



Frequently Asked Questions

Fabien Archambault

Aix-Marseille Université

2012

- 1 Rheticus' configuration
- 2 Front-end connection
- 3 Modules
- 4 OAR submission
 - Basics
 - Submission scripts
 - Queues
 - Resources
 - Projects
- 5 Visualisation
 - Asking for resources
 - Software and password
- 6 Tutorials, Libraries, softwares and contacts

Global view of the hardware:

- One front-end computer (**login**);
- 96 fine nodes (**nodeXXX**, about 12 Tflops). For each node: 12 cores, 24 GB of memory and InfiniBand QDR;
- 1 big memory node (**smp001**, about 600 Gflops) with 64 cores, 1 TB of memory;
- 1 visualisation node (**visu**) with 12 cores, 64 GB of memory and 2 NVIDIA Quadro 5000 cards (2 GB of memory each);

For the storage part, it is important to know that **no backup** is made. The usable disks are:

- Permanent storage (**/home**): personal disk folder (limited to **5 GB**);
- Temporary storage (**/scratch**): fast disk for computation (about 8 GB/s for reading) limited to 10 TB per account;
- Fine nodes computation (**/tmp**): the nodes **nodeXXX** are equipped with SSD drives (about 70 GB free). On those, it is possible to temporarily store data which will be **purged** by the end of the job.

Front-end connection:

```

$ ssh user@login.ccamu.u-3mrs.fr
user@login's password:
Last login: Xxx Xxx 00 00:00:00 0000 from xxx

  -----
 /  _  \ /  /_  _  /  /_  ( )  -----
 /  /_ /  /  _  \ /  _  \ /  //  ___ /  /  /  /  ___ /
 /  _ ,  _ /  /  /  _  /  /_ /  //  /_  /  /_  ( _  )
 /_ /  |_ /  /  /_ /  \_  /  \_  /  /  \_  /  \_  ,  _ /  ___ /

[user@login ~]$

```

Modification of the password: **passwd**

Website for Ganglia, Monika and Drawgantt:

<http://cbrl.up.univ-mrs.fr/~mesocentre/mirror.php>

The environment `modules` is used to define libraries and/or path to compilers. An example of available modules:

```
$ module avail
----- /softs/Modules -----
ATLAS/gcc/3.8.4                molekel/5.4.0
ATLAS/gcc46/3.8.4             mpich2/gcc/1.2.1
ATLAS/gcc47/3.8.4             mpich2/gcc/1.4.1
ATLAS/intel/3.8.4             mpich2/gcc46/1.4.1
ATLAS/smp/gcc47/3.8.4         mpich2/gcc47/1.4.1
ATLAS/smp/intel/3.8.4         mpich2/intel/1.4.1
[...]
```

For example, the compiler Intel 12.1 can be loaded with: `module load intel/12.1`.

The loaded modules are available with: `module list`.

To unload a module: `module unload intel/12.1`

The queue and job scheduler used is **OAR** for submitting jobs.

Basic commands are:

oarsub -I: interactive submission;

oarsub -S ./mon_script.oar: OAR script submission;

oarstat: show the submitted jobs (see also **Monika**);

oarsub -C JOB_ID: to connect to compute nodes being used. The **JOB_ID** can be obtained with **oarstat**. On the master node, it is possible to connect to other nodes by achieving the command **oarsh name_node** where **name_node** can be obtained with: **cat \$OAR_NODEFILE**;

oardel JOB_ID: delete the job. The **JOB_ID** can be obtained with **oarstat**.

Information

By default, one core is assigned to the resources. Use **nodes=X** to specify the number of hosts!

OAR options to execute scripts are:

#OAR -n name_of_job: give a name to the job;

#OAR -l resources: specify asked resources. Example: to ask all CPUs on a node for 24 hours: `-l nodes=1,walltime=24:00:00`. To ask only one CPU: `-l core=1`;

#OAR -O output: specify the standard output. For example: `output.%jobid%.out`;

#OAR -E error: specify the error output. For example: `error.%jobid%.out`;

More information: consult the online [OAR 2.5.x](#) documentation.

Important remark

Submission scripts need to be set as executables:
(`chmod +x ./mon_fichier.oar`).

Some routing and queues definition are taken into account when submitting a job. To specify the queue name: `-q queue`. It is not compulsory to specify the queue as giving a walltime will automatically direct into a `short`, `medium` or `long` queue. If you need the `development` or the `besteffort` queue you must specify it.

The queues are, by priority order:

development: very high priority queue, restricted to interactive jobs, to perform code tests;

short: for short jobs or by default (maximum 11 hours);

medium: for jobs with a maximum of 2 days;

long: for longer jobs (maximum 7 days);

besteffort: dedicated to jobs with the capability to be stopped at **any** time. In this queue, no penalty is applied to your account for resource utilisation.

To select the resources, it is possible to use the option `-p resource`.

The submission properties are:

cluster: fine nodes with fast interconnect and low latency (each nodes have 12 cores at 3.03 GHz);

smp: big memory node (1 TB for 64 cores at 2.67 GHz);

visu: visualisation node (12 cores at 2.67GHz and 2 NVIDIA Quadro 5000 card). If you ask this resource without using the `visu_sub.sh` script, your job will be set into the **besteffort** queue.

Each users have a **uniq id** to connect to Rheticus. To submit in a specific project, you should give its value:

Interactive job: `oarsub -I --project project_name [...];`

Batch job: add the option `#OAR --project project_name` in the submission file.

If your account has only one project, you do not have to specify it. It will be automatically added.

For users who have multiple projects, the default project will be the one with the most incremented number. For example, a user on projects 13b030 and 13b050 will have as a "principal" project the 13b050. If no option `--project` is set when submitted, the hours will be accounted on this project. In any cases, the project selected will be displayed at the submission:

```
[JOB PROJECT] Using project 13b050.
```

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 ~]$
```

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 and with the associated password.

It is possible to connect several people simultaneously on the same session (each connection need a different password). By default, `tigervnc` does not accept the sharing, it is important to tick the option *Shared (don't disconnect other viewers)*.

In the session, to start a 3D application for the shell terminal:

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

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

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

More information are available at the address (yet, only in French):

<http://cbrl.up.univ-mrs.fr/~mesocentre/tutoriaux.php>

A list of softwares and libraries is available at (yet, only in French):

<http://cbrl.up.univ-mrs.fr/~mesocentre/software.php>

For any technical issue, please send an email to:

equipex-mesocentre-techn@univ-amu.fr