Introduction of the resources for the CEMRACS’ projects

Fabien Archambault

Aix-Marseille Université
1. Rheticus’ configuration
2. Connection to the front-end
3. The Modules
4. OAR submission
   - Basics
   - Submission scripts
   - Resources
5. Visualisation
   - Asking for resources
   - Software and password
6. Tutorial, Libraries, softwares
Global view of the hardware:

- One front-end computer (login);
- 96 fine nodes (nodeXXX, about 12 Tflops). For each nodes: 12 cores, Intel X5675 (Westmere), 24 GB of memory and InfiniBand QDR;
- 1 big memory node (smp001, about 600 Gflops) with 64 cores, Intel E7-8837, 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);
Global view of the hardware:

- One front-end computer (**login**);
- 96 fine nodes (**nodeXXX**, about 12 Tflops). For each nodes: 12 cores, Intel X5675 (Westmere), 24 GB of memory and InfiniBand QDR;
- 1 big memory node (**smp001**, about 600 Gflops) with 64 cores, Intel E7-8837, 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 data storage, it is important to note that no backup is made! Usable disks are:

- NFS sharing (**/home**): disk for personal use (limited to 5 GB);
- GPFS sharing (**/scratch**): fast disk for computation (about 8 GB/s for reading);
- CEMRACS sharing (**/softs/cemracs**): used to share data or libraries between projects for the CEMRACS’ projects;
- Local nodes (**/tmp**): on fine nodes SSD disks are available (about 70 GB free). This folder is purged after each job.
Connection to the front-end:

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

---- ___ ___ ___
 /  \ /  \ /  \ /  \ /  \ (\)______ _______
 /  ___/  \ /  \ __/ /  \ ___/ /  \ ___/
 / __/  \ /  \ __/ /  \ __/ /  \ __/ /  \ (____
/  / /  \ /  \ /  \ /  \ /  \ /  \ /  \ /  \(__
/    |    |    |    |    |    |    |    |

| You have consumed XX hours (0%)
| On project 1010: XX hours (0%) have been consumed
| You are using XX/4882 MB (XX%) on /home
| You are using XX/9.00 TB (XX%) on /scratch

[user@login ~]$ $

Seeing resources with Ganglia, Monika and Drawgantt:

http://master.ccamu.u-3mrs.fr
The environment **modules** is used to define libraries and/or path to compilers. An example of available modules:

```bash
$ 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**.
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).
To select the resources it is possible to use the option \texttt{-p resource}.

The properties for this option are:

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

\textbf{smp}: Symmetric MultiProcessing node (1 TB for 64 cores at 2.67 GHz);

\textbf{visu}: visualisation node (12 cores at 2.67GHz and 2 NVIDIA Quadro 5000 card).
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 can be found at the address:

http://cbrl.up.univ-mrs.fr/~mesocentre

A list of libraries and softwares is available at:

http://cbrl.up.univ-mrs.fr/~mesocentre/software.php
More information can be found at the address:

http://cbrl.up.univ-mrs.fr/~mesocentre

A list of libraries and softwares is available at:

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

Questions?