

Note: You need an account first!
hpc.uni.lu/users/get_an_account.html

Cluster access

Tools (SSH protocol):

Windows – *Putty*
OS X/Linux – *ssh* (under Terminal or console)

Windows configuration (Putty):

- Session
- Host Name: *access-gaia.uni.lu* or *access-chaos.uni.lu*
- Port: *8022* - Connection type: *SSH*
- Connection → Data
- Auto-login username: *your-username*
- SSH → Auth
- Private key file for authentication: *your-key.ppk*

OS X/Linux configuration (ssh):

- Create or append to the *~/.ssh/config* file:


```
Host gaia-cluster
  HostName access-gaia.uni.lu
  Port 8022
  User your-username
  IdentityFile ~/.ssh/your-private-key
```
- Create a similar entry for *chaos-cluster*
- In a terminal window you can now connect with:


```
ssh gaia-cluster and ssh chaos-cluster
```

Tools (SFTP protocol):

Windows – *WinSCP* (or *FileZilla Client*)
OS X – *FileZilla Client* (*Cyberduck*; *rsync* or *scp* in Terminal)
Linux – *FileZilla Client* (*rsync* or *scp* in console)

File transfer

WinSCP configuration (Windows):

- Host name: *access-gaia.uni.lu* or *access-chaos.uni.lu* - Port: *8022*
- User name: *your-username*
Advanced → SSH → Authentication: Private key file: *your-key.ppk*

FileZilla Client configuration (OS X / Linux):

- Host: *sftp://access-gaia.uni.lu* - User: *your-username* - Port: *8022*

Rsync (OS X / Linux command line):

```
rsync -avP dir_name gaia-cluster:destination_dir
rsync -avP gaia-cluster:dir_name destination_dir
```

Storage spaces

- \$HOME – 50GB, with backup, designed for storing *results*
- \$WORK – 3TB, no backup, designed for *intermediate data*
- \$SCRATCH – 10TB, fast, no backup, **Gaia only**; **Chaos**: *local* node /tmp designed for *temporary storage of intermediate data requiring very fast processing speeds*
- \$PROJECT – users and groups can request project directories to share applications and data

OAR Batch Scheduler:

managing user *jobs* (computing tasks) that require specific *resources* and *time* which can be executed *interactively*, *passively* or by *reservation*

Job management

Basic commands:

```
oarstat – job state
oarsub – job submission
oarde1 – job deletion
```

Job state:

```
oarstat -u – status of own jobs
oarstat -u -f – full status of own jobs
oarstat -f -j JOBID – full status of job JOBID
```

Job submission:

```
oarsub -I – request an (I)nteractive job: the user is connected to the console of first node when resources are available
oarsub -S SHELLSCRIPT – request (S)cript job: the (S)cript is started on the first node when resources are available
oarsub -r "DATE TIME" – (r)eserve resources for job to start at specified time, defined in YYYY-MM-DD HH:MM:SS format
oarsub -C JOBID – (C)onnect to console of first node in specified job
```

Resource hierarchy and properties (to be used with oarsub):

For either interactive, passive or reservation jobs, a **(l)ist** of resources can be set:

```
-l core=1,walltime=2:00:00 – request 1 core for 2 hours (default)
-l core=4,walltime=10:0:0 – request 4 cores for 10 hours, cores may be scheduled on different nodes
-l nodes=1/core=4,walltime=2:0:0 – request 4 cores for 2h on the same node
-l nodes=5/core=4,walltime=0:5:0 – request 20 cores: 4 cores on each of 5 nodes, for 5 minutes
```

Additionally, **(p)roperties** (see FAQ) can be specified to direct the job allocation:

```
-p "gpu='YES'" – request that the listed resources have GPU cards (Nvidia Tesla on Gaia)
-p "nodeclass='e'" – request that the listed resources are in the 'e-cluster' (on Chaos)
```

Job deletion:

```
oarde1 JOBID – remove job from execution (user tasks will be terminated) or from the waiting queue
```

Job monitoring

Web-based monitoring tools:

Monika – current cluster state (active/waiting jobs)
Drawgantt – Gantt (timeline view) of scheduled jobs
Ganglia – Compute node details (cpu, memory, network usage, etc.) historical and live

Links:

hpc.uni.lu/gaia/monika hpc.uni.lu/chaos/monika
hpc.uni.lu/gaia/drawgantt hpc.uni.lu/chaos/drawgantt
hpc.uni.lu/gaia/ganglia hpc.uni.lu/chaos/ganglia

Software environment

Environment Modules / Lmod:

managing application profiles and dependencies

Usage:

```
module available – list available applications
module load APP – load application profile
module load APP/VER – load a specific version
module list – list loaded applications
module unload APP – unload application
module purge – unload all currently loaded applications
```

Search for software '*simsoft*' (any version):
module av 2>&1 | grep -i simsoft

Quick FAQ

Q: Where can I find documentation for the platform?
A: All the details of the UL HPC are on: hpc.uni.lu.

Q: I have a question, I need help - what do I do?
A: First search for the answer on hpc.uni.lu, then send a mail to hpc-users@uni.lu to ask the other users, and finally write to hpc-sysadmins@uni.lu.

Q: What are the computing resources I can access?
A: Two main separate clusters: *Gaia* and *Chaos*.

Q: What properties can I specify to select specific computing resources?
A: There are properties for CPU type and speed, cores per CPU, amount of memory, etc. which can be seen in the *Monika* job monitoring tool (see above).

Q: Is there a quick reference for Linux commands?
A: Yes: tinyurl.com/unixquickref-pdf

Examples

- E1.** Transfer data to Gaia, start a passive job for 10 hours on 24 cores over 2 computing nodes and check its status:
(*workstation*)\$ *rsync -avP ~/data gaia-cluster:toProcess/*
(*workstation*)\$ *ssh gaia-cluster*
(*access-gaia*)\$ *cd toProcess/data*
(*access-gaia*)\$ *oarsub -l nodes=2/core=12,walltime=10:0:0 -S ./process.sh*
OAR_JOB_ID=123456
(*access-gaia*)\$ *oarstat -f -j 123456*
- E2.** Reserve four complete nodes over the Easter 2015 weekend (Friday evening to Monday morning) in the s-cluster of Chaos:
(*access-chaos*)\$ *oarsub -r "2015-04-03 19:00:00"*
-l nodes=4,walltime=61:00:00 -p "nodeclass='s'"
OAR_JOB_ID=789012
- E3.** Connect to the (E2) job on Friday at 19:01 and launch a data processing script:
(*access-chaos*) \$ *oarsub -C 789012*
(*s-cluster1-10*)\$ *scriptsdir/process.sh &*

• Main site: hpc.uni.lu Follow us!
• Twitter : [@ULHPC](https://twitter.com/ULHPC)
• Github : github.com/ULHPC

