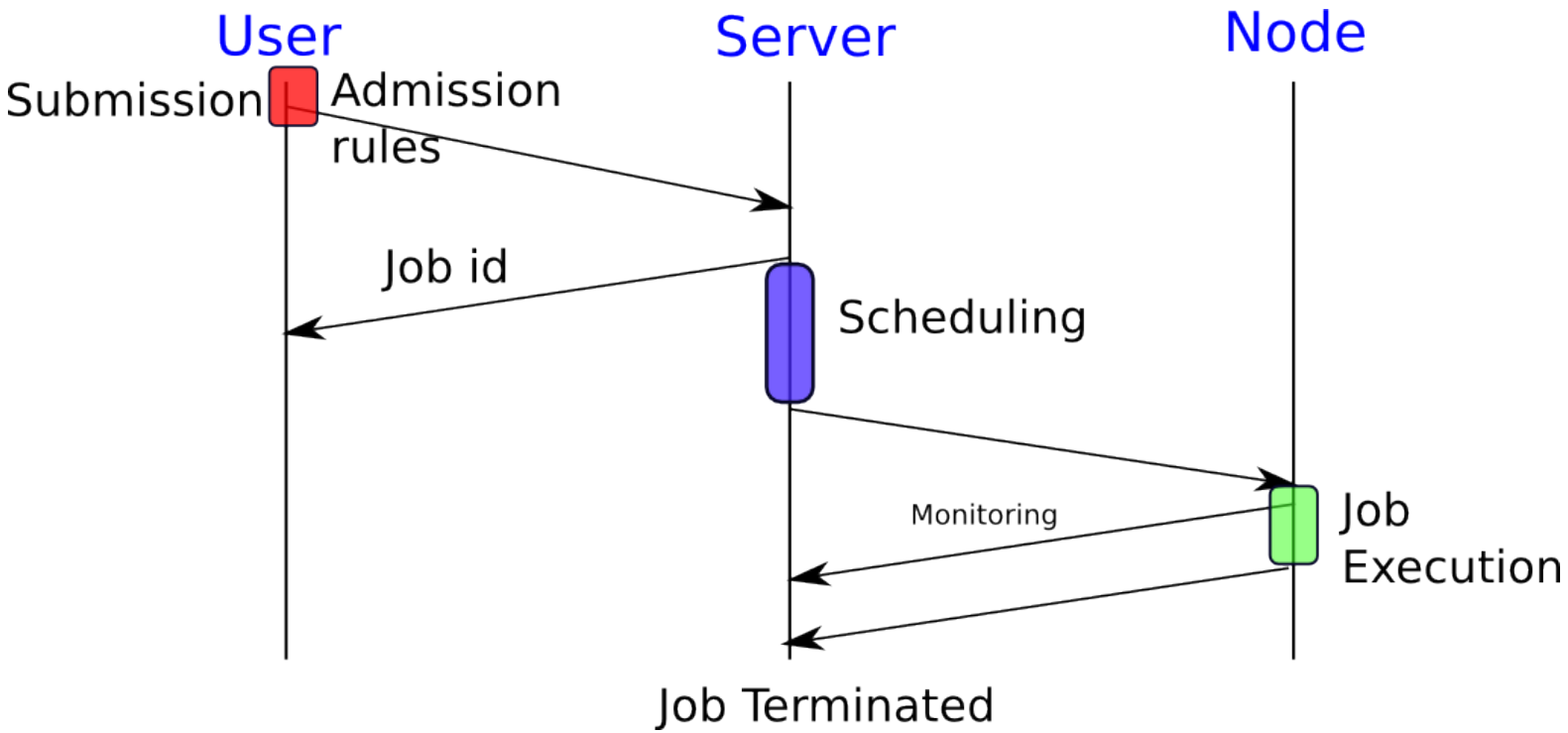
OAR : principes de conception

Utilisation de composants logiciels de haut niveau

- **Base de donnée relationnelle** (MySql/PostgreSQL) au coeur du système pour stocker et échanger
- **Language(s) de script** (Perl, Ruby) pour le moteur d'exécution et les modules (interchangeables)
- **Autres composants** : SSH, CPUSETS, Taktuk



Cycle d'un job

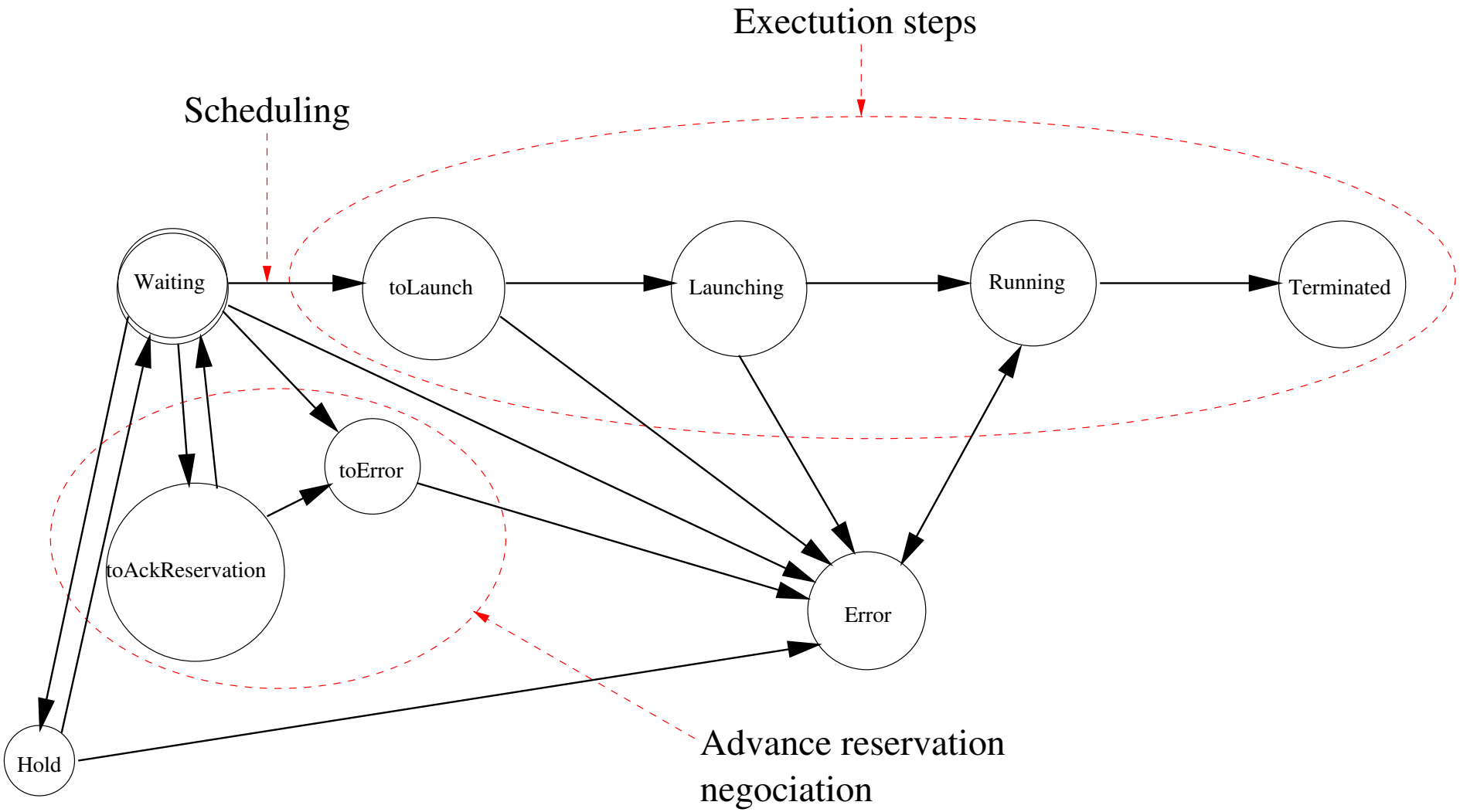


Règles d'admissions

Un point de paramétrage important

- Grandes possibilités de *personnalisation* pour l'administrateur
- **Cadrage** des requêtes
- fixe des valeurs par défaut : walltime, queue, nombre de ressources demandées,
- contrôle d'accès (utilisateur, groupe, plage horaire...)

Diagramme d'état d'une tâche



Exemples 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**

Connexion 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**.

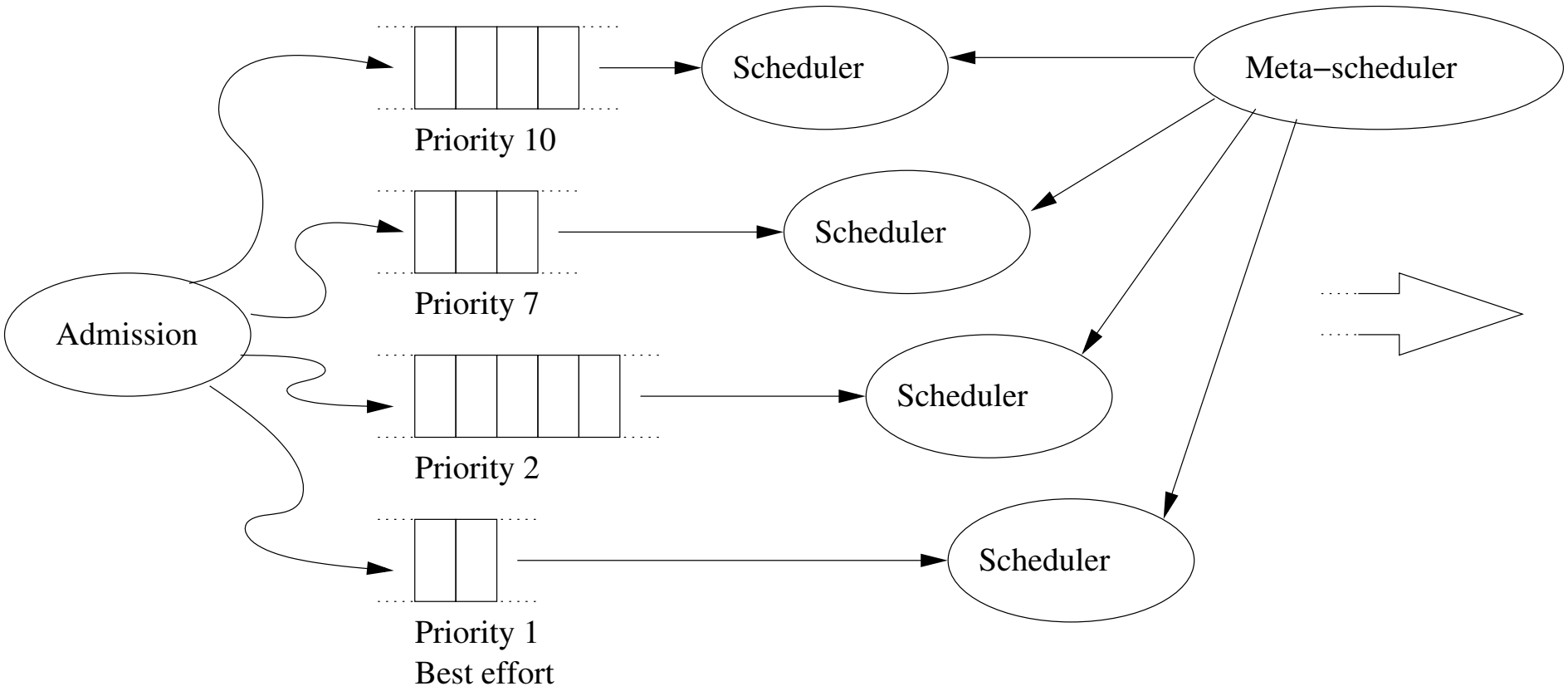
De plus de nombreux **critères et paramètres** sont utilisés pour guider et cadrer les allocations et les priorités.

[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

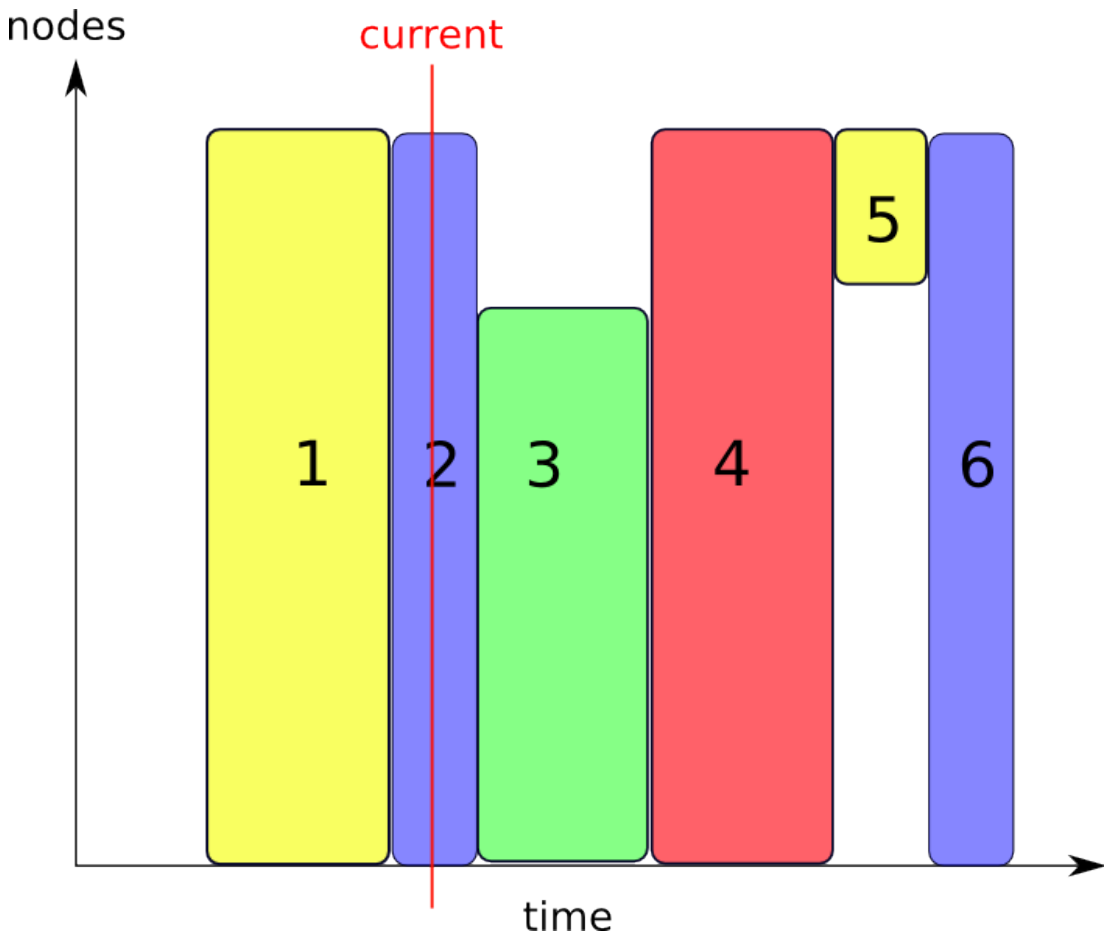
Une étape préliminaire à l'ordonnancement

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

Politiques d'ordonnancement

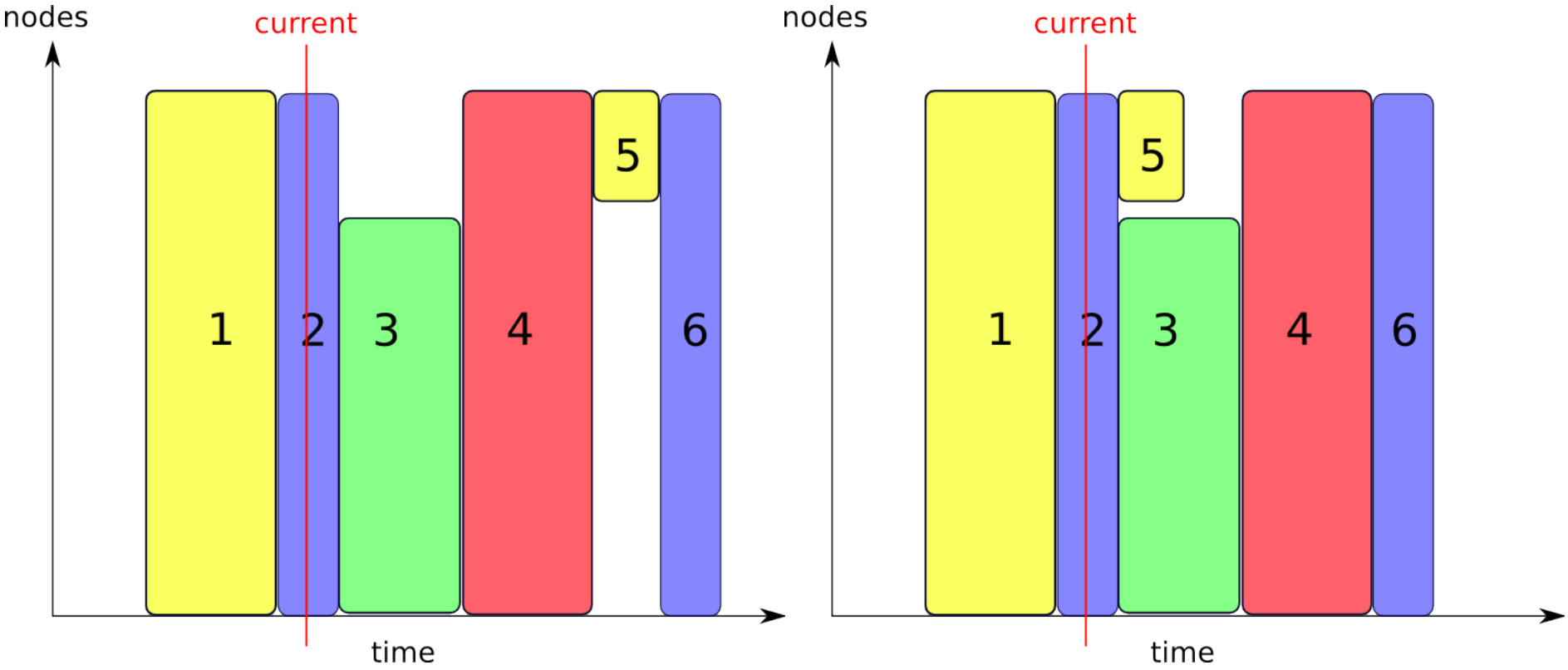
- FIFO (First-In First-Out)
- First-Fit (Backfilling)
- FairSharing
- Timesharing
- Advance reservation
- Récursivité

FIFO : Fisrt-In First-Out



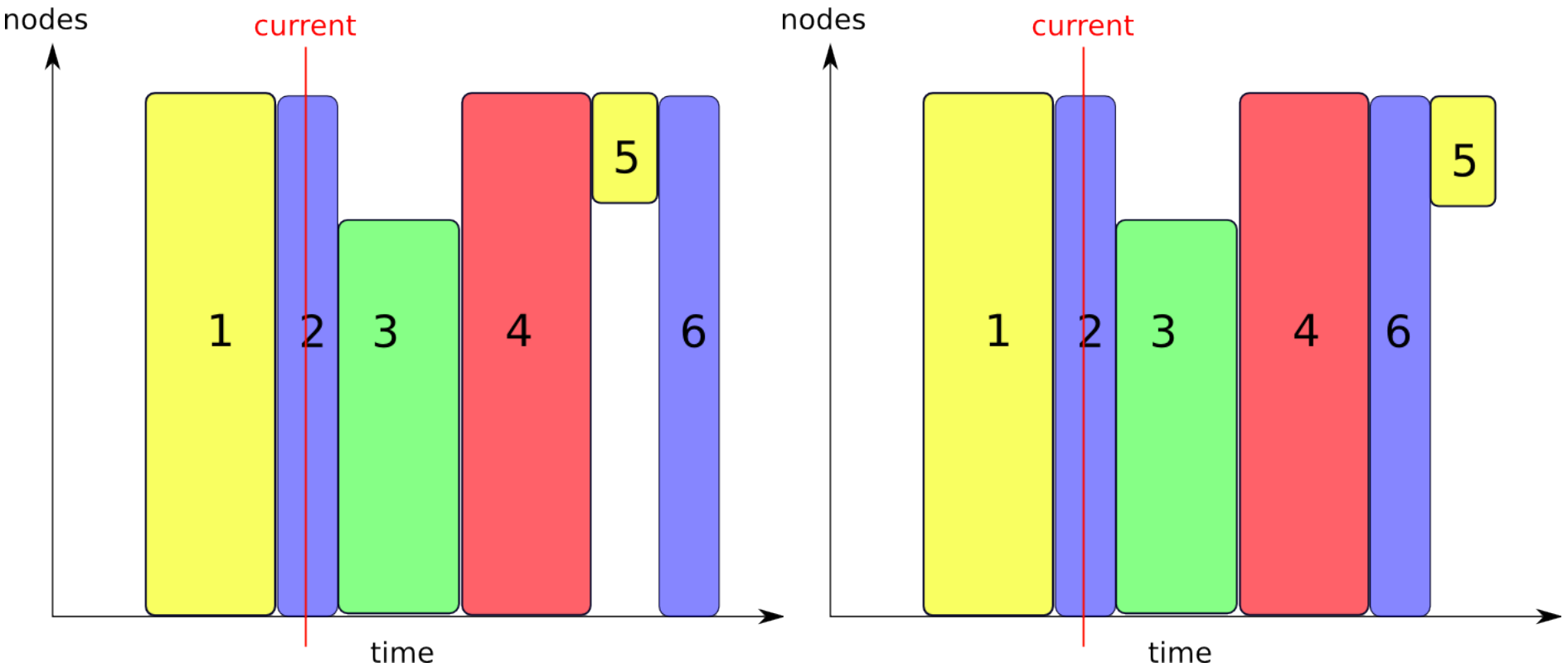
First-Fit (Backfilling)

Remplissage des trous si l'ordre des tâches soumises antérieurement n'est pas modifié



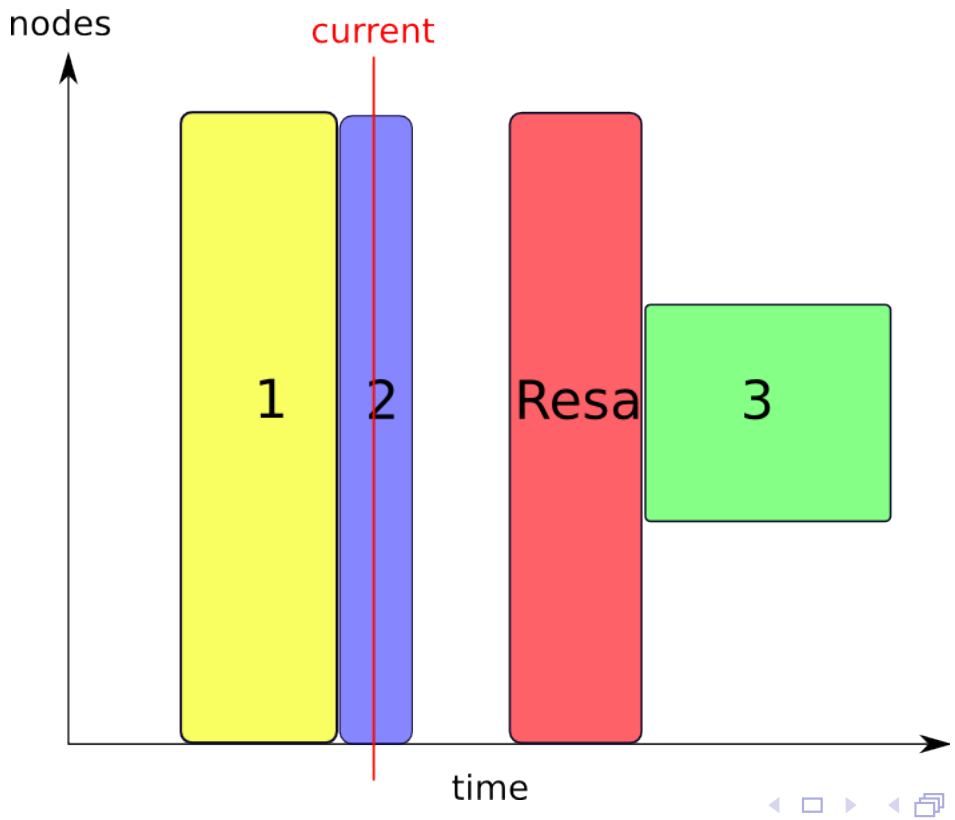
FairSharing (partage équitable)

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

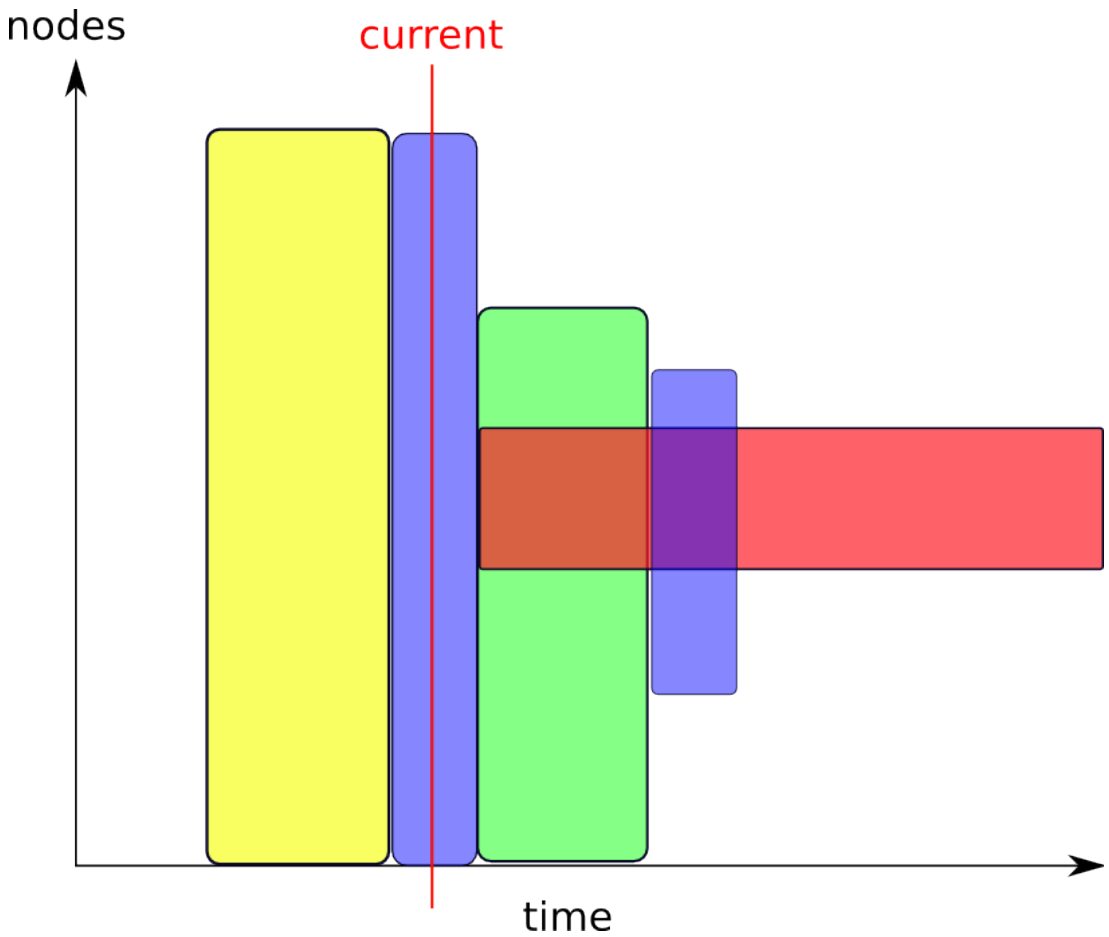


Réservation (*Advance Reservation*)

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

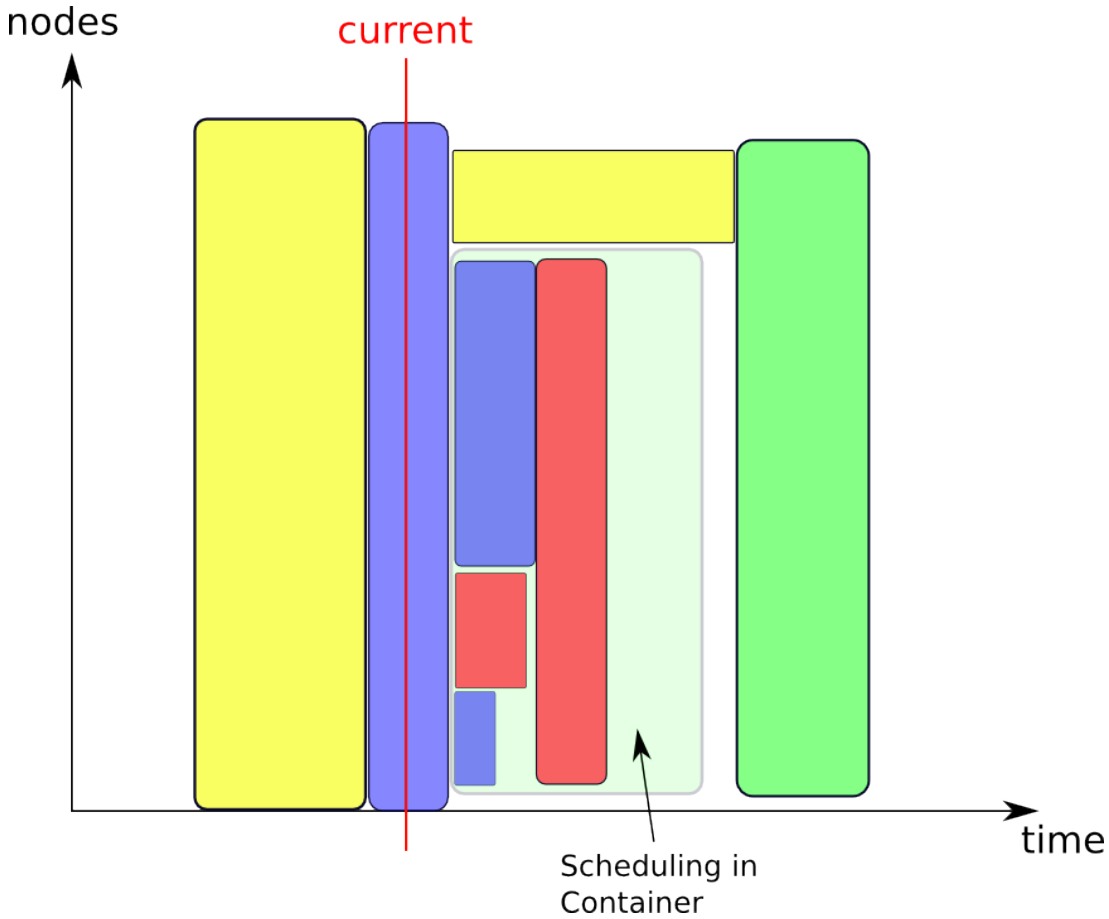


TimeSharing



Récursivité

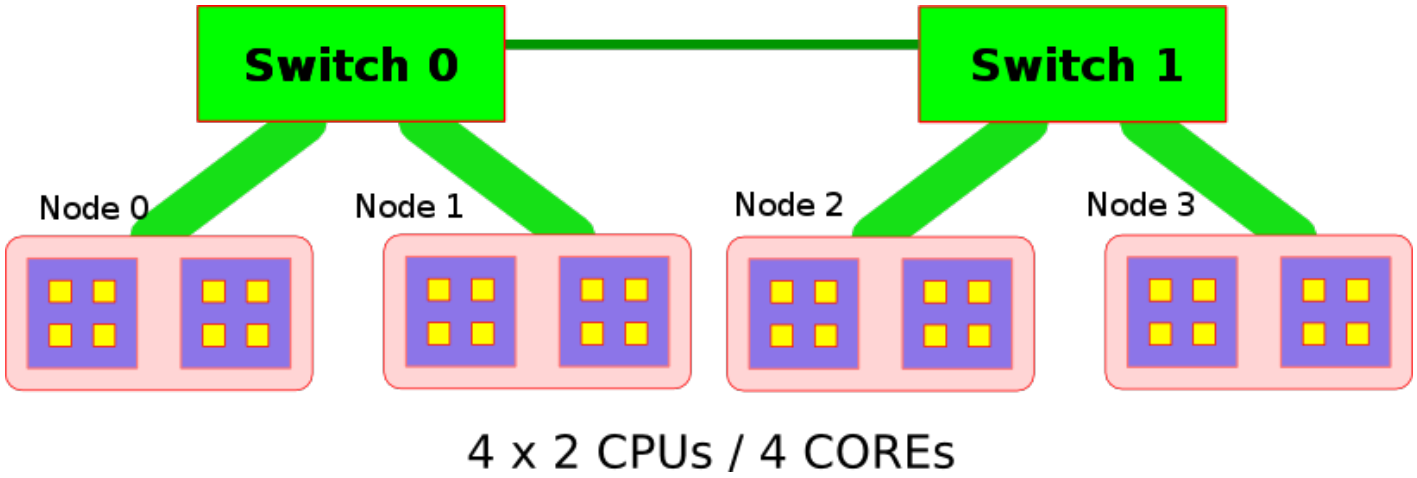
Faire de l'ordonnancement dans une allocation/réservation. Intéressant pour formation, démo, partage de ressource plus flexible par groupe d'utilisateurs / projet. **Tâche de type container.**



Contraintes Topologiques

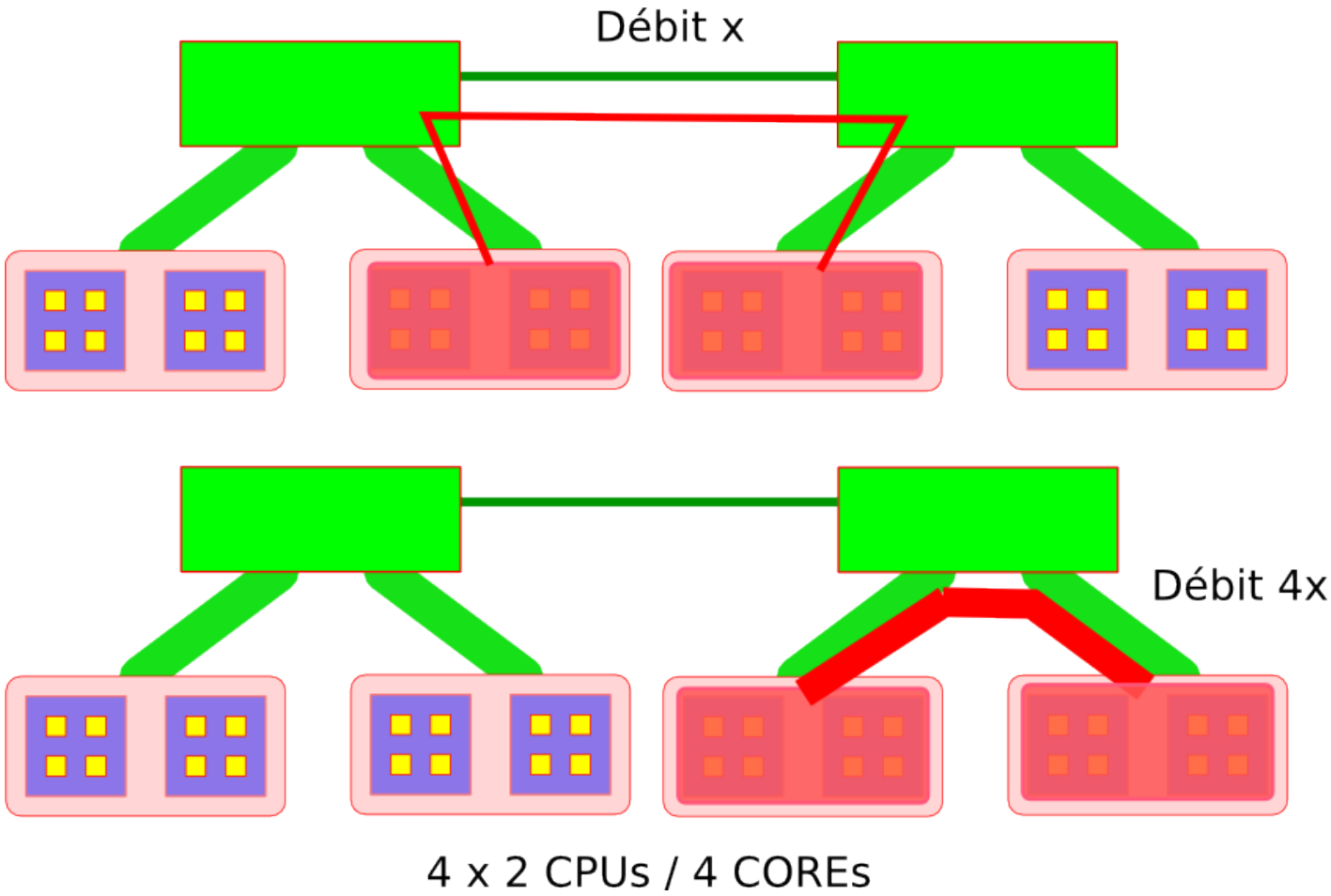
Evolution du matériel

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



Contraintes Topologiques hiérarchiques

Problème avec les applications parallèles sensible au débit communication.

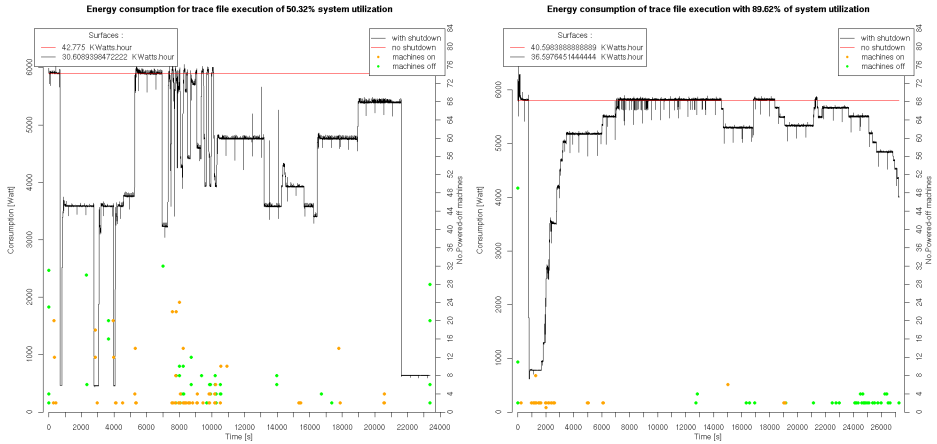


Contraintes Topologiques hiérarchiques

- Notion de hiérarchie dans les requetes :
  oarsub -l switch=1/nodes=2/cpu=2/core=2 appli-parallèle
  = 1x2x2x2 = 8 coeurs
- Gestion des cpusets linux (attention à l'affinité même au sein d'un cpuset)

Gestion de l'énergie

- **Arrêt des machines inutilisées (réveil aux besoins)**
- Priorité heures creuses/pleines par paramétrage
- Développées lors du *Google Summer Of Code 2008* (Gsoc'08)
  - Un nouveau type de job paramétrique : **powersaving** + options (cpufreq, arrêt sélectif de périphérique disque, video ..., politique spécifique)
  - Ex Job BestEffort → fréquence CPU la plus faible.



▶ ...

OAR : Monika

OAR Cluster nodes

default summary

	Free	Busy	Total
network_address	4	15	32
resource_id	32	120	256

Reservations:

Reservations for property iru=0:

rli0n0	182270	182270	182270	182270	182270	182270	182270	182270
rli0n1	182270	182270	182270	182270	182270	182270	182270	182270
rli0n2	Free	Free	Free	Free	Free	Free	Free	Free
rli0n3	Free	Free	Free	Free	Free	Free	Free	Free
rli0n4	182267	182267	182267	182267	182267	182267	182267	182267
rli0n5	Free	Free	Free	Free	Free	Free	Free	Free
rli0n6	182271	182271	182271	182271	182271	182271	182271	182271
rli0n7	182271	182271	182271	182271	182271	182271	182271	182271
rli0n8	182271	182271	182271	182271	182271	182271	182271	182271
rli0n9	StandBy	StandBy	StandBy	StandBy	StandBy	StandBy	StandBy	StandBy
rli0n10	StandBy	StandBy	StandBy	StandBy	StandBy	StandBy	StandBy	StandBy
rli0n11	182294	182294	182294	182294	182294	182294	182294	182294
rli0n12	182282	182282	182282	182282	182282	182282	182282	182282

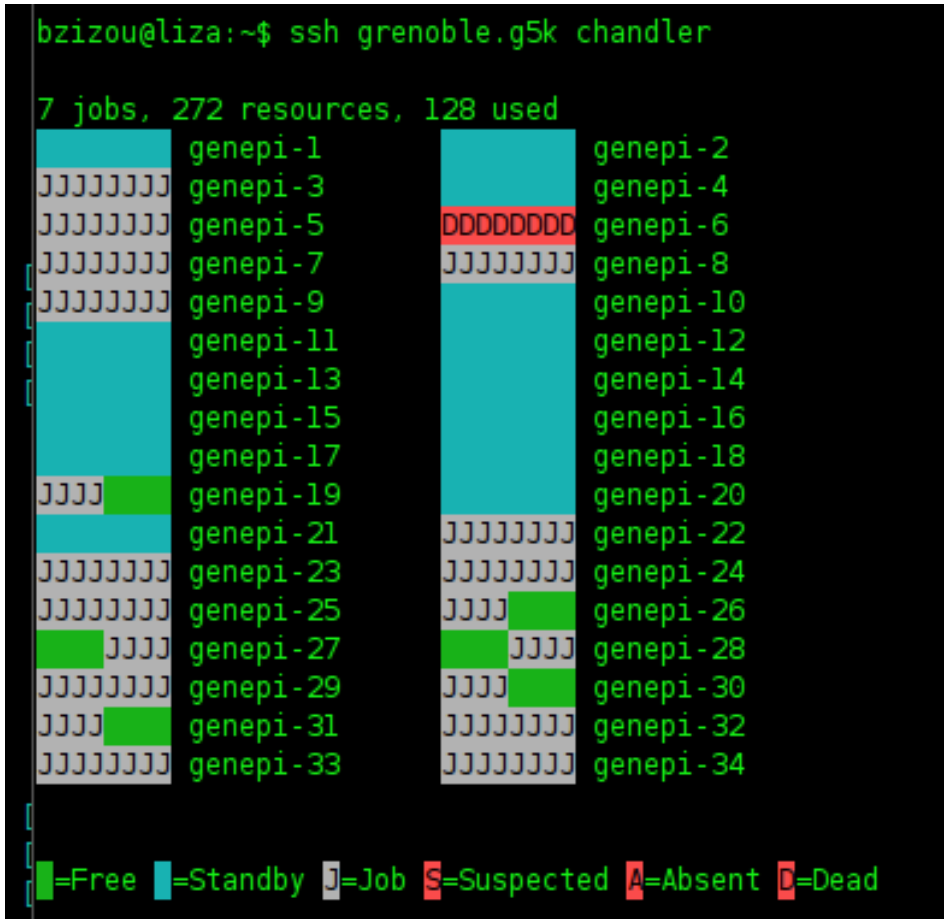
OAR : Diagramme de Gantt

API REST

- REST = protocole HTTP PUT/GET/POST/DELETE sur des ressources
- **Interface simple et puissante !**
- Pas la lourdeur des Web Services à la Soap.

- `http://mydomain.org/oarapi/resources.json`
- Donne la liste de toutes les ressources de la grappe au format json

Exemple d'utilisation de l'interface REST



Chandler : en 80 lignes de code, affichage du status des noeuds et des coeurs en mode console.

Equipe

Equipe

- **1 ingénieur senior (%50) + 2 ingénieurs CDD + 1 doctorant**
- **1 MdC (direction, codage) + 3 MdC (consultants / codage ponctuel)**
- Contributeurs (dont l'ancien ingénieur développeur principal)
- 5 stagiaires Gsoc (Google Summer of Code 08 et 09)
- Une 10zaine de personnes qui ont contribués

Références/support

Références

- Entre 7000 et 15000 coeurs, autour de 30 clusters
- Mesocentre CIMENT, Grid'5000, BRGM, Footways, Usharesoft, Université/Labo (France, Chine, Brésil,Luxembourg, US, Slovakie...)

Support

- OAR Licence GPL
- Mailing List, Gestionnaire de Bug
- Aide à l'installation (nous ou partenaire ex : BULL/Serviware)
- Support Pro (cas par cas)

Perspectives

- **Interfacage à gLite** (serveur blahp ?)
- **Ordonnanceur (nouvelle version)**
- Interface web intégrée
- Actuellement de 1000 à 10000 (ressources : noeuds, cpus, coeurs...)
- 100K ressources, numa massivement multi-coeur ?
- détection des inéfficacités ?
- Environnement de test/ de re-exécution (vision globale ordonnancement et système de fichier)
- Cluster virtuel / couplage Boinc
- Partenariats BULL, CEA

Conclusion

- **Polyvalent et personnalisable**
- Niveau de fonctionnalité comparable à la concurrence
- Très bonne stabilité
- Version stable : **2.4.1 (Thriller)** (.tgz, .deb, .rpm)
- Domaine encore en évolution

Des questions ?

http ://oar.imag.fr/

Liens


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


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


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

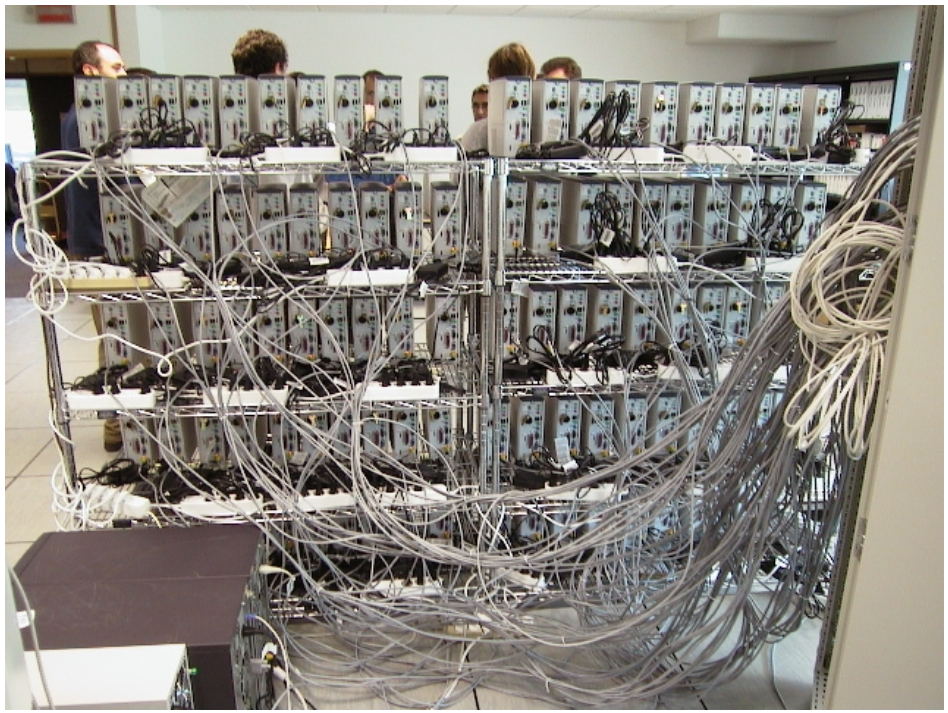

SLURM
www.llnl.gov/linux/slurm/


LSF
http ://www.platform.com

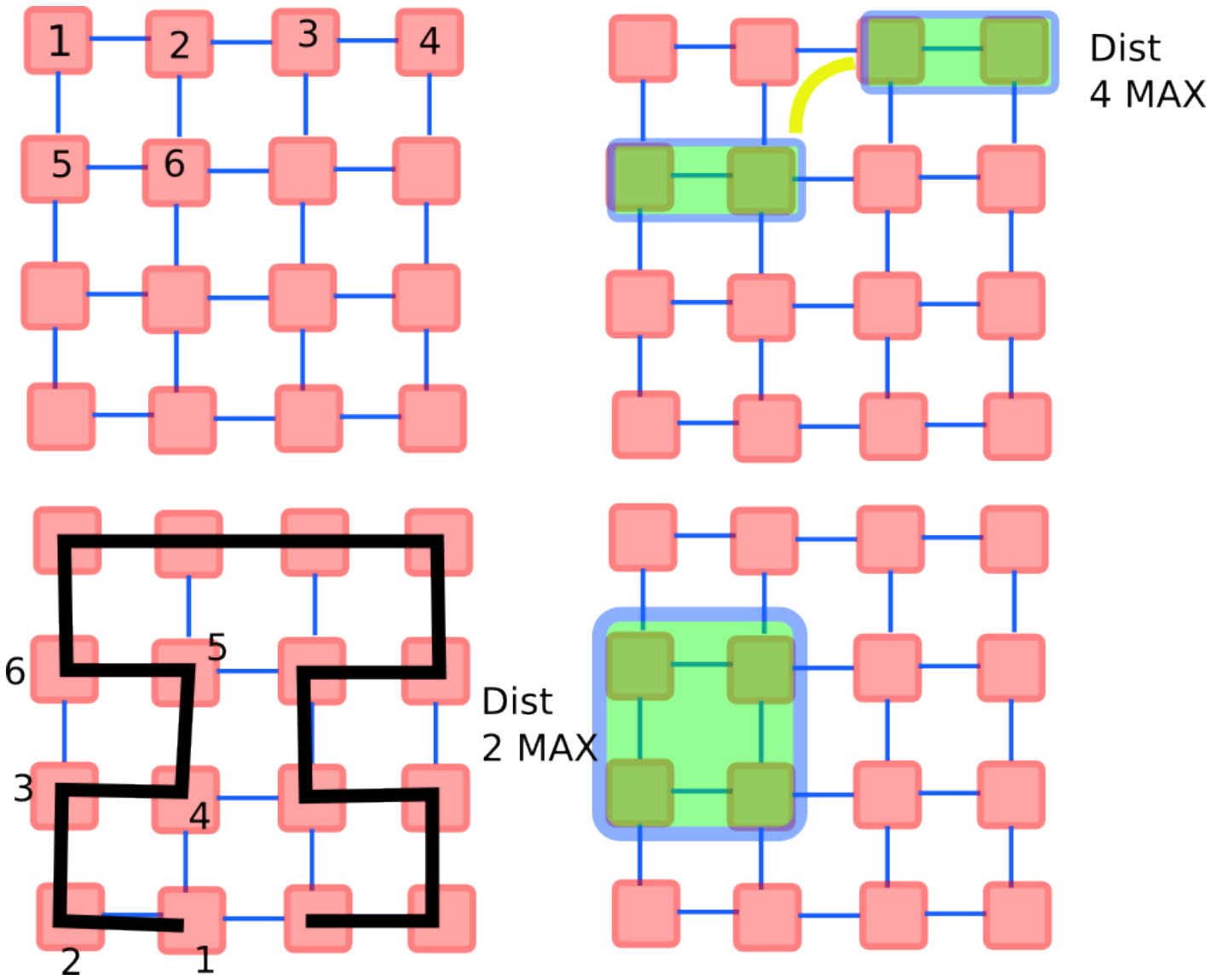

OAR
http ://oar.imag.fr

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)

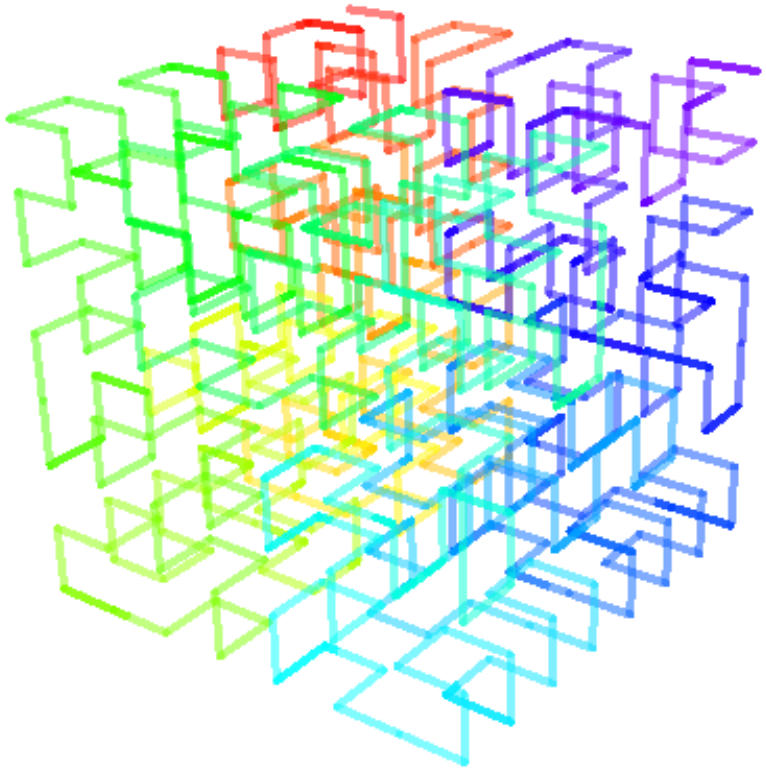


Contraintes Topologiques : grille/tore 2D



Contraintes Topologiques : grille/tore 3D

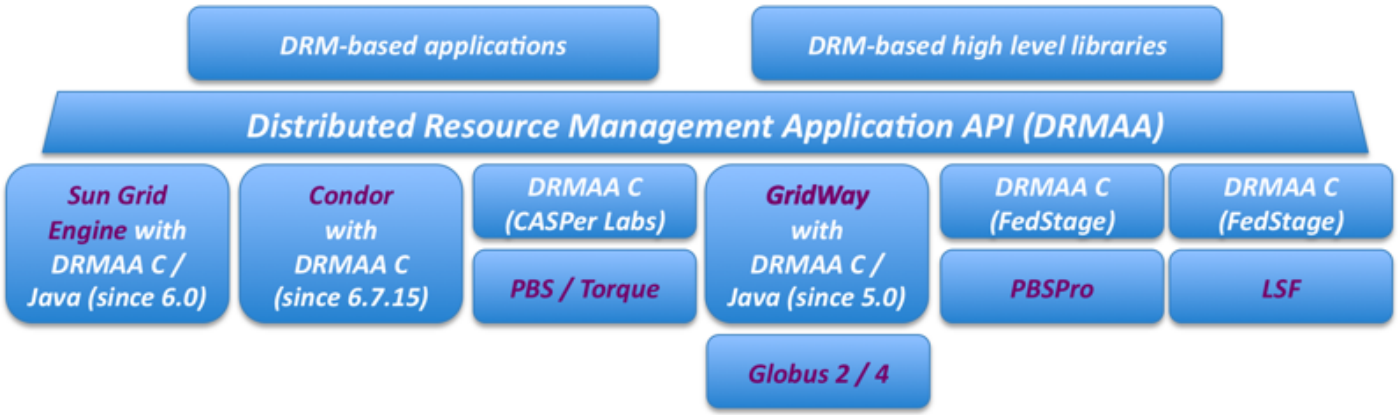
- Courbe de Hilbert (Slurm / topology)
- Wikipedia / *Hilbert_curve*



Back.

Interfaces :

- Interface commande en ligne (CLI)
- Application exemple DRMAA (v1, v2)
- Grille : Globus GT2, GT4/ OGSA-BESS, JSDL, G-Lite - BLAHp, SAGA
- Beaucoup d'interface, souvent limitatives ?



OAR : DRMAA (85%), Glite (**commandée**), REST

Le mesocentre de calcul CIMENT

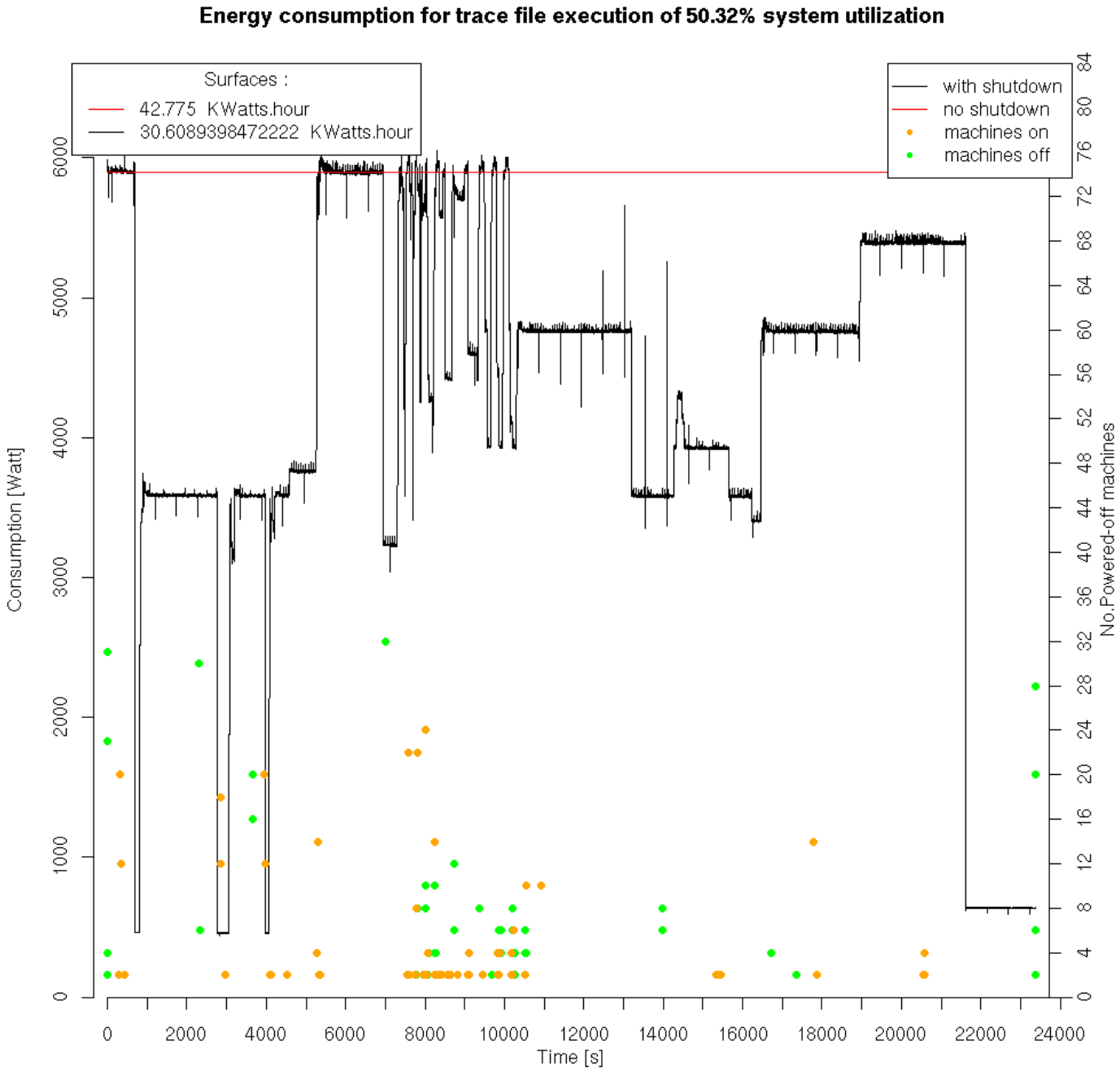
- Mesocentre de calcul intensif de l'Université Joseph Fourier (Grenoble)
- Une douzaine de calculcateurs hétérogènes, plus de 2000 cores autjourd'hui
- Particularité : mutualisation autour d'une grille légère (CiGri)
- Forte collaborations production/recherche

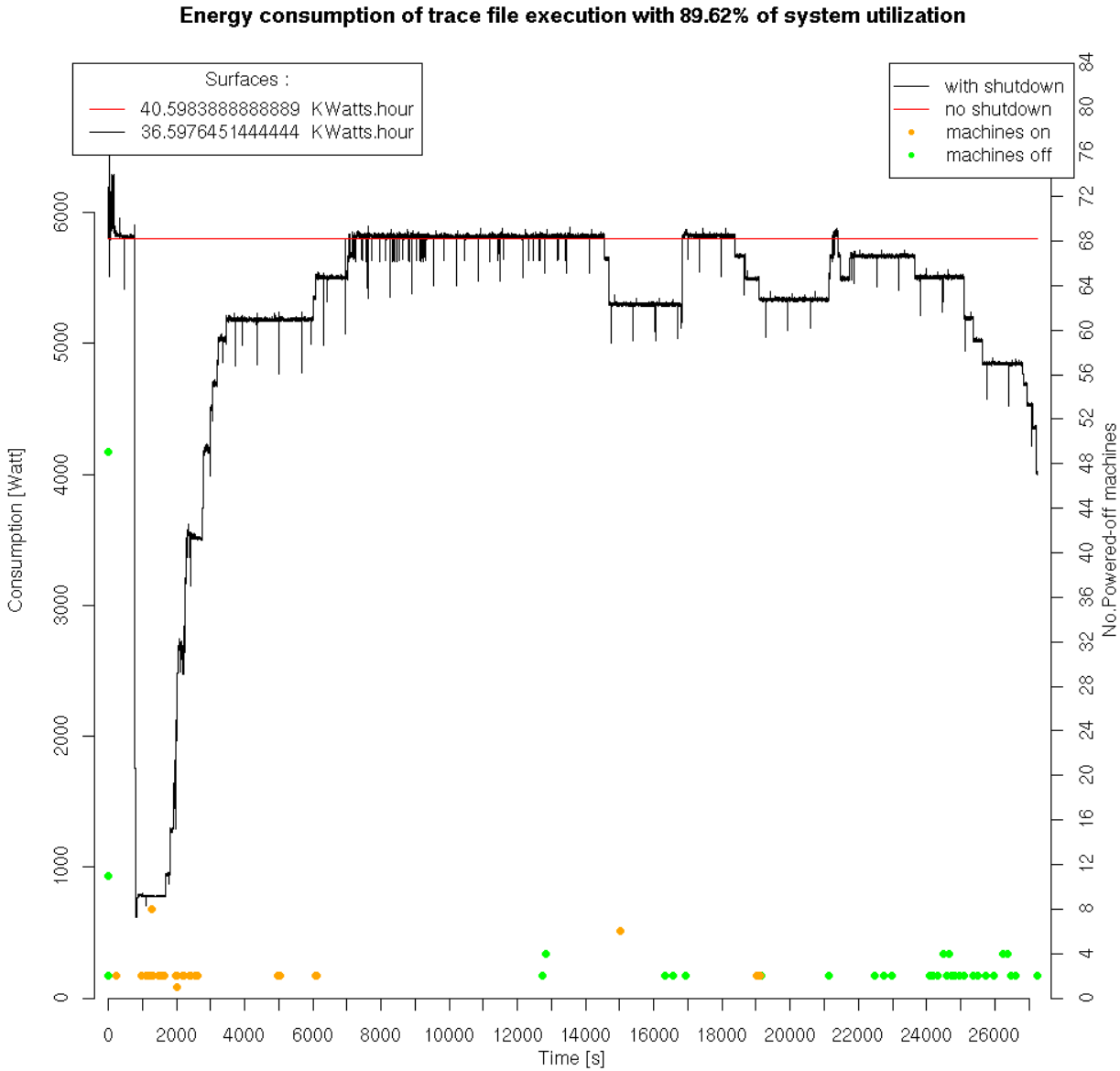
Back.

Grid 5000

- Grille pour l'expérimentation informatique
- 9 sites en France + 2 sites à l'étranger
- Interconnexion gigabit dédiée Renater
- Plus de 5000 cores aujourd'hui

Back.


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