# Infrastructure and Resources

- Curnagl
- Curnagl - 2022
- Storage on Curnagl
- Jura
- Urblauna

# Curnagl

# Kesako?

Curnagl (Romanche), or Chocard à bec jaune in French, is a sociable bird known for its acrobatic exploits and is found throughout the alpine region. More information is available here

It's also the name of the HPC cluster managed by the DCSR for the UNIL research community.

A concise description if you need to describe the cluster is:

> Curnagl is a 96 node HPC cluster based on AMD Zen2/3 CPUs providing a total of 4608 compute cores and 54TB of memory. 8 machines are equipped with 2 A100 GPUs and all nodes have 100Gb/s HDR Infiniband and 100Gb/s Ethernet network connections in a fat-tree topology. The principal storage is a 2PB disk backed filesystem and a 150TB SSD based scratch system. Additionally all nodes have 1.6 TB local NVMe drives.

If you experience unexpected behaviour or need assistance please contact us via helpdesk@unil.ch starting the mail subject with DCSR Curnagl.

# How to connect

For full details on how to connect using SSH please read the documentation

Please be aware that you must be connected to the VPN if you are not on the campus network.

Then simply `ssh username@curnagl.dcsr.unil.ch` where username is your UNIL account

The login node must not be used for any form of compute or memory intensive task apart from software compilation and data transfer. Any such tasks will be killed without warning.

You can also use the cluster thourgh the OpenOnDemand interface.

# Hardware

## Compute

The cluster is composed of 96 compute nodes of which eight have GPUs.

| Number of nodes | Memory | CPU | GPU |
|---|---|---|---|
| 52 | 512 GB | 2 x AMD Epyc2 7402 | - |
| 12 | 1024 GB | 2 x AMD Epyc2 7402 | - |
| 8 | 512 GB | 2 x AMD Epyc2 7402 | 2 x NVIDIA A100 |
| 24 | 512 GB | 2 x AMD Epyc3 7443 | |

## Network

The nodes are connected with both HDR Infiniband and 100 Gb Ethernet. The Infiniband is the primary interconnect for storage and inter-node communication.

# Cluster partitions

There are 3 main partitions on the cluster:

## interactive

The interactive partition allows rapid access to resources but comes with a number of restrictions, the main ones being:

- Only one job per user at a time
- Maximum run time of 8 hours but this decreases if you ask for lots of resources.

For example:

| CPU cores requested | Memory requested | GPUs requested | Run Time Allowed |
|---|---|---|---|

| | | | |
|---|---|---|---|
| 4 | 32 | 1 | 8 hours |
| 8 | 64 | 1 | 4 hours |
| 16 | 128 | 1 | 2 hours |
| 32 | 256 | 1 | 1 hour |

We recommend that users access this using the `Sinteractive` command. This partition should also be used for compiling codes.

This partition can also be accessed using the following sbatch directive:

`#SBATCH -p interactive`

**Note on GPUs in the interactive partition**

There is one node with GPUs in the interactive partition and in order to allow multiple users to work at the same time these A100 cards have been partitioned into 2 instances each with 20GB of memory for a total of 4 GPUs.

The maximum time limit for requesting a GPU is 8 hours with the CPU and memory limits applying.

For longer jobs and to have whole A100 GPUs please submit batch jobs to the gpu partition.

Please do not block resources if you are not using them as this prevents other people from working.

If you request too many resources then you will see the following error:

`salloc: error: QOSMaxCpuMinutesPerJobLimit`
`salloc: error: Job submit/allocate failed: Job violates accounting/QOS policy (job submit limit, user's size and/or time limits)`

Please reduce either the time or the cpu / memory / gpu requested.

# cpu

This is the main partition and includes the majority of the compute nodes. Interactive jobs are not permitted. The partition is configured to prevent long running jobs from using all available resources and to allow multi-node jobs to start within a reasonable delay.

The limits are:

Normal jobs - 3 days

Short jobs - 12 hours

Normal jobs are restricted to ~2/3 of the resources which prevents the cluster being blocked by long running jobs.

In exceptional cases wall time extensions may be granted but for this you need to contact us with a justification before submitting your jobs!

The cpu partition is the default partition so there is no need to specify it but if you wish to do so then use the following sbatch directive

`#SBATCH -p cpu`

# gpu

This contains the GPU equipped nodes.

To request resources in the gpu partition please use the following sbatch directive:

`#SBATCH -p gpu`

The limits are:

Normal jobs - 3 days

Short jobs - 12 hours

Normal jobs are restricted to ~2/3 of the resources which prevents the cluster being blocked by long running jobs.

`--gres=gpu:N`

where N is 1 or 2.

# Software

For information on the DCSR software stack see the following link:

https://wiki.unil.ch/ci/books/high-performance-computing-hpc/page/dcsr-software-stack

# Storage

The storage is provided by a Lenovo DSS system and the Spectrum Scale (GPFS) parallel filesystem.

# /users

Your home space is at /users/username and there is a per user quota of 50 GB and 100,000 files.

We would like to remind you that all scripts and code should be stored in a Git repository.

# /scratch

The scratch filesystem is the primary working space for running calculations.

The scratch space runs on SSD storage and ***has an automatic cleaning policy*** so in case of a shortage of free space files older than 2 weeks (starting with the oldest first) will be deleted.

Initially this cleanup will be triggered if the space is more than 90% used and this limit will be reviewed as we gain experience with the usage patterns.

The space is per user and there are no quotas (*). Your scratch space can be found at /scratch/username

e.g. `/scratch/ulambda`

Use of this space is not charged for as it is now classed as temporary storage.

* *There is a quota of 50% of the total space per user to prevent runaway jobs wreaking havoc*

# /work

The work space is for storing data that is being actively worked on as part of a research project. Projects have quotas assigned and while we will not delete data in this space there is no backup so all critical data must also be kept on the DCSR NAS.

The structure is:

`/ work / FAC / FACULTY / INSTITUTE / PI / PROJECT`

This space can, and should, be used for the installation of any research group specific software tools including python virtual environments.

# Curnagl - 2022

Following the migration to the CCT datacenter there are a number of things that have changed that you should be aware of:

## New login node

When you first connect to curnagl.dcsr.unil.ch you will receive a warning that the host key has changed and you will  not be allowed to connect.

Please remove the old host key for curnagl.dcsr.unil.ch in your .ssh/known_hosts (ssh-keygen -R curnagl.dcsr.unil.ch) file and reconnect .

The new login node is identical to the compute nodes (it is a compute node) but as previously it should not be used for running calculations.

## New software stack

The slightly delayed 2022 DCSR software stack is now in production and includes more recent compilers as well as new versions of packages and libraries.

For more information see [https://wiki.unil.ch/ci/books/high-performance-computing-hpc/page/dcsr-software-stack](https://wiki.unil.ch/ci/books/high-performance-computing-hpc/page/dcsr-software-stack)

The old software stack remains available although no new packages will be added to it.

To switch between software stacks there is the new dcsrsoft tool:

```
# Show which stack is being used

[ulambda@curnagl ~]$ dcsrsoft show
Running with Prod

# Switch to the 2021 stack

[ulambda@curnagl ~]$ dcsrsoft use old
Switching to the old software stack
```

```
# Switch to the unsupported Vital-IT software stack

[ulambda@curnagl ~]$ dcsrsoft use vitalit
Switching to the distant past

# Switch back to the 2022 stack

[ulambda@curnagl ~]$ dcsrsoft use prod
Switching to the prod software stack
```

> The dcsrsoft command is a bash function and it should be executed on the fronted node. In order to use an old stack on a job, you need to execute the commands above before launching your job using sbatch.

## More disk space

Soon the available disk space will be doubled with 2PB available for /work

## More nodes

Once the migration is complete there will be an additional 24 compute nodes bringing the total to 96 machines of which 12 have 1TB of memory and 8 have A100 GPUs.

# Storage on Curnagl

## Where is data stored

The recommended place to store all important data is on the DCSR NAS which fulfils the UNIL requirement to have multiple copies. For more information please see the user guide

This storage is accessible from within the UNIL network using the SMB/CIFS protocol. It is also accessible on the cluster login node at /nas (see this guide)

The UNIL HPC clusters also have dedicated storage that is shared amongst the compute nodes but this is not, in general, accessible outside of the clusters  except via file transfer protocols (scp).

This space is intended for active use by projects and is not a long term store.

## Cluster filesystems

The cluster storage is based on the IBM Spectrum Scale (GFPS) parallel filesystem. There are two disk based filesystems (users and work) and one SSD based one (scratch). Whilst there is no backup the storage is reliable and resilient to disk failure.

The role of each filesystem as well as details of the data retention policy is given below.

### How much space am I using?

For the users and work filesystems the quotacheck command allows you to see the used and allocated space:

```
[ulambda@login ~]$ quotacheck
### Work Quotas ###

Project: pi_ulambda_100111-pr-g

                    Block Limits                        |     File Limits
Filesystem type         blocks      quota      limit   in_doubt    grace |    files   quota    limit in_doubt    grace
Remarks
```

```
work       FILESET     304.6G     1.999T        2T         0     none |  1107904 9990000 10000000       0     none
DCSR-DSS.dcsr.unil.ch


Project: gruyere_100666-pr-g

                    Block Limits                            |     File Limits
Filesystem type         blocks      quota      limit   in_doubt    grace |    files   quota    limit in_doubt    grace
Remarks
work       FILESET          0       99G       100G          0     none |        1  990000  1000000       0     none DCSR-
DSS.dcsr.unil.ch


### User Quota ###

                    Block Limits                            |     File Limits
Filesystem type         blocks      quota      limit   in_doubt    grace |    files   quota    limit in_doubt    grace
Remarks
users      USR         8.706G       50G       51G      160M     none |    66477  102400   103424     160     none
DCSR-DSS.dcsr.unil.ch
```

# Users

/users/<username>

This is your home directory and can be used for storing small amounts of data. The per user quota is 50 GB and 100,000 files.

There are daily snapshots kept for seven days in case of accidental file deletion. See here for more details.

# Work

/work/<path to my project>

This space is allocated per project and the quota can be increased on request by the PI as long as free space remains.

This space is not backed up but there is no over-allocation of resources so we will never ask you to remove files.

# Scratch

/scratch/<username>

The scratch space is for intermediate files and the results of computations. There is no quota and the space is not charged for. You should think of it as temporary storage for a few weeks while running calculations.

In case of limited space files will be automatically deleted to free up space. The current policy is that if the usage reaches 90% files, starting with the oldest first, will be removed until the occupancy is reduced to 70%. ***No files newer than two weeks old will be removed***.

## $TMPDIR

For certain types of calculation it can be useful to use the NVMe drive on the compute node. This has a capacity of ~400 GB and can be accessed inside a batch job by using the $TMPDIR variable.

***At the end of the job this space is automatically purged.***

# Jura

Jura is a cluster for the analysis of sensitive data and is primarily used by the CHUV.

The Jura cluster is replaced by Urblauna

# Computing ressources

- 10 compute nodes
  - cpt01: CPUs=40 Boards=1 SocketsPerBoard=4 CoresPerSocket=10 ThreadsPerCore=1 RealMemory=515712
  - cpt02: CPUs=32 Boards=1 SocketsPerBoard=4 CoresPerSocket=8 ThreadsPerCore=1 RealMemory=257754
  - cpt[03-04]: CPUs=48 Boards=1 SocketsPerBoard=2 CoresPerSocket=12 ThreadsPerCore=2 RealMemory=257680
  - cpt[05-06]: CPUs=48 Boards=1 SocketsPerBoard=2 CoresPerSocket=12 ThreadsPerCore=2 RealMemory=64156
  - cpt[07-08]: CPUs=160 Boards=1 SocketsPerBoard=4 CoresPerSocket=20 ThreadsPerCore=2 RealMemory=1031536
  - cpt09: NodeName=cpt09 CPUs=160 Boards=1 SocketsPerBoard=4 CoresPerSocket=20 ThreadsPerCore=2 RealMemory=3095999
  - cpt10: NodeName=cpt10 CPUs=160 Boards=1 SocketsPerBoard=4 CoresPerSocket=20 ThreadsPerCore=2 RealMemory=999282

- 4 nodes with Xeon PHI accelerators
  - cpt[03-04]: 82:00.0 Co-processor: Intel Corporation Xeon Phi coprocessor 31S1 (rev 11)
  - cpt[05-06]: 82:00.0 Co-processor: Intel Corporation Xeon Phi coprocessor 5100 series (rev 11)

- Login node
  - frt: CPUs=48 Boards=1 SocketsPerBoard=2 CoresPerSocket=12 ThreadsPerCore=2 RealMemory=65697804
  - 15 TB local disk space

# Storage ressources

- Fast scratch based on SSD

    - /scratch/beegfs 112 TB
    - Not purged

- Data directory
    - /data 160 TB
    - For static datasets (including reference ones (TCGA, ADNI et al)
    - Not purged

**ATTENTION /data directory is NOT BACKED UP**

- Archive with encrypted tapes
    - /archive
    - 600 TB available
    - Data are copied transparently on two tape libraries located in two different datacenters for disaster recovery

# Getting ressources on Jura

- For sensitive data only
- Organized by PI
- Use DCRS request form and specify Sensitive or Personal data
- https://requests.dcsr.unil.ch


# Accessing the infrastructure from UNIL

- Any user is expected to take a short training to get familiar with the environment, the do’s and dont’s
- Once the demand is approved, you will receive a mail with a QR-Code like

- You need an app like Google Authenticator or FreeOTP on your smartphone to scan it
- Google Authenticator:

  https://play.google.com/store/apps/details?id=com.google.android.apps.authenticator2&hl=en

  https://apps.apple.com/us/app/google-authenticator/id388497605

  FreeOTP:

  https://play.google.com/store/apps/details?id=org.fedorahosted.freeotp&hl=en

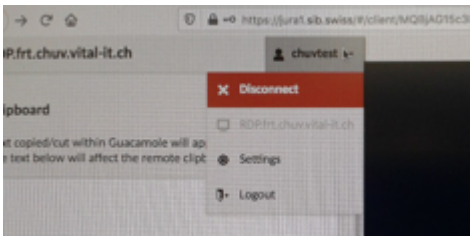
  https://apps.apple.com/us/app/freeotp-authenticator/id872559395
- Go to https://jura.dcsr.unil.ch web site and log in with your **UNIL credentials**
- 

- Enter the code displayed by the application
- 

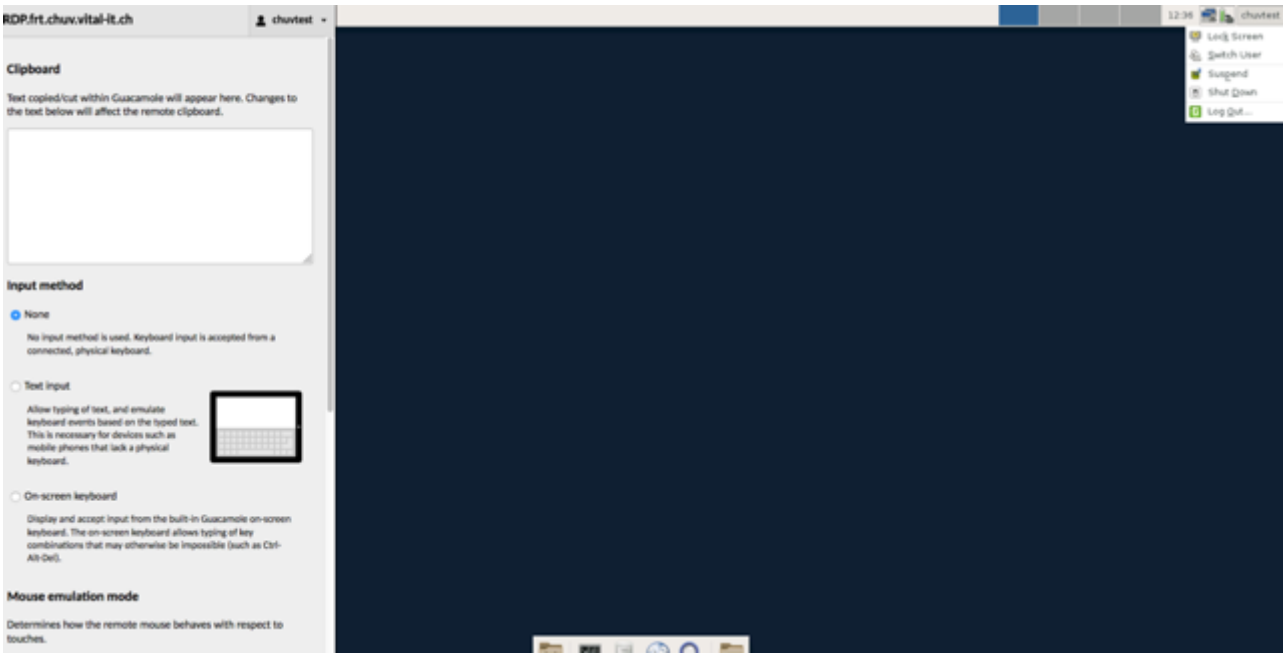
- Congratulations! you are now logged in

ATTENTION PROPER LOG OUT

- CTRL+ALT+SHIFT to display guacamole menu




- Or session logout




# Transferring data in

- Transfer your data to the Jump Host

```
sib-1-24:~ someuser$ sftp someuser@jura.dcsr.unil.ch
Password:
Verification code:
Connected to someuser@jura.dcsr.unil.ch.
sftp> dir
data
sftp> cd data
sftp> dir
sftp> put AVeryImportantFile.tgz
Uploading AVeryImportantFile.tgz to /data/AVeryImportantFile.tgz
AVeryImportantFile.tgz
```

- The verification code of the Google Authenticator or FreeOTP is required
- Transfer your data from the Jump Host

```
[someuser@frt ~]$ sftp jura.dcsr.unil.ch
Password:
Verification code:
Connected to jura.dcsr.unil.ch.
sftp> cd data
```

`sftp> dir`
`AVeryImportantFile.tgz`
`sftp> get AVeryImportantFile.tgz`
`Fetching /data/AVeryImportantFile.tgz to AVeryImportantFile.tgz`
`/data/AVeryImportantFile.tgz`

- To repeatedly transfer large files from reputable external sources a direct access can be granted.
- The verification code of the Google Authenticator or FreeOTP is required but if you have many files to transfer we can set up an automated system

# Transferring code in/out

There is a DCSR managed Git service accessible from Jura. More information can be found at

https://wiki.unil.ch/ci/books/service-de-calcul-haute-performance-%28hpc%29/page/why-is-there-a-dcsr-gitlab-service-and-what-is-it

# Accessing the infrastructure from CHUV

`ssh<unil-username>@stockage-horus.chuv.ch`

# Urblauna

## Kesako?

Urblauna (Romanche), or Lagopède Alpin in French, is a bird known for its changing plumage which functions as a very effective camouflage. More information is available at

https://www.vogelwarte.ch/fr/oiseaux/les-oiseaux-de-suisse/lagopede-alpin

It's also the name of our new sensitive data compute cluster which will replace the Jura cluster.

Information on how to connect to Urblauna can be found here.

Information on the Jura to Urblauna migration can be found here

The differences between Jura and Urblauna are described here

## Hardware

### Compute

The cluster is composed of 18 compute nodes of which two have GPUs. All have the same 24 core processor.

| Number of nodes | Memory | CPU | GPU |
|---|---|---|---|
| 16 | 1024 GB | 2 x AMD Epyc3 7443 | - |
| 2 | 1024 GB | 2 x AMD Epyc3 7443 | 2 x NVIDIA A100 |

The GPUs are partitioned to create 4 GPUs on each machine with 20GB of memory per GPU

### Storage

The storage is based on IBM Spectrum Scale / Lenovo DSS and provides 1PB of space in the /data filesystem.

Whilst reliable this space is not backed up and all important data should also be stored on /archive

The Curnagl /work filesystem is visible in read-only mode on Urblauna and can be used to install software on an internet connected system before using it on Urblauna.

| **Filesystem mount point** | **Description** |
|---|---|
| /users | Urblauna home directory |
| /scratch | Urblauna scratch space (automatic cleanup) |
| /data | Urblauna data space (no backup) |
| /archive | Secure data space with backup (login node access only) |
| /work | Curnagl data space (read only) |
| /jura_home | Jura home directories (read only, login node only) |
| /jura_data | Jura data space (read only, login node only) |

# Software

For information on the DCSR software stack see the following link:

https://wiki.unil.ch/ci/books/high-performance-computing-hpc/page/dcsr-software-stack

# Slurm partitions

On Urblauna there are two partitions - "urblauna" and "interactive"

```
$ sinfo

PARTITION   AVAIL  TIMELIMIT  NODES  STATE NODELIST
urblauna*      up 3-00:00:00     17   idle sna[002-016],snagpu[001-002]
interactive    up    8:00:00      4   idle sna[015-016],snagpu[001-002]
```

There is no separate GPU partition so to use a GPU simply request

#SBATCH --gres=gpu:1

To launch an interactive session you can use `Sinteractive` as on Curnagl