MESO*CENTRE
Aix*Marseille Université

UTILISATION DES RESSOURCES DU MESOCENTRE

Annie Clément
Matvey Sapunov

18/05/2015

Aix*Marseille
université

Programme :

- Présentation du Mésocentre
- Comment se connecter aux machines ?
- L'environnement de l'utilisateur
- Les logiciels
- Le gestionnaire de ressources : OAR
- L'outil de visualisation : Monika

Le mésocentre d'AMU

- Crée en 2012
- Financement initial Equipex Investissement d'avenir - projet national – 10 mésocentres concernés
- Autres financements : Agence Nationale pour la Recherche par ex. CAPSHYDR (Ecole Centrale/AMU)
- Positionnement au niveau régional
- En 2014, 150 utilisateurs actifs et plus de 6 millions d'heures de calcul

Les équipements

- Total de 1404 cœurs (linux centos v6.6) **:**
  - 1152 cœurs de calculs (96 noeuds x 12) Intel X5675 Westmere à 3 GHz - 2.3 To de RAM – 14 Tflops.
  - 128 cœurs à forte mémoire (SMP) : Bullx S6010 Intel E7-8837 à 2.6 GHz - 2 To de RAM.
  - 60 cœurs à forte mémoire (SMP) : Dell R720 Intel Xeon E5-2670 à 2.6 GHz - 512 Go de RAM
  - 36 coeurs GPU – Intel Xeon sur CPU E5-2670 - 7 cartes graphiques NVIDIA Tesla K20x
  - 16 coeurs sur cartes Xeon PHI
  - 12 coeurs de visualisation Dell Precision R5500 - Intel Quad Core Xeon X5650 - 2x cartes graphiques NVIDIA Quadro 5000

- Espace de stockage partagé GPFS (300 téraoctets)
- Réseaux d’interconnexion
  - Infiniband QDR – échanges entre coeur
  - Ethernet - data

Fonctionnement en mode projet

Les allocations d'heures de calcul se font par projet, porté par un coordinateur et avalisé par le comité scientifique.

3 types de projets :

- A : Projet d'une durée de 6 mois maximum pour la découverte et/ou le portage - forfait de 5 000 heures – examen immédiat des demandes
- B : Allocation annuelle, entre 10 000 et 400 000 heures, 3 sessions annuelles d'examen :
  - session principale (allocation annuelle) : début février
  - sessions secondaires (allocation valable jusqu'à la prochaine session principale) : début juin et début octobre.
- Mesochallenge : Réservation ponctuelle de la majorité des ressources sur un temps très court - examen immédiat des demandes

Les demandes d'allocation se font en ligne à partir du site web du mésocentre, rubrique déposer un projet.

Règles d’utilisation

- Charte : respect de la législation et de l’éthique
- Notification au conseil scientifique des communications et publications faites dans le cadre des projets mésocentre
- Remercier le mésocentre dans ces communications et publications
- Stockage de données et installation de logiciel sous la responsabilité des usagers
- Les comptes d’accès sont nominatifs, personnels et non cessibles.

Comment se connecter au Mésocentre ?

- **Faire partie d'au moins un projet actif et avoir un compte utilisateur**
  - identifiant et mot de passe
  - quota d'heures par projet

- **Se connecter à la frontale en accès ssh :**
  - Linux / macOS – depuis un terminal :

```
ssh identifiant@login.ccamu.u-3mrs.fr
```

  - Windows : logiciel type putty

- **Mot de passe et cryptage par clé ssh**

Environnement utilisateur

- **Espaces de stockage alloués :**
  - /home/utilisateur : stockage persistant, partagé en NFS, 5Go, avertissement
  - /tmp : stockage temporaire sur disque SSD, local au noeud
  - /scratch : calcul, partagé en GPFS, 9To, avertissement

Ces espaces sont sous la responsabilité des usagés, il n’y a pas de sauvegardes automatiques => *Sauvegarde usagers sur une machine locale par scp, sftp, rsync*

- **Utilisation des modules :** permet de configurer, à la demande, l’environnement de l’utilisateur. Par exemple définir le compilateur à utiliser.
  - *module avail* liste des modules disponibles
  - *module list* liste les modules chargés dans l’environnement
  - *module load intel/15.0.0* charge le compilateur intel 15.0
  - *module unload intel/15.0.0* décharge le compilateur intel 15.0
  - *module purge* décharge tous les modules de l’environnement

Environnement utilisateur

- **Espaces de stockage alloués :**
  - /home/utilisateur : stockage persistant, partagé en NFS, 5Go, avertissement
  - /tmp : stockage temporaire sur disque SSD, local au noeud
  - /scratch : calcul, partagé en GPFS, 9To, avertissement

Ces espaces sont sous la responsabilité des usagés, il n'y a pas de sauvegardes automatiques => *Sauvegarde usagers sur une machine locale par scp, sftp, rsync*

- **Utilisation des modules :** permet de configurer, à la demande, l'environnement de l'utilisateur. Par exemple définir le compilateur à utiliser.
  - *module avail* liste des modules disponibles
  - *module list* liste les modules chargés dans l'environnement
  - *module load intel/15.0.0* charge le compilateur intel 15.0
  - *module unload intel/15.0.0* décharge le compilateur intel 15.0
  - *module purge* décharge tous les modules de l'environnement

Environnement utilisateur

- **Le karma :** « note » propre à chaque utilisateur qui fluctue en fonction des calculs réalisés et dont l'ordonnanceur des travaux tient compte
- **Les quotas** : heures consommées par projet / espace disque

```
*---------------------------------------------------------------------
| On project 15a009: 0.0/5000 (0%) hours have been consumed
| On project 15b005: 4800.0/20000 (24%) hours have been consumed
| You are using 1072/4882 MB (21%) on /home
| You are using 0.00/9.00 TB ( 0%) on /scratch
*---------------------------------------------------------------------
```

Les bibliothèques, logiciels, utilitaires

- **Mis à disposition :**
  - Installés sur partition /softs
  - Modules associés

- **Installés par les utilisateurs :**
  - Sur un de leurs espaces disque
  - Sous leur responsabilité

Si le logiciel souhaité n'est pas proposé, le mésocentre peut étudier son acquisition et son installation

Few more words about software

- **Respect other users. !NEVER ! run any CPU consuming code on login machine**
  - Iogin is used as a frontend to computational nodes
  - One user can slow down the work of hundreds

- **A rule of thumb: Libraries and compilers are installed and maintained by mesocentre team. End user application is installed by a user**
  - If a user believes that application can be useful for other users he/she should contact mesocentre team, so we will concider to install it systemwide as a module
- **There is no magic. If your application is not developed with MPI in mind, most likely it will be executed on a single core while others nodes/cores allocated for your job will be idle. Know how to execute your code with correspondend versions of MPI**
  - mvapich2
    - HOSTS=$(wc -l ${OAR_NODEFILE} | awk '{print $1}')
    - mpiexec -launcher ssh -launcher-exec /usr/bin/oarsh -f ${OAR_NODEFILE} -iface ib0 -n ${HOSTS} ./application
  - OpenMPI
    - HOSTS=$(wc -l ${OAR_NODEFILE} | awk '{print $1}')
    - mpirun -n "${HOSTS}" -machinefile "${OAR_NODEFILE}" ./application

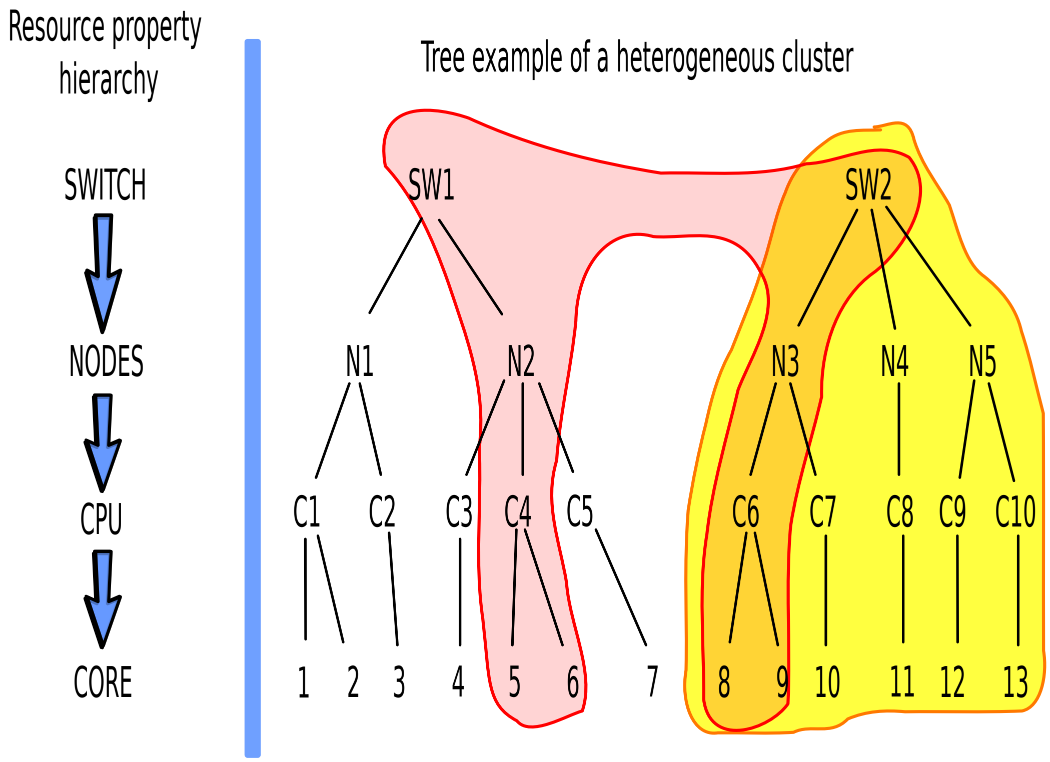
OAR features

- Batch and Interactive jobs
- Multi-queues with priority
- Reservation
- Support of moldable tasks
- Epilogue/Prologue scripts
- Suspend/resume jobs
- Checkpoint/resubmit
- Hierarchical resource requests (handle heterogeneous clusters)
- Full or partial time-sharing.
- Licenses servers management support.
- Best effort jobs : if another job wants the same resources then it is deleted automatically

OAR architecture

- **server** node which runs the oar server daemon and a database which store all job related information
  - Key component
- **frontend** node on which you will be allowed to login and to reserve computing resources
  - login.ccamu.u-3mrs.fr
- **computing** nodes on which the jobs will run
  - node001 – node096
  - smp001 - smp004
  - visu
  - phi001
  - gpu001
- **visualization** node on which all the visualization web interfaces are accessible
  - Not available from external network

Resource allocation



You can configure your own hierarchy with the property names that you want

```
oarsub -l /switch=2/nodes=1/cpu=1/core=2
```

This command reserves 2 cores on a cpu on a node on 2 different switchs (so 2 computers)

```
oarsub -l /switch=1
```

This command reserves 1 switch entirely

Wanted resources have to be described in a hierarchical manner

Complete syntax :
"{ sql1 }/prop1=1/prop2=3+{sql2}/prop3=2/prop4=1/prop5=1+...,walltime=HH:mm:ss"

walltime is always the last parameter

Examples :

- nodes=1/core=4,walltime=80:00:00
- core=2,walltime=168:00:00
- nodes=2,walltime=30:00:00
- host=16,walltime=47:59:00
- nodes=5/core=6,walltime=1:59:00

Fine resource allocation

Fine resource selection is done by using properties attributed to a resource

- SQL syntax
  - "cluster = 'YES' AND shortnode = 'NO' AND host NOT IN 'gpu001'"
  - "((smp='YES' and host='smp004') AND shortnode = 'NO') AND host NOT IN ('gpu001')"
  - "smp and nodetype='SMP512Gb'"

- Shortcuts
  - cluster → "cluster = 'YES'"
  - smp → "smp = 'YES'"
  - visu → "visu = 'YES'"
  - gpu → "gpu = 'YES' AND visu = 'NO'"
  - phi → "phi = 'YES'"

OAR resource states

- oarnodes – command to display resource related information
- OAR resource states : oarnodes -s

```
node002
        13 : Alive
        14 : Alive
        15 : Alive
        16 : Alive
        …
        23 : Alive
        24 : Alive
```

- **Alive**: the resource is ready to accept a job.
- **Absent**: the oar administrator has decided to pull out the resource. This resource can come back.
- **Suspected**: OAR system has detected a problem on this resource and so has suspected it. This resource can come back automatically or manually.
- **Dead**: The oar administrator considers that the resource will not come back and will be removed from the pool

Resource properties

Display available ressource properties : oarnodes -r **resource_id**

- freq=3.07
- cpuset=9
- model=X5675
- smp=NO
- phi=NO
- gpudevice=0
- cpu=180
- swib=9
- gpu=NO
- ib=YES
- board=90
- mem=24
- type=default
- shortnode=NO
- gpunum=0
- deploy=NO
- core=1080
- cluster=YES
- ip=192.168.71.90
- visu=NO
- available_upto=2147483646
- nbcores=12
- nodetype=Westmere
- desktop_computing=NO
- last_available_upto=2147483646
- network_address=node090
- host=node090
- vgldisplay=:0.0
- last_job_datebesteffort=YES
- vncdisplay=0

Project in OAR

A user can participate in different scientific activities. To simplify accounting of the consumed resources by activity a notion of the project has been introduced since OAR version 2.5

Each user in mesocentre has a corresponding project

- One user can be registered in several projects

Attributing specific project for a job is done with --project = ProjectName switch

If no project name is given the default project for the user is used

- In case of several projects a reverse sort is applied to the list of projects and the top project is selected as the default one
    - 14b015, 14a005, 14b025, 14b005 → 14b025 is the default project

On a computer node the name of the project is stored in OAR_PROJECT_NAME variable

OAR queues

Prioritization of the jobs and used scheduler is highly depend of the nature of your job. Jobs with high walltime have lower priority then short jobs

admin

- priority = 10
- scheduler = timesharing_and_fairsharing

development

- priority = 9
- scheduler =timesharing

short

- priority = 7
- scheduler = timesharing_and_fairsharing

medium

- priority = 5
- scheduler = timesharing_and_fairsharing

long

- priority = 3
- scheduler = timesharing_and_fairsharing

default

- priority = 2
- scheduler = timesharing_and_fairsharing

besteffort

- priority = 0
- scheduler = timesharing_and_fairsharing

OAR queues

You can specify the queue name with: -q queue_name switch

- Automatic queue routing can override the value specified by a user

Automatic queue routing is taking into account the walltime value specified in job description

- **development** : 2 hours
- **short** : 12 hours
- **medium** : 48 hours
- **long** : 168 hours (a week) not available for SMP jobs

If you need the **development** or the **besteffort** queue you must specify the name of the queue explicitly

Development and besteffort queues

Jobs in the **besteffort** queue can be killed at any moment. Therefor these jobs can use any available resource in a given moment of time

- Ideal for massive Monte-Carlo simulations or any other kind of jobs which can be suddenly interrupted

Certain resources can be attached to a specific queue

Resources with property shortnode=YES are reserved for the jobs in the **development** queue

Properties can be assigned, changed or removed automatically

- During the working hours 40 nodes are reserved for development
- During the weekends only 10 nodes reserved for development queue
- Reservation is removed at midnight so all nodes are accessible for long term jobs

Job submission

The user can submit a job with command oarsub

- Passive jobs – OAR sends a script on execution on requested resources
- Interactive jobs – OAR is returning a login shell on requested resources
  - Ideal for debugging purpose

```
oarsub -p "smp and nodetype='SMP512Gb'" -l host=3,walltime=47:59:00 --project
11a011 script_name
```

- passive job
- 3 hosts for 47 hours – long queue
- Project to account used resources is 11a011
- Requested resources is a smp machine with property SMP512Gb

```
oarsub -l nodes=1/core=4,walltime=1:59:00 -p "host='node088'" -q development
-I
```

- Interactive job
- 4 cores on a single node for 2 hours in the development queue
- Requested resource is a specific machine : node088

Job submission

To connect to already running job use -C switch

```
oarsub -C 323847
```

- Interactive job

Request that the job starts at a specified time

```
oarsub -r "2014-12-01 11:00:00" -l /nodes=12/core=6 script_name
```

Job reservation status

- **none**: the job has no reservation
- **toSchedule**: the job has a reservation and must be approved by the scheduler
- **scheduled**: the job has a reservation and it's scheduled by OAR

Parametric job submission

Submit an array job with 10 identical subjobs:

```
oarsub -l /nodes=4 /home/users/toto/prog --array 10
```

Parametric job with parameters stored in a file params.txt

```
# my param file
#a subjob with single parameter
100
#a subjob without parameters
""
#a subjob with string containing spaces as delimiter for parameters
"arg1a arg1b arg1c" "arg2a arg2b"
```

OAR generates 3 jobs and a special identifier called OAR_ARRAY_ID

```
oarsub /home/test/prog --array-param-file /home/test/params.txt
```

- OAR_JOB_ID=323848
- OAR_JOB_ID=323849
- OAR_JOB_ID=323850
- OAR_ARRAY_ID=323848

Job submission

User can prepare a script with OAR directives which can be scanned during script submission. The script has to have exec permissions

```
chmod +x /home/username/script.oar
```

Script example (file /home/username/script.oar):

```
#!/bin/bash

#OAR -n test

#OAR --notify mail:matvey.sapunov@univ-amu.fr

#OAR -l nodes=2/core=8,walltime=50:00:00

#OAR -p cluster

#OAR --project 14a026

#OAR -O OAR.%jobid%.out

#OAR -E OAR.%jobid%.err

/home/username/program
```

Submit the script :

```
oarsub -S /home/username/script.oar
```

Job notifications

User notification can be done via e-mail or a script

- The user wants to receive an email
  - The syntax is "**mail**:name@domain.com".
  - The subject of the mail is of the form: *OAR* [TAG]: job_id (job_name) on OAR_server_hostname
- The user wants to launch a script:
  - The syntax is "**exec**:/path/to/script args".
  - OAR server will connect (using OPENSSH_CMD) on the node where the oarsub command was invoked and then launches the script with the following arguments : job_id, job_name, TAG, comments

TAG can be:

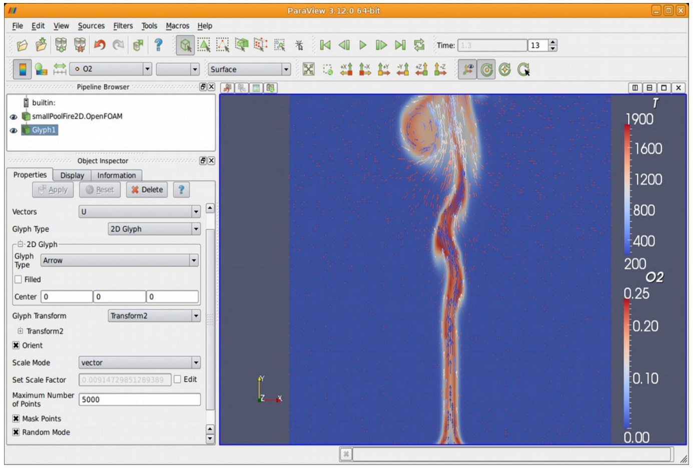
- **RUNNING** : when the job is launched
- **END** : when the job is finished normally
- **ERROR** : when the job is finished abnormally
- **INFO** : used when oardel is called on the job
- **SUSPENDED** : when the job is suspended
- **RESUMING** : when the job is resumed

Visualisation job

Special type of job dedicated for visualisation. Can execute a 3-D application with GUI like OpenFOAM, Molekel etc

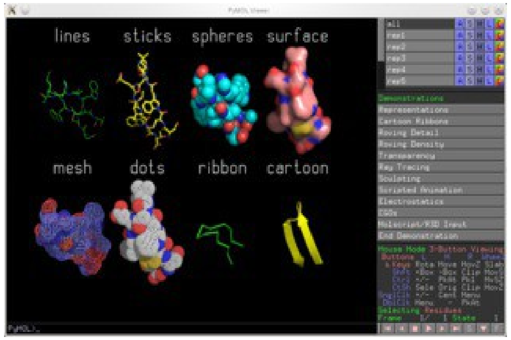
From the front-end, to ask for a visualisation session:

```
[user@login ~]$ visu_sub.sh
[ADMISSION RULE] Modify resource description with type constraints
OAR_JOB_ID=559
Waiting job 559 to be running.
You can launch your VNC viewer on the address:
visu.ccamu.u-3mrs.fr:11
Password: 28405608
Note: This password is only valid ONE time. If you want to generate another password for this session then type:
OAR_JOB_ID=559 oarsh visu vncpasswd -o -display visu:11
[user@login ~]$
```



builtin:
smallPoolFire2D.OpenFOAM
Glyph1

Vectors: U
Glyph Type: 2D Glyph
2D Glyph
Glyph Type: Arrow
□ Filled
Center: 0 0 0
Glyph Transform: Transform2
Transform2
☒ Orient
Scale Mode: vector
Set Scale Factor: 0.00914729851289389 □ Edit
Maximum Number of Points: 5000
☒ Mask Points
☒ Random Mode



Visualisation job

To connect, you need a VNC client. We advise you to use tigervnc version 1.2 or higher

From your local machine, start tigervnc and connect to the indicated address given at the submission time and with the associated password

Hostname: **visu.ccamu.u-3mrs.fr:11**

Password: **28405608**

It is possible to connect several people simultaneously on the same session (each connection needs a different password).

To ask for a new password (from the front-end):

```
OAR_JOB_ID=559 oarsh visu vncpasswd -o -display visu:11
```

By default, tigervnc does not accept the sharing, it is important to tick the option **Shared** (don't disconnect other viewers)

On the visualisation node, to start a 3D application from the shell terminal:

```
[user@login ~]$ vglrun /chemin/vers/mon/application
```

OAR job states

- **Waiting**: the job is waiting OAR scheduler decision
- **Hold**: user or administrator wants to hold the job. So it will not be scheduled by the system
- **toLaunch**: the OAR scheduler has attributed some nodes to the job. So it will be launched
- **toError**: something wrong occurred and the job is going into the error state
- **toAckReservation**: the OAR scheduler must say "YES" or "NO" to the waiting oarsub command because it requested a reservation
- **Launching**: OAR has launched the job and will execute the user command on the first node
- **Running**: the user command is executing on the first node
- **Suspended**: the job was in Running state and there was a request to suspend this job. In this state other jobs can be scheduled on the same resources
- **Finishing**: the user command has terminated and OAR is doing work internally
- **Terminated**: the job has terminated normally
- **Error**: a problem has occurred

Job monitoring

To show information about a job or set of jobs use oarstat command

Status of the job

```
oarstat -sj 323847

323847: Terminated
```

Job's event

```
oarstat -ej 323847

2014-11-30 19:09:32| 323847| SWITCH_INTO_TERMINATE_STATE: [bipbip 323847] Ask to
change the job state
```

Information about the job

```
oarstat -j 323847

Job id     Name           User           Submission Date     S Queue

323847     interactive    msapunov       2014-11-30 19:07:49 T developmen
```

Job details : oarstat -fj 323847

```
Job_Id: 323847
    job_array_id = 323847
    job_array_index = 1
    name = interactive
    project = rheticus
    owner = msapunov
    state = Terminated
    wanted_resources = -l "{type = 'default'}/host=1/core=4,walltime=1:59:0"
    types =
    dependencies =
    assigned_resources = 1045+1046+1047+1048
    assigned_hostnames = node088
    queue = development
    command =
    launchingDirectory = /home/msapunov
    stdout_file = OAR.interactive.323847.stdout
    stderr_file = OAR.interactive.323847.stderr
    jobType = INTERACTIVE
    properties = ((host='node088') AND cluster='YES') AND host NOT IN ('gpu001')
    reservation = None
    walltime = 1:59:0
    submissionTime = 2014-11-30 19:07:49
    startTime = 2014-11-30 19:07:50
    stopTime = 2014-11-30 19:09:32
    cpuset_name = msapunov_323847
    initial_request = oarsub -l nodes=1/core=4,walltime=1:59:00 -p host='node088' -q development -I
    message = FIFO scheduling OK
    scheduledStart = no prediction
    resubmit_job_id = 0
    events = [2014-11-30 19:09:32] SWITCH_INTO_TERMINATE_STATE:[bipbip 323847] Ask to change the job state
```

Accounting

Accounting information between two dates

```
oarstat --accounting '2014-11-18, 2014-11-19' -u msapunov

Usage summary for user 'msapunov' from 2014-11-18 to 2014-11-19:

Start of the first window:   2014-11-17 01:00:00

End of the last window:      2014-11-19 00:59:59

Asked consumption:           897800 ( 10 days 9 hours 23 minutes 20 seconds )

Used consumption:            259704 ( 3 days 8 minutes 24 seconds )

By project consumption:

  rheticus:

    Asked :                  897800 ( 10 days 9 hours 23 minutes 20 seconds )

    Used :                   259704 ( 3 days 8 minutes 24 seconds )

    Last Karma :             Karma = 0.003
```

Important note : consumption = walltime * number of cores

Useful commands

The command to delete or to checkpoint the job(s) : **oardel**

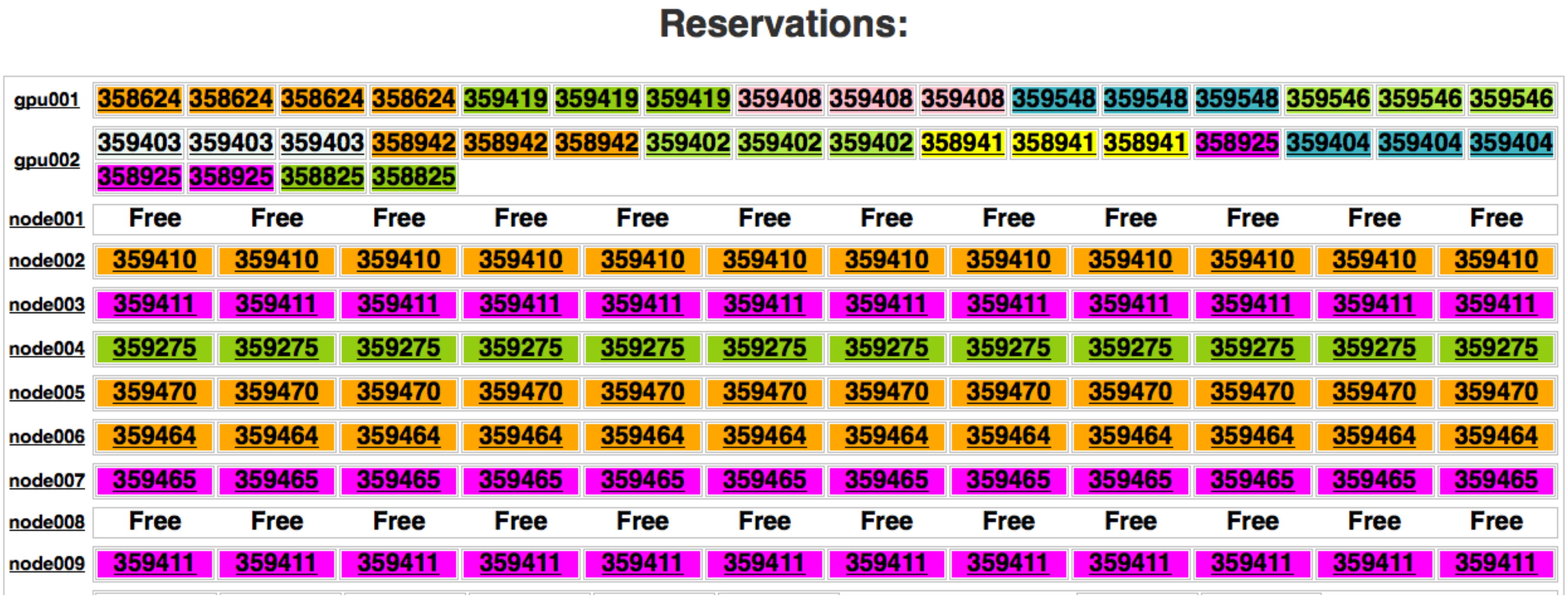
- `oardel 323848 323849`
  - Delete two jobs 323848 and 323849
- `oardel -c 323849`
  - Send a checkpoint signal to the job 323849 (type of signal defined as oarsub option)

User can hold a job in OAR batch scheduler with command **oarhold**

- Remove a job from the scheduling queue if it is in the "**Waiting**" state
- Suspend a job if it is in "**Running**" state, sending SIGINT signal

Ask OAR to change a job states into "**Waiting"** when it is on "**Hold"** or in "**Running"** if it is "**Suspended"** state with **oarresume** command

Outil de visualisation : Monika



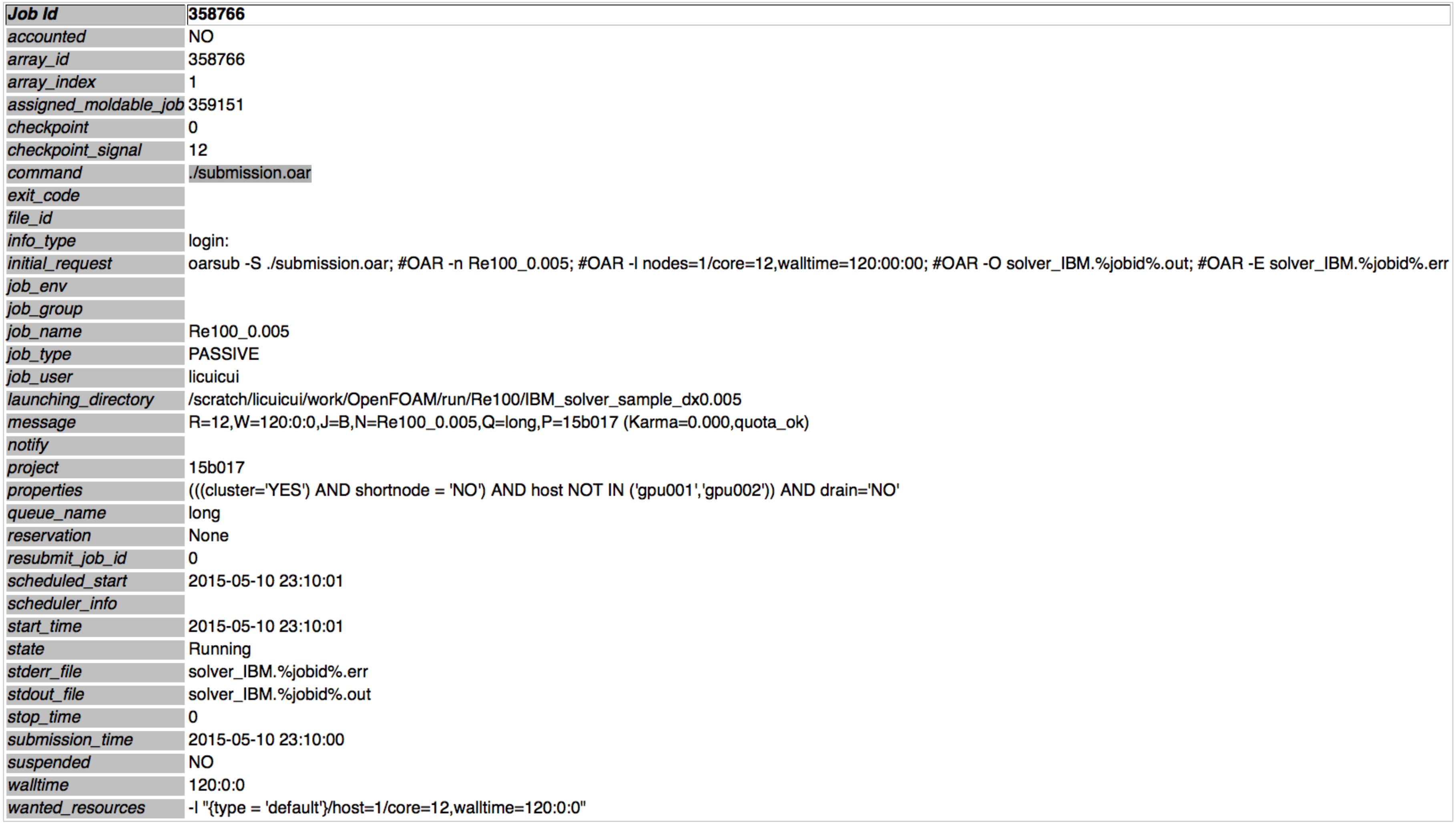
Reservations:

gpu001	358624	358624	358624	358624	359419	359419	359419	359408	359408	359408	359548	359548	359548	359546	359546	359546
gpu002	359403	359403	359403	358942	358942	358942	359402	359402	359402	358941	358941	358941	358925	359404	359404	359404
	358925	358925	358825	358825												
node001	Free	Free	Free	Free	Free	Free	Free	Free	Free	Free	Free	Free				
node002	359410	359410	359410	359410	359410	359410	359410	359410	359410	359410	359410	359410				
node003	359411	359411	359411	359411	359411	359411	359411	359411	359411	359411	359411	359411				
node004	359275	359275	359275	359275	359275	359275	359275	359275	359275	359275	359275	359275				
node005	359470	359470	359470	359470	359470	359470	359470	359470	359470	359470	359470	359470				
node006	359464	359464	359464	359464	359464	359464	359464	359464	359464	359464	359464	359464				
node007	359465	359465	359465	359465	359465	359465	359465	359465	359465	359465	359465	359465				
node008	Free	Free	Free	Free	Free	Free	Free	Free	Free	Free	Free	Free				
node009	359411	359411	359411	359411	359411	359411	359411	359411	359411	359411	359411	359411				

Etats des nœuds :

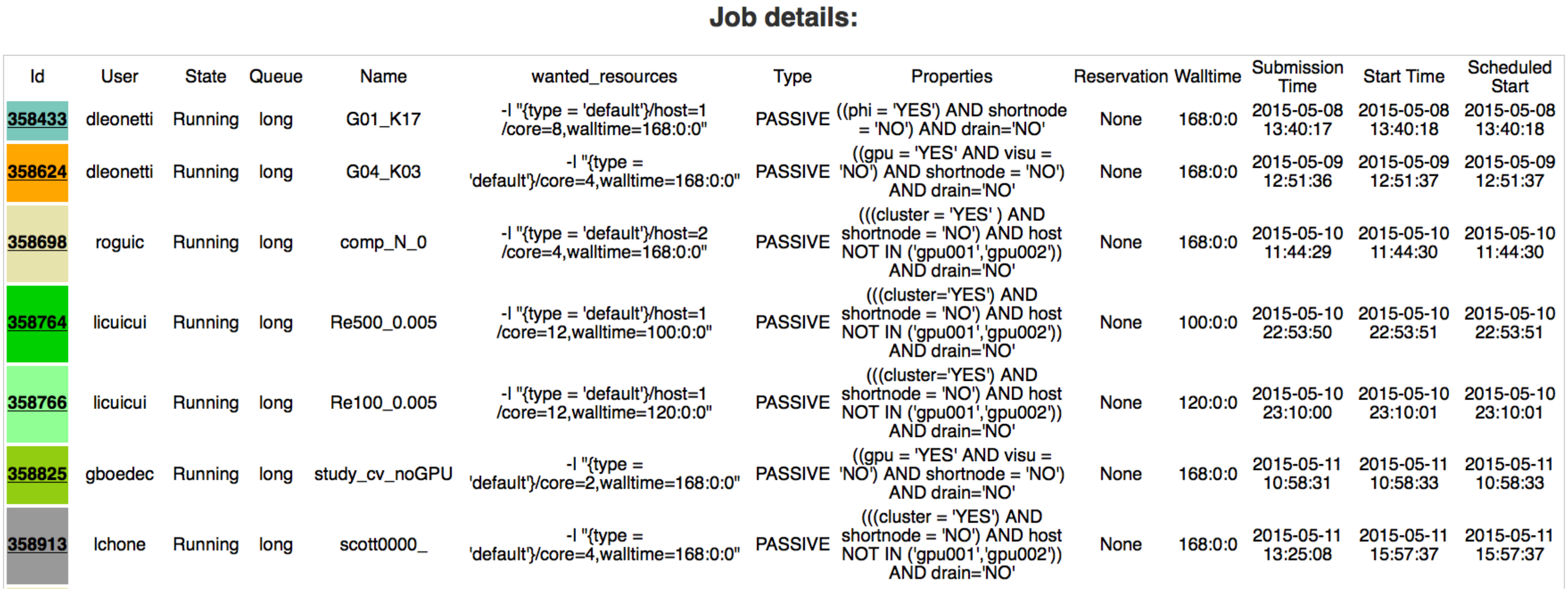
- Free, Coloré=busy, Absent, Dead, Drain

Outil de visualisation : Monika



Job Id	**358766**
accounted	NO
array_id	358766
array_index	1
assigned_moldable_job	359151
checkpoint	0
checkpoint_signal	12
command	./submission.oar
exit_code	
file_id	
info_type	login:
initial_request	oarsub -S ./submission.oar; #OAR -n Re100_0.005; #OAR -l nodes=1/core=12,walltime=120:00:00; #OAR -O solver_IBM.%jobid%.out; #OAR -E solver_IBM.%jobid%.err
job_env	
job_group	
job_name	Re100_0.005
job_type	PASSIVE
job_user	licuicui
launching_directory	/scratch/licuicui/work/OpenFOAM/run/Re100/IBM_solver_sample_dx0.005
message	R=12,W=120:0:0,J=B,N=Re100_0.005,Q=long,P=15b017 (Karma=0.000,quota_ok)
notify	
project	15b017
properties	(((cluster='YES') AND shortnode = 'NO') AND host NOT IN ('gpu001','gpu002')) AND drain='NO'
queue_name	long
reservation	None
resubmit_job_id	0
scheduled_start	2015-05-10 23:10:01
scheduler_info	
start_time	2015-05-10 23:10:01
state	Running
stderr_file	solver_IBM.%jobid%.err
stdout_file	solver_IBM.%jobid%.out
stop_time	0
submission_time	2015-05-10 23:10:00
suspended	NO
walltime	120:0:0
wanted_resources	-l "{type = 'default'}/host=1/core=12,walltime=120:0:0"

Outil de visualisation : Monika



Job details:

Id	User	State	Queue	Name	wanted_resources	Type	Properties	Reservation	Walltime	Submission Time	Start Time	Scheduled Start
358433	dleonetti	Running	long	G01_K17	-l "{type = 'default'}/host=1 /core=8,walltime=168:0:0"	PASSIVE	((phi = 'YES') AND shortnode = 'NO') AND drain='NO'	None	168:0:0	2015-05-08 13:40:17	2015-05-08 13:40:18	2015-05-08 13:40:18
358624	dleonetti	Running	long	G04_K03	-l "{type = 'default'}/core=4,walltime=168:0:0"	PASSIVE	((gpu = 'YES' AND visu = 'NO') AND shortnode = 'NO') AND drain='NO'	None	168:0:0	2015-05-09 12:51:36	2015-05-09 12:51:37	2015-05-09 12:51:37
358698	roguic	Running	long	comp_N_0	-l "{type = 'default'}/host=2 /core=4,walltime=168:0:0"	PASSIVE	(((cluster = 'YES') AND shortnode = 'NO') AND host NOT IN ('gpu001','gpu002')) AND drain='NO'	None	168:0:0	2015-05-10 11:44:29	2015-05-10 11:44:30	2015-05-10 11:44:30
358764	licuicui	Running	long	Re500_0.005	-l "{type = 'default'}/host=1 /core=12,walltime=100:0:0"	PASSIVE	(((cluster='YES') AND shortnode = 'NO') AND host NOT IN ('gpu001','gpu002')) AND drain='NO'	None	100:0:0	2015-05-10 22:53:50	2015-05-10 22:53:51	2015-05-10 22:53:51
358766	licuicui	Running	long	Re100_0.005	-l "{type = 'default'}/host=1 /core=12,walltime=120:0:0"	PASSIVE	(((cluster='YES') AND shortnode = 'NO') AND host NOT IN ('gpu001','gpu002')) AND drain='NO'	None	120:0:0	2015-05-10 23:10:00	2015-05-10 23:10:01	2015-05-10 23:10:01
358825	gboedec	Running	long	study_cv_noGPU	-l "{type = 'default'}/core=2,walltime=168:0:0"	PASSIVE	((gpu = 'YES' AND visu = 'NO') AND shortnode = 'NO') AND drain='NO'	None	168:0:0	2015-05-11 10:58:31	2015-05-11 10:58:33	2015-05-11 10:58:33
358913	lchone	Running	long	scott0000_	-l "{type = 'default'}/core=4,walltime=168:0:0"	PASSIVE	(((cluster = 'YES') AND shortnode = 'NO') AND host NOT IN ('gpu001','gpu002')) AND drain='NO'	None	168:0:0	2015-05-11 13:25:08	2015-05-11 15:57:37	2015-05-11 15:57:37

Où trouver plus d’informations ?

- **Site web du mésocentre : equipex-mesocentre.univ-amu.fr**
  - Informations générales
  - Liste des logiciels, tutoriaux
  - Accès à Monika
  - Section Suivi d’activité
- **Liste de diffusion : equipex-mesocentre@univ-amu.fr**
- **Comité technique :**
  - equipex-mesocentre-techn@univ-amu.fr
  - +33 (0)4 13 55 12 15 / 55 03 33

Questions / réponses :