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 at

https://www.vogelwarte.ch/fr/oiseaux/les-oiseaux-de-suisse/chocard-a-bec-jaune

It's also the name of our new compute cluster which will replace the Wally and Axiom clusters. See the end of the page for an overview of the migration process.

Further documentation will be provided in the coming weeks.

As the configuration is being actively worked on some details may change without warning.

For testing and interventions the cluster may be unavailable for certain periods without warning - we expect to pass into "production" running during May at which point the service can be considered stable. In particular a number of network changes are foreseen in order to connect the cluster to the datacenter firewall infrastructure.

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

The login node is **curnagl.dcsr.unil.ch**

Before connecting please add the host's key to your list of known hosts:

echo "curnagl.dcsr.unil.ch ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBCunvgFAN/X/8b1FEIxy8p3u9jgfF0NgCl7CX4ZmqlhaYis2p7AQ34foIXemaw2wT+Pq1V9dCUh18mWXnDsjGrg=" >> ~/.ssh/known_hosts

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 (same as for wally/axiom)

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.

Hardware

Compute

The cluster is composed of 72 compute nodes of which eight have GPUs. All have the same 24 core processor.

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

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.

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.

Please note that it is up to the user to copy over data from Axiom/Wally to the new system as explained in migration section.

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

Initially there is no backup of this space - this will be put in place over the coming weeks.

/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 the same as on the Axiom and Wally scratch spaces:

`/ 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.

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	-	8 hours
8	64	-	4 hours
16	128	1	2 hours
32	256	2	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 single GPU is 2 hours.

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 - 1 day

Short jobs - 6 hours

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

These limits will be reviewed during the initial testing period. To request the number of GPUs per node please use:

```
--gres=gpu:N
```

where N is 1 or 2.

Software

The DCSR software stack is now loaded by default so when you connect you will see the following:

```
$ module avail

 /dcsrsoft/spack/hetre/v1.1/spack/share/spack/lmod/Zen2-IB-test/linux-rhel8-x86_64/Core
   cmake/3.20.0    gcc/9.3.0     mpfr/3.1.6
   cuda/11.2.2     git/2.31.0    xz/5.2.5
```

```
Use "module spider" to find all possible modules and extensions.
Use "module keyword key1 key2 ..." to search for all possible modules matching
any of the "keys".
```

To see more packages load a compiler (gcc) - note that the Intel compilers will be available in the near future

```
$ module load gcc

$ module avail

 /dcsrsoft/spack/hetre/v1.1/spack/share/spack/lmod/Zen2-IB-test/linux-rhel8-x86_64/gcc/9.3.0
   admixtools/7.0.1         gsl/2.6                      python/2.7.18
   bamaddrg/0.1             htslib/1.10.2                python/3.8.8       (D)
   bamtools/2.5.1           intel-tbb/2020.3             qtltools/1.3.1
   bcftools/1.10.2          julia/1.6.0                  r/4.0.4
   bedtools2/2.29.2         maven/3.6.3                  rsem/1.3.1
   blast-plus/2.11.0        miniconda3/4.9.2             star/2.7.6a
   bowtie2/2.4.2            mvapich2/2.3.5               stream/5.10-openmp
   cmake/3.20.0      (D)    nlopt/2.6.1                  stream/5.10        (D)
   eigen/3.3.9              octave/6.2.0                 tskit/0.3.1
   fftw/3.3.9               openblas/0.3.14-openmp       xz/5.2.5           (D)
   gdb/10.1                 openblas/0.3.14        (D)   zlib/1.2.11
   gmsh/4.7.1-openmp        openjdk/11.0.8_10
   gnuplot/5.2.8            perl/5.32.1

- /dcsrsoft/spack/hetre/v1.1/spack/share/spack/lmod/Zen2-IB-test/linux-rhel8-x86_64/Core --
   cmake/3.20.0    cuda/11.2.2    gcc/9.3.0 (L)    git/2.31.0    mpfr/3.1.6    xz/5.2.5

  Where:
   D:  Default Module
   L:  Module is loaded

Use "module spider" to find all possible modules and extensions.
Use "module keyword key1 key2 ..." to search for all possible modules matching any of the
"keys".
```

The provided versions for key tools are:

- GCC - 9.3.0

- Python - 3.8.8
- R - 4.0.4
- MPI - mvapich2 2.3.5

Not all tools currently installed on the Wally/Axiom stack are currently available as we are proceeding with updates to new versions but more will be available in the coming weeks.

The old (Vital-IT) /software stack will be made available but is unsupported and there is no guarantee that it will work correctly.

Further information

Migration from Axiom and Wally

The /work space will be accessible from the Axiom and Wally login nodes which can be used to transfer important data from the /scratch/wally and /scratch/axiom spaces to /work

Note: initially this space will only be visible from wally-front1

On Curnagl your home directories on Axiom and Wally will be available (read only) under /oldusers/username - e.g. `/oldusers/ulambda`

It is your responsibility to copy data and please take the opportunity to clean and organise things.

Axiom and Wally decommissioning

At the very latest the systems and storage will remain available until the end of 2021.

Axiom

Job submission to Axiom will be stopped at the end of April and the nodes powered off.

The Axiom scratch space will remain visible on the login nodes until later in the year.

Wally

Job submission to Wally will continue until later in the year but no hardware problems will be resolved.

The new software stack will be made available but without MPI codes as the interconnect does not work correctly with recent MPI implementations.

Revision #34
Created 9 February 2021 06:58:15 by Ewan Roche
Updated 26 April 2021 12:57:05 by Ewan Roche