Efficient data storage on the CECI clusters

Ariel Lozano

CÉCI HPC Training 2022/3

DISCLOSURE

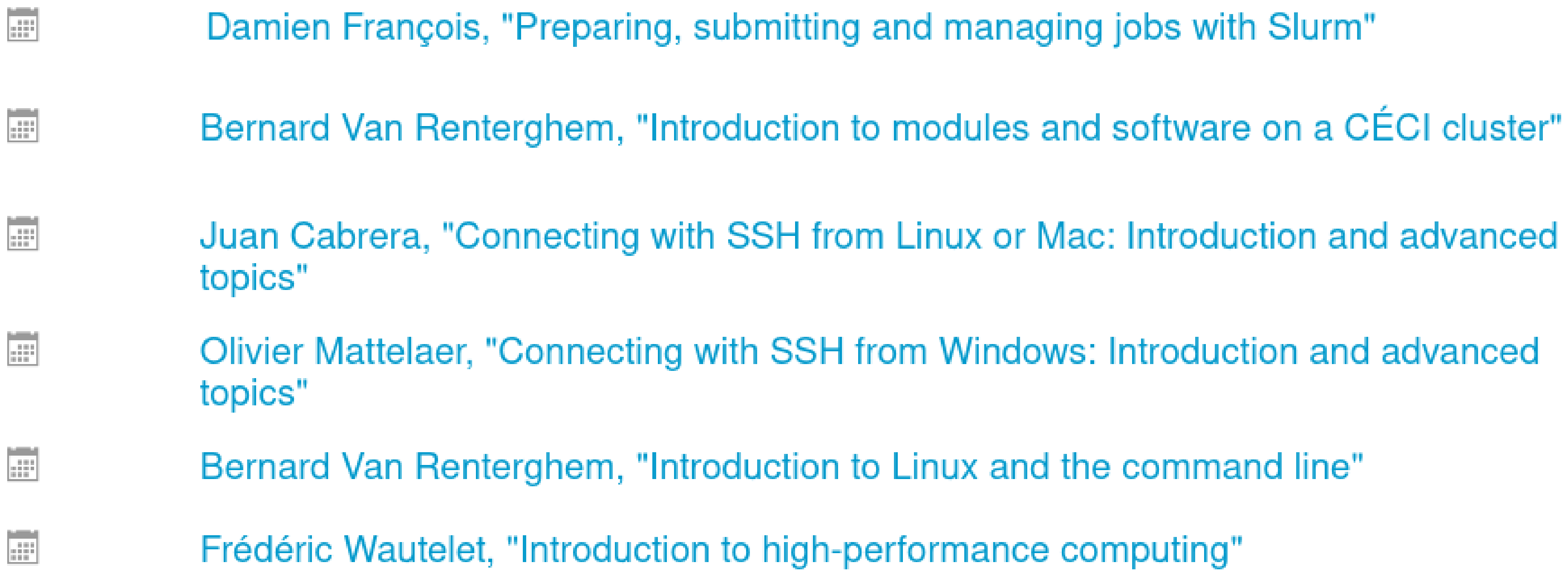
WARNING: No data on the CECI clusters has backups

You are responsible of copying over your useful data you need to store long term somewhere else

Some of the CECI universities provide solutions see:
https://support.ceci-hpc.be/doc/_contents/ManagingFiles/LongtermStorage.html

Prereqs

- To follow properly this presentation you must be already familiar with:



- Damien François, "Preparing, submitting and managing jobs with Slurm"
- Bernard Van Renterghem, "Introduction to modules and software on a CÉCI cluster"
- Juan Cabrera, "Connecting with SSH from Linux or Mac: Introduction and advanced topics"
- Olivier Mattelaer, "Connecting with SSH from Windows: Introduction and advanced topics"
- Bernard Van Renterghem, "Introduction to Linux and the command line"
- Frédéric Wautelet, "Introduction to high-performance computing"

Some context

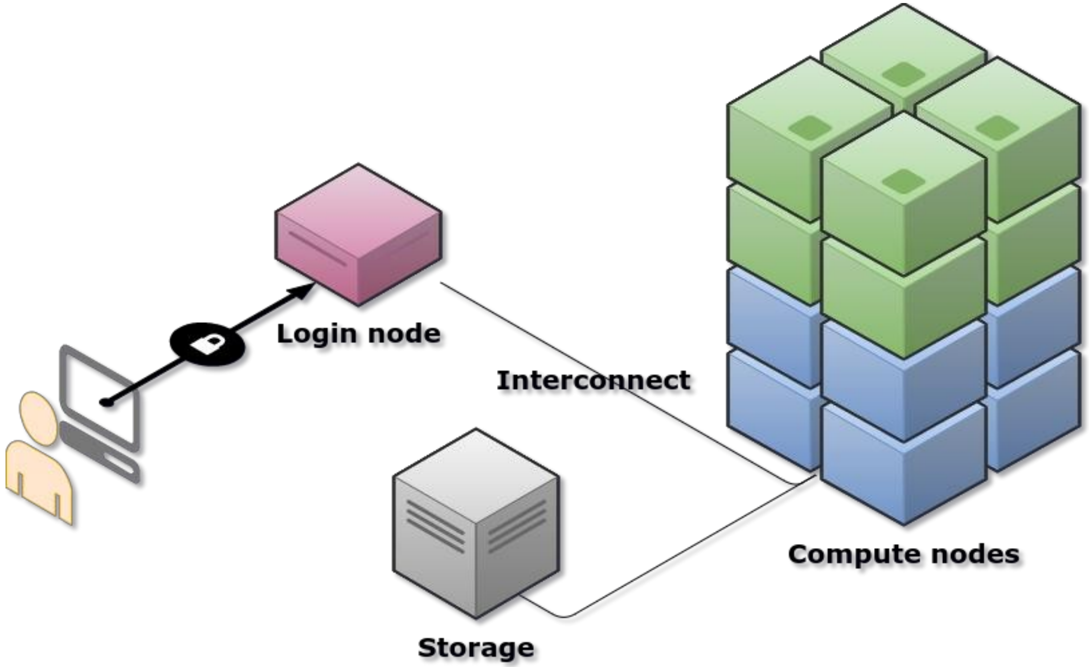
- Nowadays the best performant **units** of long term storage provides ~2 GB/s of sequential read/write. This goes down to about ~400MB/s for random read/write of many small files.

- Basic sequential write test on a laptop with a consumer NVMe SSD: *2TB Intel SSD 660P Series*

```
$ dd if=/dev/zero of=test2GBdump bs=1M count=2048; sync
2048+0 records in
2048+0 records out
2147483648 bytes (2.1 GB, 2.0 GiB) copied, 0.842955 s, 2.5 GB/s
```

- Basic test with a single task writing on the storage. The CPU access the SSD directly via PCI express lanes.

Previous: HPC cluster

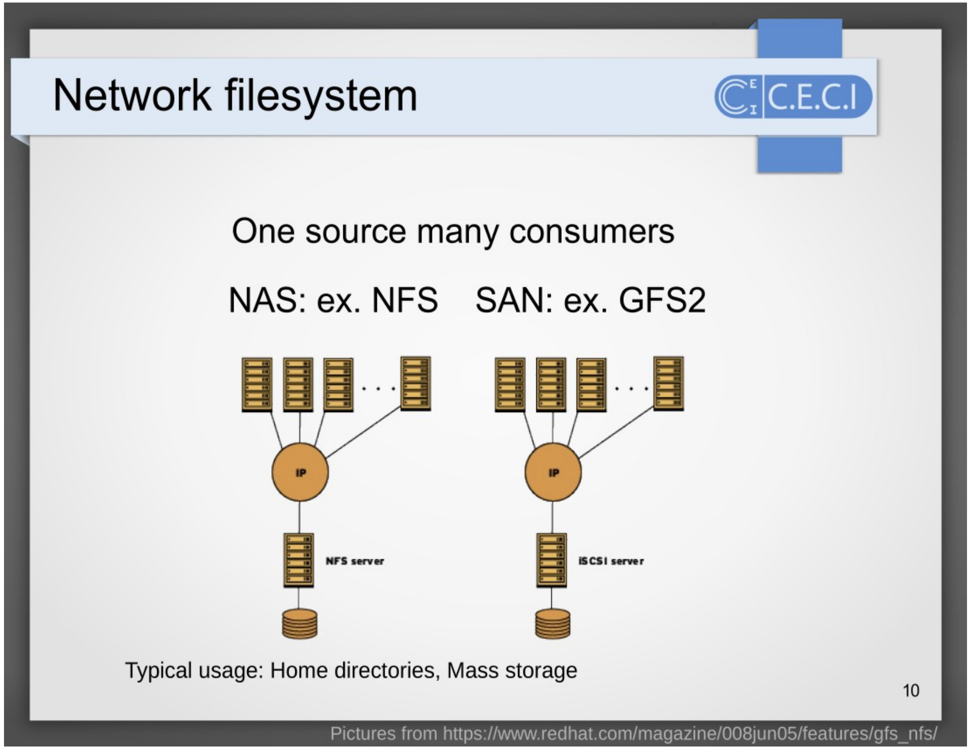


"Introduction to high-performance computing" (Frédéric Wautelet)

- A computer ‘cluster’ is a group of **linked** computers working together closely, so that in many respects they form a single computer

- Corollary: Access to **most** of the different storage solutions available on these systems happens via the network

Previous: Network storage solutions



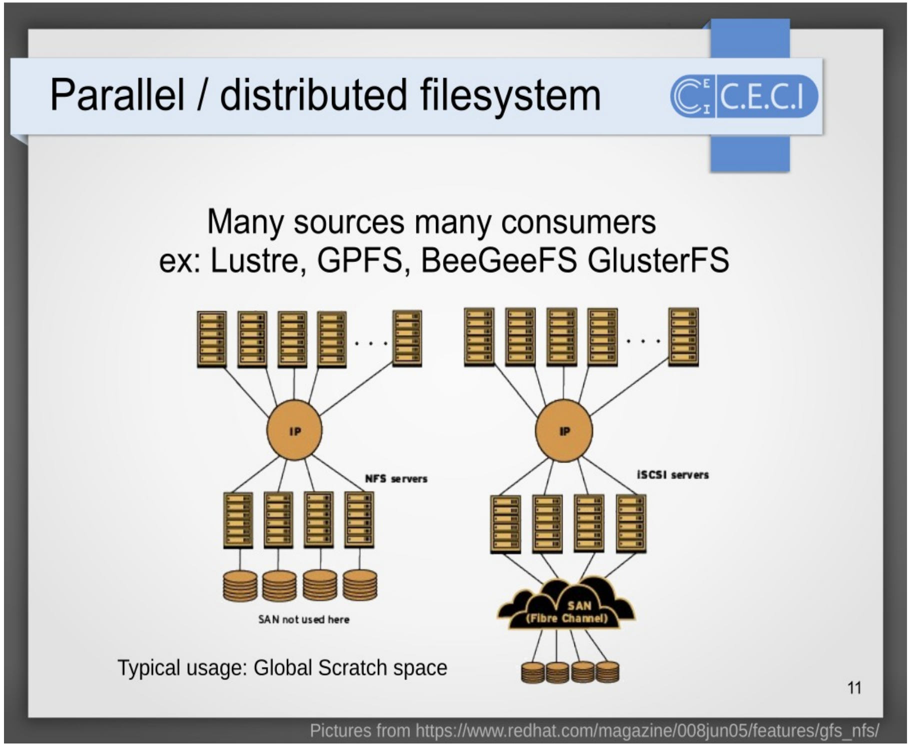
Network filesystem

One source many consumers

NAS: ex. NFS   SAN: ex. GFS2

Typical usage: Home directories, Mass storage

Pictures from https://www.redhat.com/magazine/008jun05/features/gfs_nfs/



Parallel / distributed filesystem

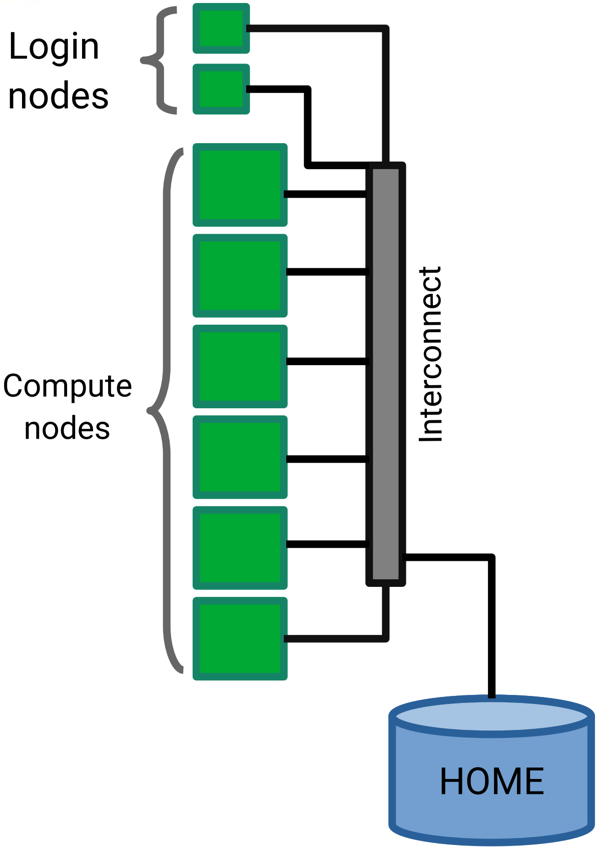
Many sources many consumers
ex: Lustre, GPFS, BeeGeeFS GlusterFS

Typical usage: Global Scratch space

Pictures from https://www.redhat.com/magazine/008jun05/features/gfs_nfs/

Damien François, "Introduction to data storage and access"

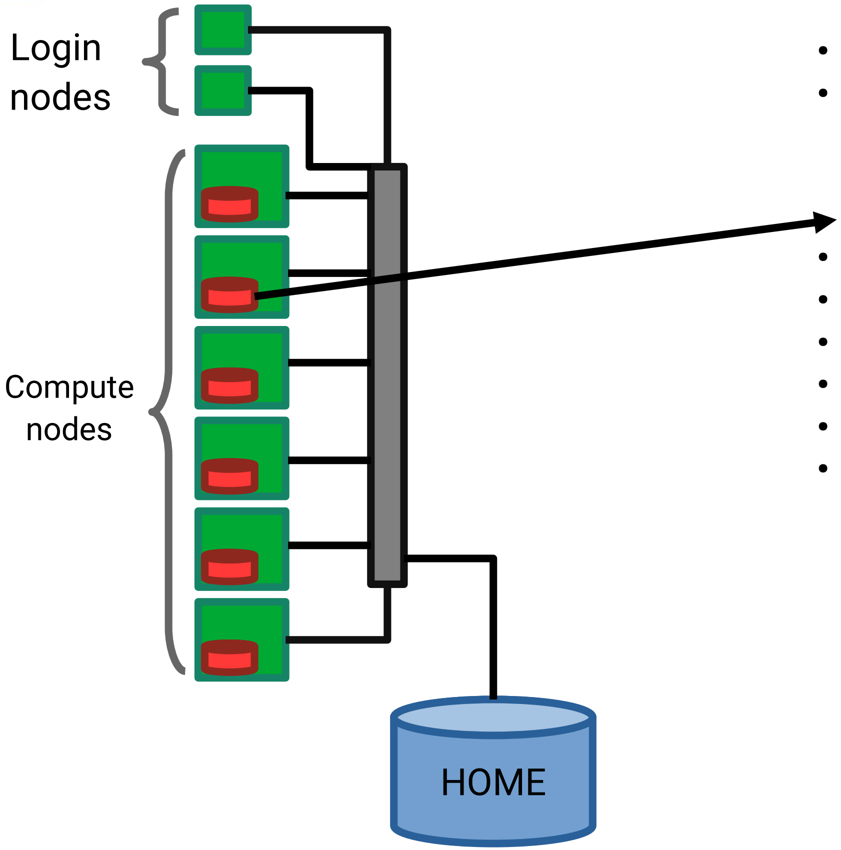
Storages on CECI clusters



$HOME

- Area where you land on login
- Network storage visible from all compute nodes and login

Storages on CECI clusters



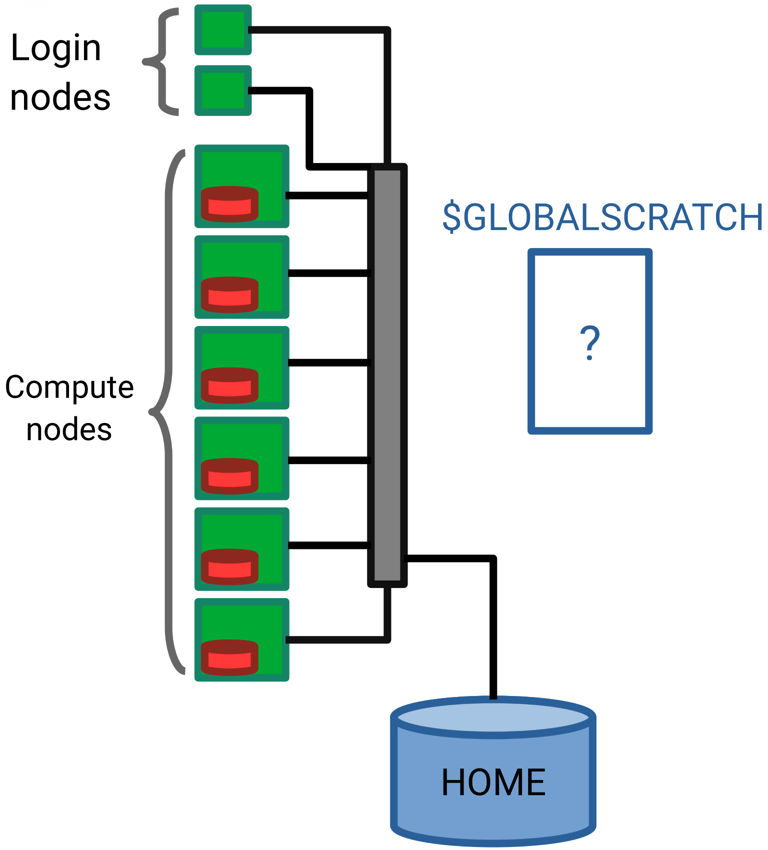
$HOME

- Area where you land on login
- Network storage visible from all compute nodes and login

$LOCALSCRATCH

- Internal storage inside each compute node (HDD or SSD)
- Can provide high performance via a RAID0 setup
- Accessible **only** by the compute node hosting it
- Shared with other jobs running on the same node
- Usage only temporary during the job runtime
- Serial jobs or Parallel SMP (openmp, python multiprocessing)

Storages on CECI clusters



$HOME

- Area where you land on login
- Network storage visible from all compute nodes and login

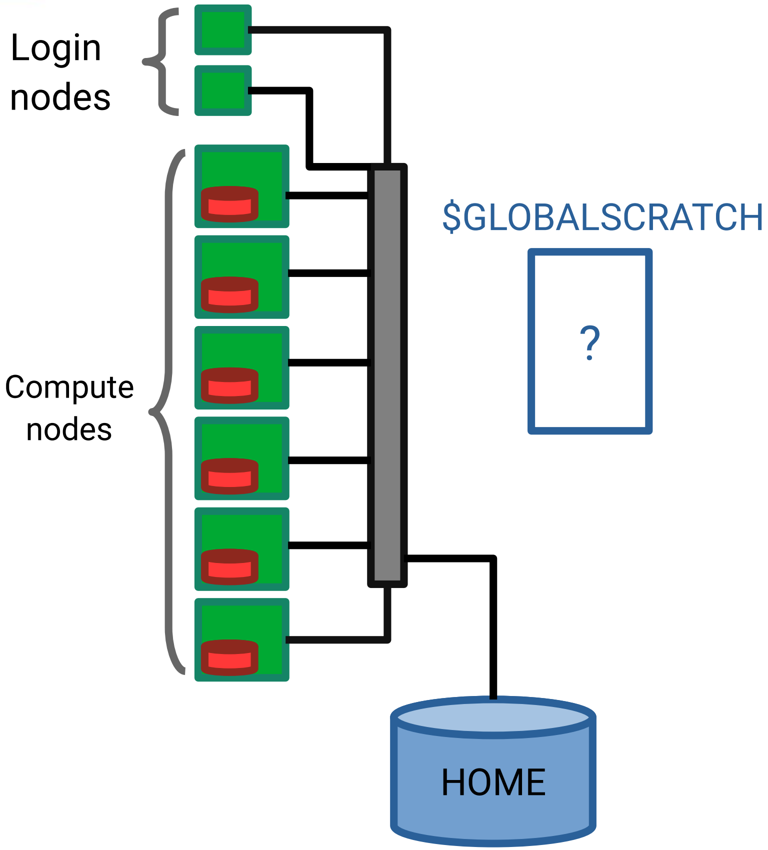
$LOCALSCRATCH

- Internal storage inside each compute node (HDD or SSD)
- Can provide high performance via a RAID0 setup
- Accessible **only** by the compute node hosting it
- Shared with other jobs running on the same node
- Usage only temporary during the job runtime
- Serial jobs or Parallel SMP (openmp, python multiprocessing)

$GLOBALSCRATCH

- Implemented via different setups
- Accessible by all compute nodes and login
- Accessible via a network interconnet
- Can be composed of a single or multiple storage sources
- Data there stays persistently (but all is removed in yearly maintenances)
- You must clenaup from time to time
- All jobs but **only option** for multinode-parallel jobs (big MPI jobs)

Storages on CECI clusters



How do we access these storage areas ?

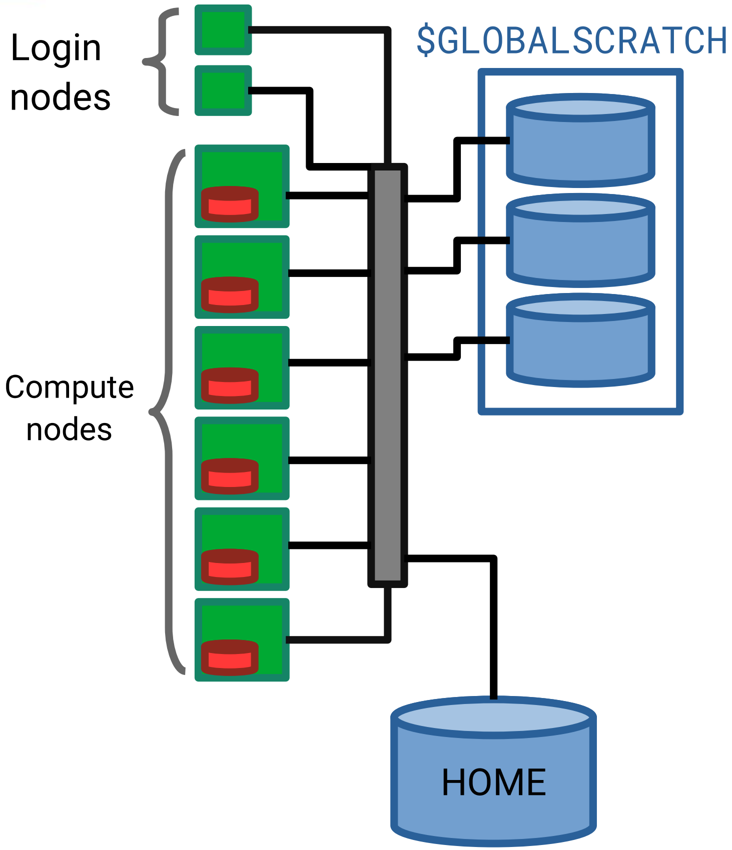
There are environment variables defined on the clusters pointing to them

- `$HOME`
- `$LOCALSCRATCH`
- `$GLOBALSCRATCH`

For LOCALSCRATCH as it’s internal to each node, it can be accessed only by jobs submitted to a given node

Lemaitre3 and NIC5

Dedicated global parallel filesystem



$HOME

- 100GB quota

$LOCALSCRATCH

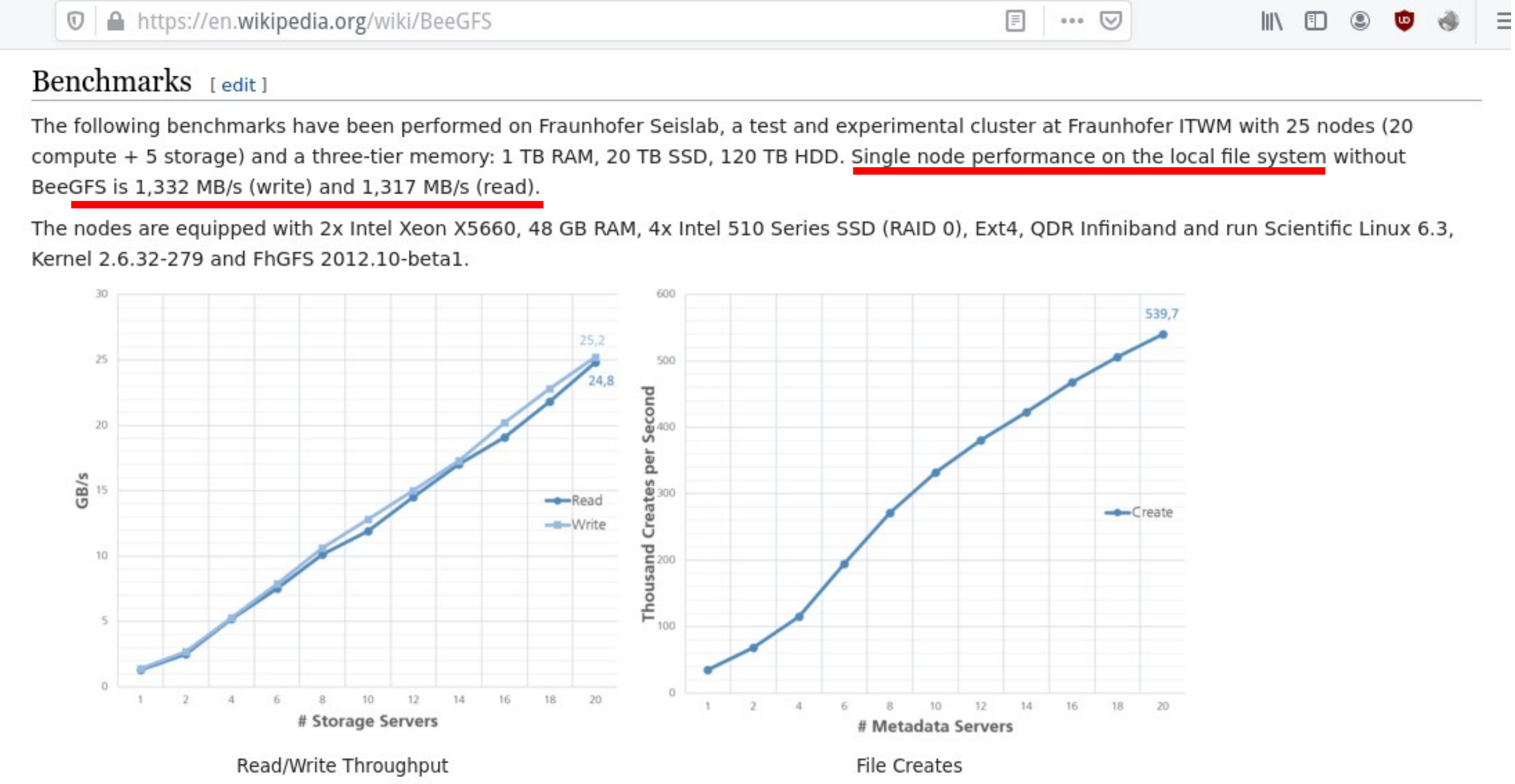
- Single SSD
- **lemaitre3: 200GB**, **NIC5: 370GB**
- Data removed when job finished!

$GLOBALSCRATCH

- Parallel filesystem distributed among multiple storage servers (BeeGFS)
- Accessible via multiples high speed network interconnet (100Gb/s)
- Visible as one single volume from login/compute nodes
- Full net size: **lemaitre3: 415TB**, **NIC5: 520TB**
- No quotas on lemaitre3, 5TB quota on NIC5 (remember to cleanup)
- The storage can be fully purged on yearly maintenances

Lemaitre3 and NIC5

Dedicated global parallel filesystem



Benchmarks [edit]

The following benchmarks have been performed on Fraunhofer Seislab, a test and experimental cluster at Fraunhofer ITWM with 25 nodes (20 compute + 5 storage) and a three-tier memory: 1 TB RAM, 20 TB SSD, 120 TB HDD. Single node performance on the local file system without BeeGFS is 1,332 MB/s (write) and 1,317 MB/s (read).

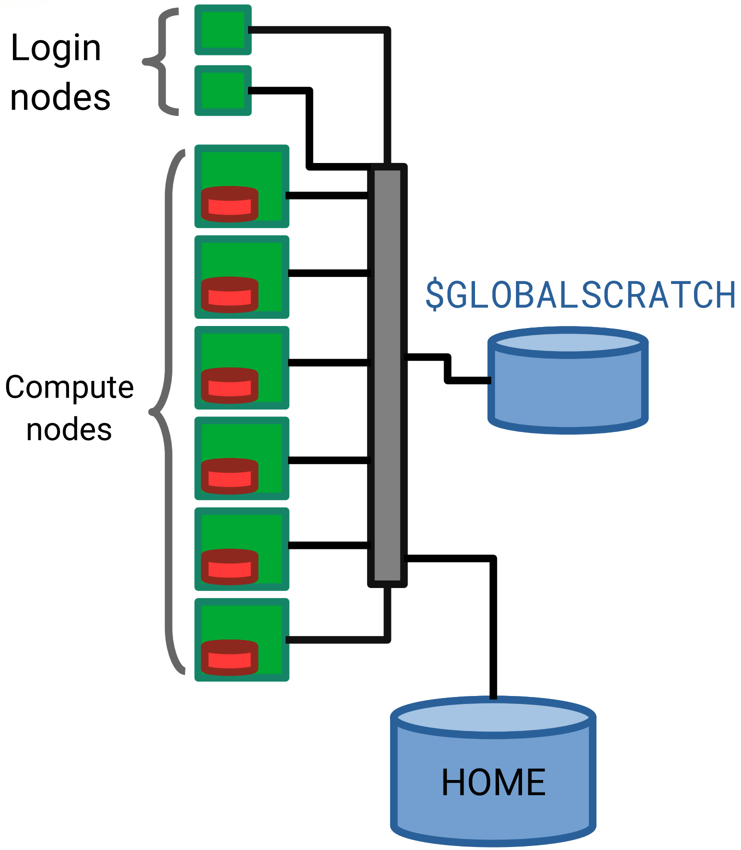
The nodes are equipped with 2x Intel Xeon X5660, 48 GB RAM, 4x Intel 510 Series SSD (RAID 0), Ext4, QDR Infiniband and run Scientific Linux 6.3, Kernel 2.6.32-279 and FhGFS 2012.10-beta1.

Read/Write Throughput

File Creates

https://www.beegfs.io/c/resources/
https://indico.mathrice.fr/event/5/session/5/contribution/12/material/slides/0.pdf

Hercules



$HOME

- 200GB quota

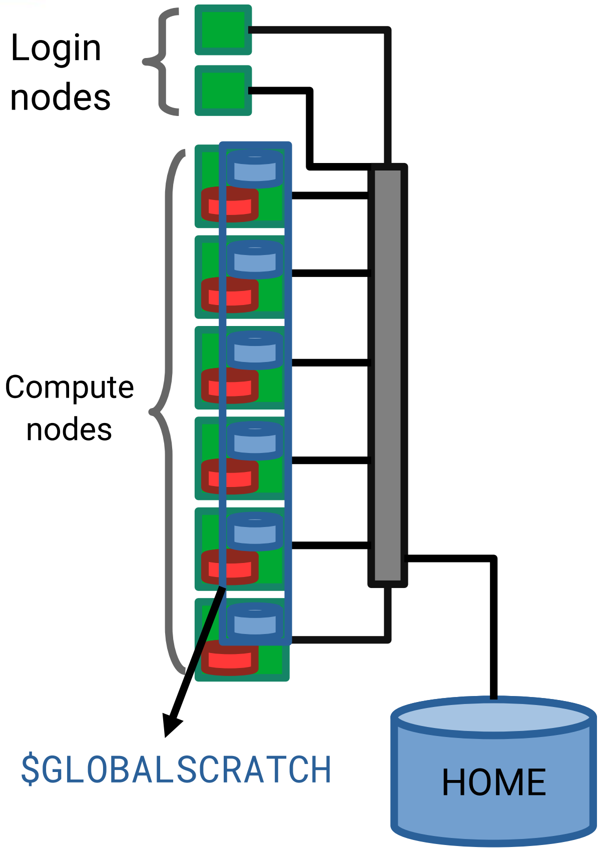
$LOCALSCRATCH

- RAID0 of 4 HDDs
- her2-w065...096: **1TB** (features=intel)
- her2-w099...126: **4TB** (features=amd)
- her2-w127...128: **8TB** (only nodes with 2TB RAM)
- Data deleted when job finished!

$GLOBALSCRATCH

- Single storage server mounted by a NFS share
- Accessible via a single network link (10Gb/s)
- 400GB soft 4TB hard quota

Dragon2



$HOME

- 40GB quota

$LOCALSCRATCH

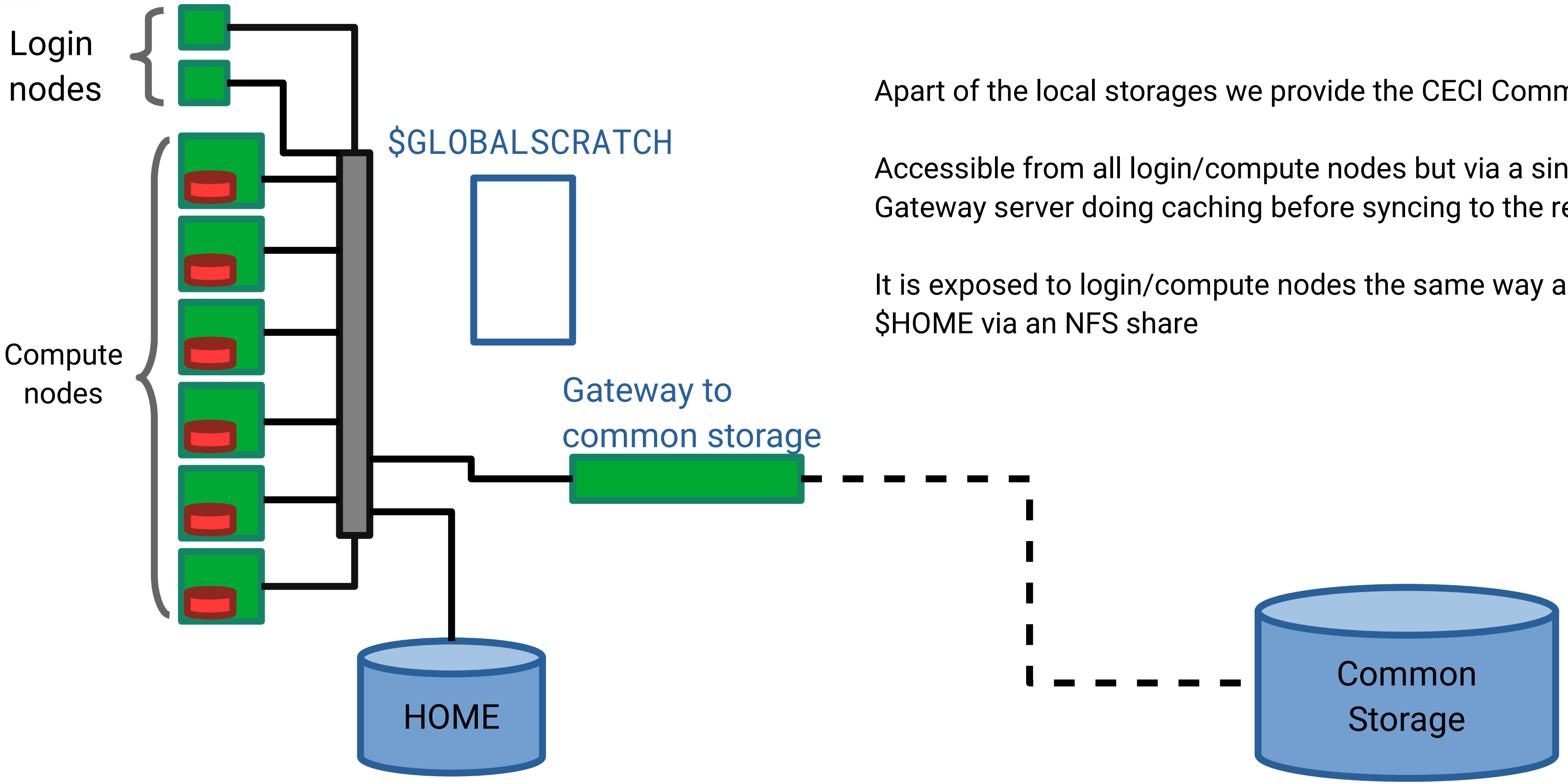
- Raid0 of 3 HDDs
- 3TB maximum capacity
- Data deleted when job finished!

$GLOBALSCRATCH

- Parallel filesystem distributed among multiple storage targets (BeeGFS)
- A partition on each compute node is part to build the scratch
- Visible as one single volume from login/compute nodes
- 52 TB size in total
- Accessible via the same network interconnet as the nodes (10Gb/s)
- No hard quotas enforced (remember to cleanup)

CECI Common storage

external remote storage accesible by all clusters



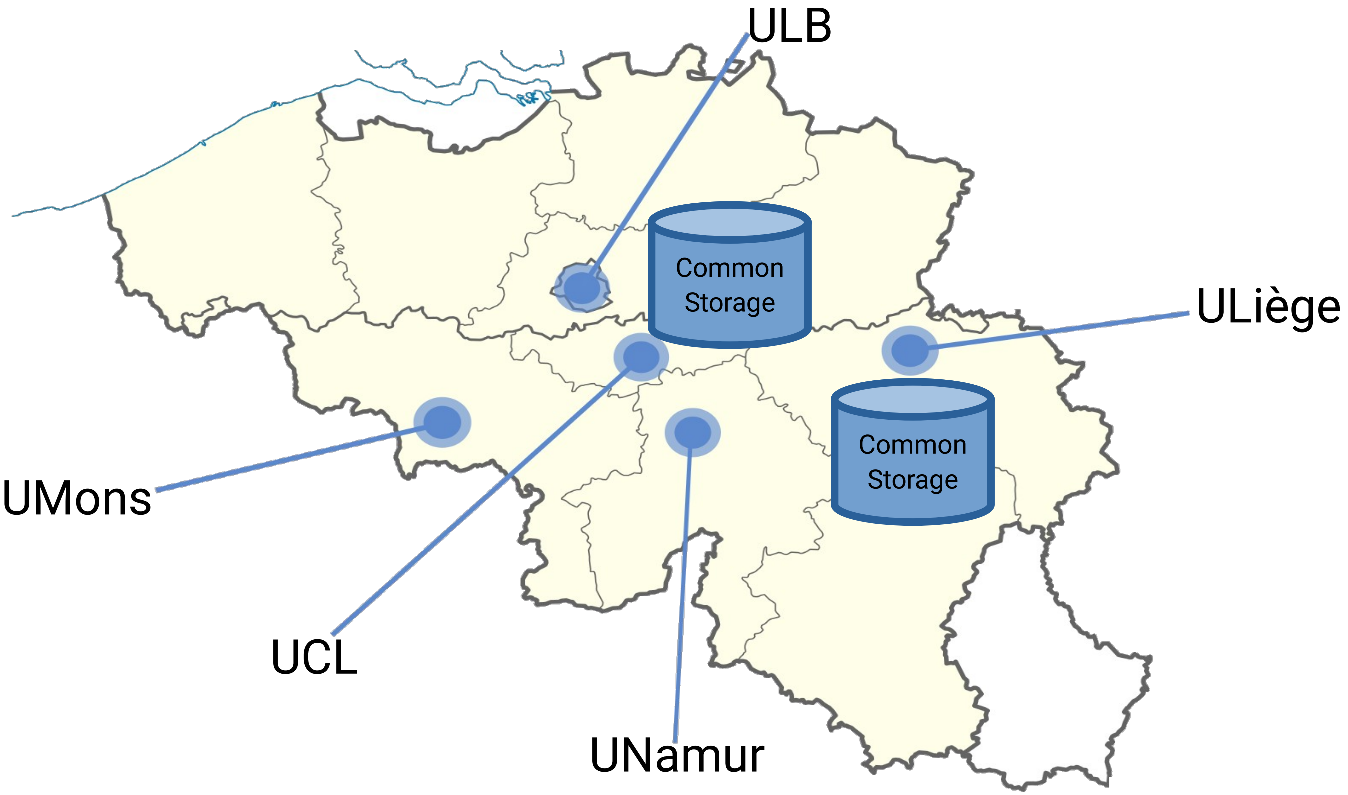
Apart of the local storages we provide the CECI Common Storage

Accessible from all login/compute nodes but via a single network link
Gateway server doing caching before syncing to the remote storage

It is exposed to login/compute nodes the same way as the $HOME via an NFS share

CECI Common storage

external remote storage accesible by all clusters



The main storage servers are in ULiège and UCL

There is a dedicated fiber among the 5 sites for this solution

CECI Common storage

external remote storage accesible by all clusters

`/CECI/home`
- Each user gets a personal area here by default
- Full personal path is pointed with `$CECIHOME` variable from any cluster
- Quota of 100GB

`/CECI/proj`
- Area where a team with a project can get a common folder for sharing data
- Must be requested by a PI
- Quota decided according to the project’s needs

`/CECI/trsf`
- Area to be used to move big amounts of data between clusters
- Common area pointed with `$CECITRSF` (create your own subfolder)
- Meant only for **temporary** copying from one cluster to another
- Data here can be purged every 6 months
- Quota of 1TB soft 10TB hard

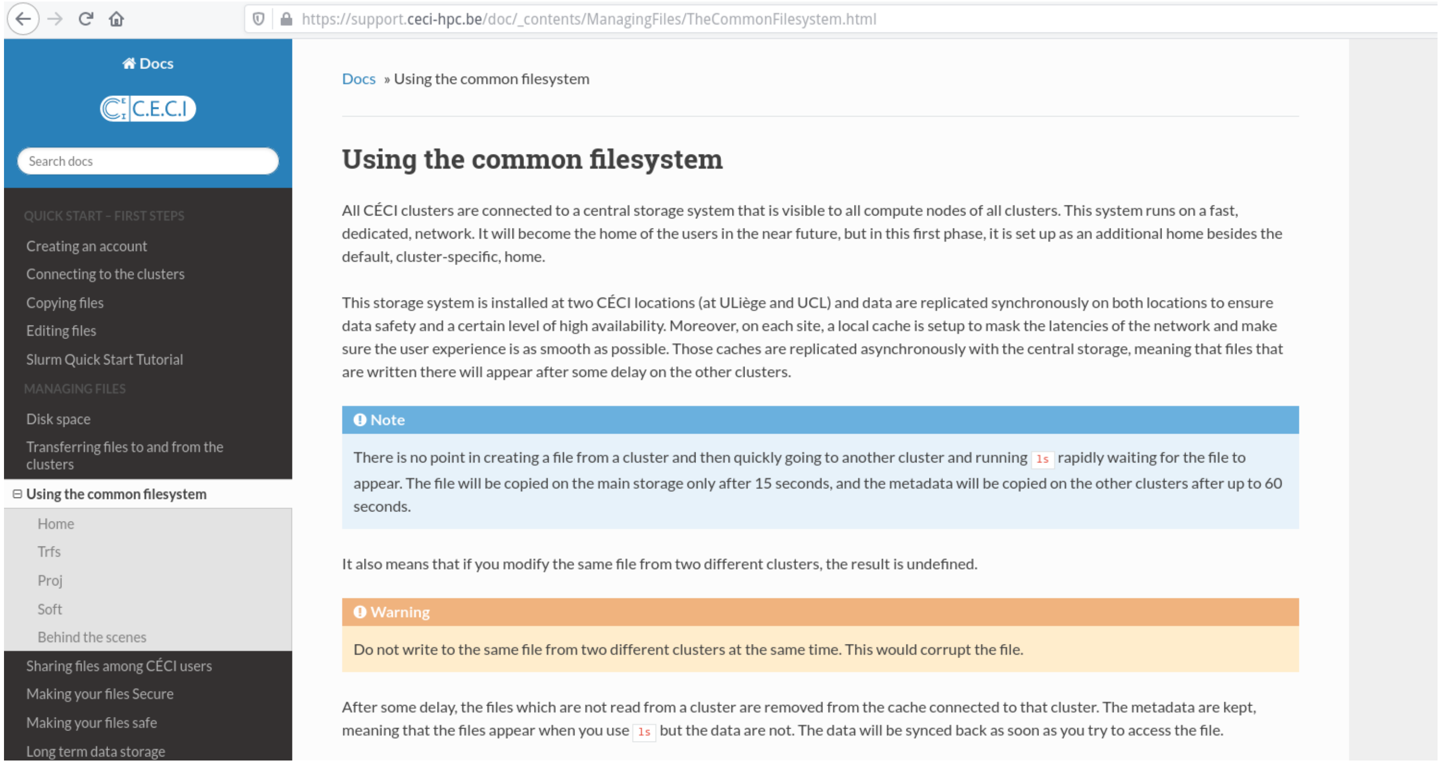
`/CECI/soft`

Used only by the sysadmins for software installations

CECI Common storage

external remote storage accesible by all clusters

For more details check our detailed documentation
https://support.ceci-hpc.be/doc/_contents/ManagingFiles/TheCommonFilesystem.html


https://support.ceci-hpc.be/doc/_contents/ManagingFiles/TheCommonFilesystem.html
Docs
C.E.C.I
Search docs
QUICK START - FIRST STEPS
Creating an account
Connecting to the clusters
Copying files
Editing files
Slurm Quick Start Tutorial
MANAGING FILES
Disk space
Transferring files to and from the clusters
⊟ Using the common filesystem
Home
Trfs
Proj
Soft
Behind the scenes
Sharing files among CÉCI users
Making your files Secure
Making your files safe
Long term data storage
Docs » Using the common filesystem

Using the common filesystem

All CÉCI clusters are connected to a central storage system that is visible to all compute nodes of all clusters. This system runs on a fast, dedicated, network. It will become the home of the users in the near future, but in this first phase, it is set up as an additional home besides the default, cluster-specific, home.

This storage system is installed at two CÉCI locations (at ULiège and UCL) and data are replicated synchronously on both locations to ensure data safety and a certain level of high availability. Moreover, on each site, a local cache is setup to mask the latencies of the network and make sure the user experience is as smooth as possible. Those caches are replicated asynchronously with the central storage, meaning that files that are written there will appear after some delay on the other clusters.

Note

There is no point in creating a file from a cluster and then quickly going to another cluster and running `ls` rapidly waiting for the file to appear. The file will be copied on the main storage only after 15 seconds, and the metadata will be copied on the other clusters after up to 60 seconds.

It also means that if you modify the same file from two different clusters, the result is undefined.

Warning

Do not write to the same file from two different clusters at the same time. This would corrupt the file.

After some delay, the files which are not read from a cluster are removed from the cache connected to that cluster. The metadata are kept, meaning that the files appear when you use `ls` but the data are not. The data will be synced back as soon as you try to access the file.

Used space and quotas?

Just use the `ceci-quota` command on any cluster

```
 [myuser@dragon2.dragon2-ctrl0: ~]---> $ ceci-quota

 Diskquotas for user myuser
Filesystem       used      limit      files      limit
$HOME         7.3 GiB   40.0 GiB     205641  unlimited
$CECIHOME    11.4 GiB  100.0 GiB       4390     100000
$CECITRSF    64.0 kiB    1.0 TiB          8  unlimited
```

```
 [myuser@lemaitre3.lm3-w001: ~]---> $ ceci-quota

 Diskquotas for user myuser
Filesystem       used      limit      files      limit
$HOME           4.14G       100G      3.82K
/scratch       4.3 GB  unlimited          8  unlimited
$CECIHOME    11.4 GiB  100.0 GiB       4390     100000
$CECITRSF    64.0 kiB    1.0 TiB          8  unlimited
```

Jobs submission

How do we control the data location from a Slurm job?

With the pre-defined environment variables:

`$HOME`

`$LOCALSCRATCH`

`$GLOBALSCRATCH`

`$CECIHOME`

Extra useful variables defined on-the-fly when submitting a job:

`$SLURM_JOB_ID` the Job ID value
`$SLURM_SUBMIT_DIR` directory where the job was submitted from

Example of basic sequential write

```
#!/bin/bash
#SBATCH --job-name=job-test
#SBATCH --time=00:15:00 # hh:mm:ss
#SBATCH --ntasks=1
#SBATCH --mem-per-cpu=2000 # megabytes
#SBATCH --partition=batch

echo ""
hn=`hostname`
echo "running on $CLUSTER_NAME node: $hn"

echo ""
echo dump file to GLOBALSCRATCH: $GLOBALSCRATCH

dd if=/dev/zero of=$GLOBALSCRATCH/testdump bs=1M count=2048
sync

echo ""

...
```

Please **DON'T** run this on your own !!
Is shown here just for illustrative purposes.

Example on lemaitre3

```
running on lemaitre3 node: lm3-w080.cluster


dump file to GLOBALSCRATCH:
2048+0 records in
2048+0 records out
2147483648 bytes (2.1 GB) copied, 1.66903 s, 1.3 GB/s

dump file to LOCALSCRATCH:
2048+0 records in
2048+0 records out
2147483648 bytes (2.1 GB) copied, 1.99117 s, 1.1 GB/s

dump file to HOME:
2048+0 records in
2048+0 records out
2147483648 bytes (2.1 GB) copied, 5.33424 s, 403 MB/s

dump file to CECIHOME:
2048+0 records in
2048+0 records out
2147483648 bytes (2.1 GB) copied, 18.8179 s, 114 MB/s
```

Similar order of magnitude for both SCRATCH

In the case of multithreaded multinode jobs GLOBALSCRATCH performance can be pushed higher (and is the only option anyway for those jobs)

An order of magnitude below respect the others

Example on hercules2

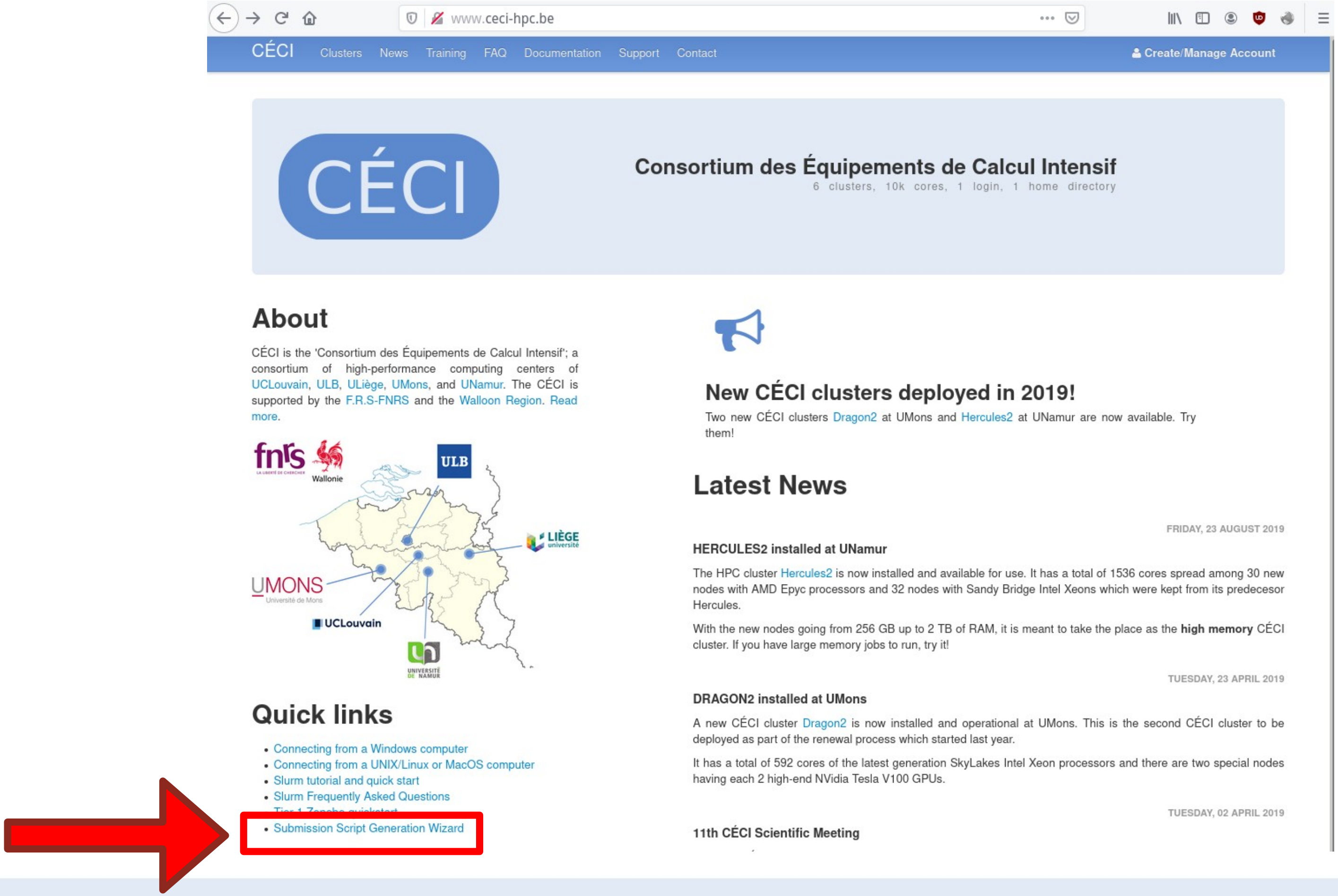
```
running on hercules node: her2-w113

dump file to GLOBALSCRATCH:
2048+0 records in
2048+0 records out
2147483648 bytes (2.1 GB) copied, 5.24254 s, 410 MB/s

dump file to LOCALSCRATCH:
2048+0 records in
2048+0 records out
2147483648 bytes (2.1 GB) copied, 1.19075 s, 1.8 GB/s

dump file to HOME:
2048+0 records in
2048+0 records out
2147483648 bytes (2.1 GB) copied, 9.93967 s, 216 MB/s

dump file to CECIHOME:
2048+0 records in
2048+0 records out
2147483648 bytes (2.1 GB) copied, 13.4418 s, 160 MB/s
```

LOCALSCRATCH is an order of magnitude above all other solutions

But still GLOBALSCRATCH is there to be used (or to store data after a job is done with I/O LOCALSCRATCH)

These are still lower than the others

Jobs submission



Consortium des Équipements de Calcul Intensif

6 clusters, 10k cores, 1 login, 1 home directory

About

CÉCI is the 'Consortium des Équipements de Calcul Intensif'; a consortium of high-performance computing centers of UCLouvain, ULB, ULiège, UMons, and UNamur. The CÉCI is supported by the F.R.S-FNRS and the Walloon Region. Read more.

Quick links

- Connecting from a Windows computer
- Connecting from a UNIX/Linux or MacOS computer
- Slurm tutorial and quick start
- Slurm Frequently Asked Questions
- Tier-1 Zenobe quickstart
- Submission Script Generation Wizard

New CÉCI clusters deployed in 2019!

Two new CÉCI clusters Dragon2 at UMons and Hercules2 at UNamur are now available. Try them!

Latest News

FRIDAY, 23 AUGUST 2019

HERCULES2 installed at UNamur

The HPC cluster Hercules2 is now installed and available for use. It has a total of 1536 cores spread among 30 new nodes with AMD Epyc processors and 32 nodes with Sandy Bridge Intel Xeons which were kept from its predecesor Hercules.

With the new nodes going from 256 GB up to 2 TB of RAM, it is meant to take the place as the **high memory** CÉCI cluster. If you have large memory jobs to run, try it!

TUESDAY, 23 APRIL 2019

DRAGON2 installed at UMons

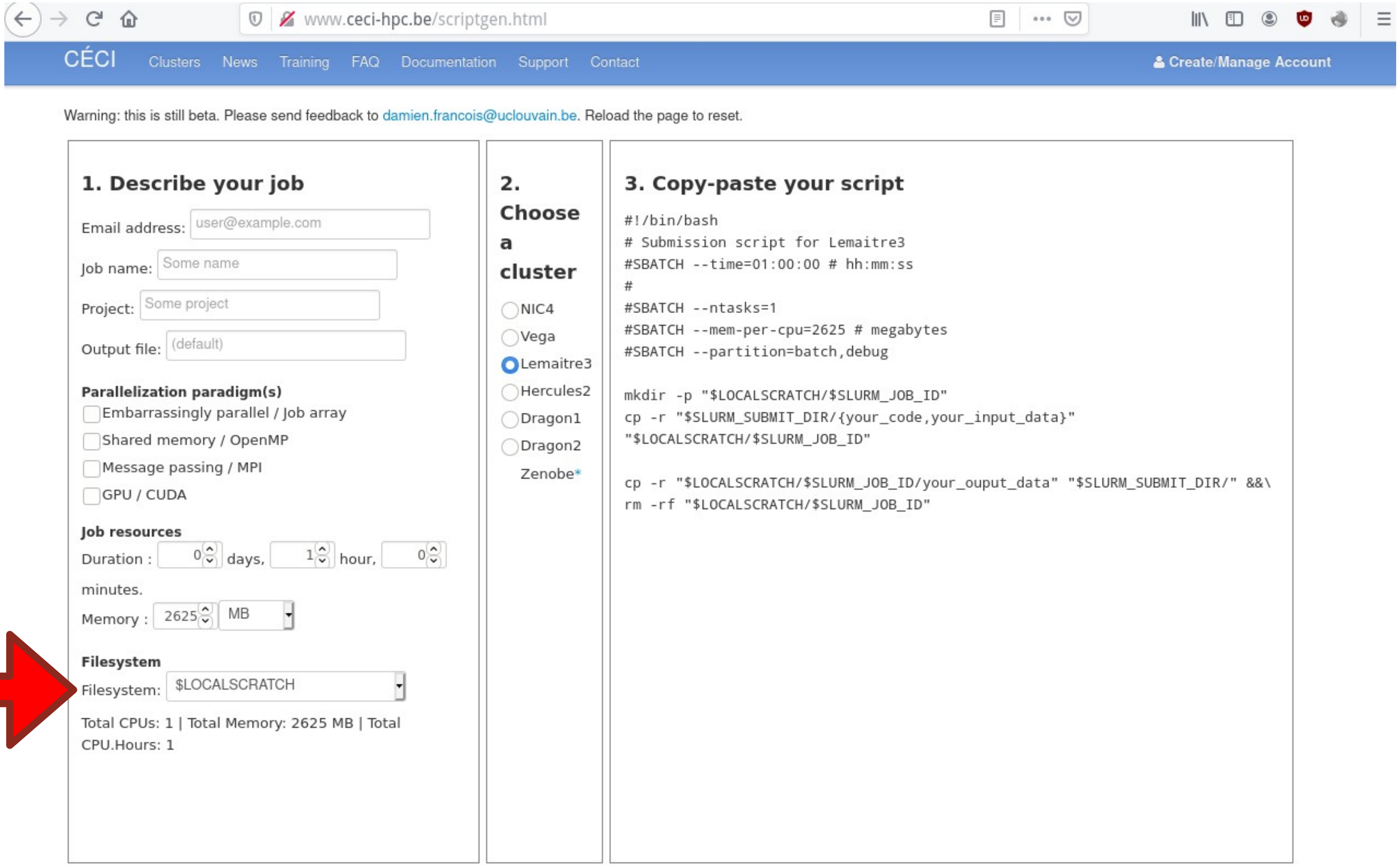
A new CÉCI cluster Dragon2 is now installed and operational at UMons. This is the second CÉCI cluster to be deployed as part of the renewal process which started last year.

It has a total of 592 cores of the latest generation SkyLakes Intel Xeon processors and there are two special nodes having each 2 high-end NVidia Tesla V100 GPUs.

TUESDAY, 02 APRIL 2019

11th CÉCI Scientific Meeting

Jobs submission



Warning: this is still beta. Please send feedback to damien.francois@uclouvain.be. Reload the page to reset.

1. Describe your job

Email address: user@example.com

Job name: Some name

Project: Some project

Output file: (default)

Parallelization paradigm(s)

- □ Embarrassingly parallel / Job array
- □ Shared memory / OpenMP
- □ Message passing / MPI
- □ GPU / CUDA

Job resources

Duration : 0 days, 1 hour, 0 minutes.

Memory : 2625 MB

Filesystem

Filesystem: $LOCALSCRATCH

Total CPUs: 1 | Total Memory: 2625 MB | Total CPU.Hours: 1

2. Choose a cluster

- ○ NIC4
- ○ Vega
- ● Lemaitre3
- ○ Hercules2
- ○ Dragon1
- ○ Dragon2
- Zenobe*

3. Copy-paste your script

```
#!/bin/bash
# Submission script for Lemaitre3
#SBATCH --time=01:00:00 # hh:mm:ss
#
#SBATCH --ntasks=1
#SBATCH --mem-per-cpu=2625 # megabytes
#SBATCH --partition=batch,debug

mkdir -p "$LOCALSCRATCH/$SLURM_JOB_ID"
cp -r "$SLURM_SUBMIT_DIR/{your_code,your_input_data}"
"$LOCALSCRATCH/$SLURM_JOB_ID"

cp -r "$LOCALSCRATCH/$SLURM_JOB_ID/your_ouput_data" "$SLURM_SUBMIT_DIR/" &&\
rm -rf "$LOCALSCRATCH/$SLURM_JOB_ID"
```

Examples ….

We are going to check the examples available on the clusters at:

```
/CECI/proj/training/ceci_storages
```

DISCLOSURE

WARNING: No data on the CECI clusters has backups

You are responsible of copying over your useful data you need to store long term somewhere else

Some of the CECI universities provide solutions see:
https://support.ceci-hpc.be/doc/_contents/ManagingFiles/LongtermStorage.html

To wrap up

- For all clusters

  Never do direct I/O on your `$HOME`

  Prioritize the usage of `$LOCALSCRATCH` if your jobs allow it (e.g. jobs running on a single node)
  Remember this area is shared with other users of the node and there's no quota!!

  Never redirect outputs to -> `/tmp` use always `$LOCALSCRATCH` instead

- Lemaitre3 and NIC5

  For your multi-node MPI jobs always rely on using `$GLOBALSCRATCH` **never** your `$HOME`

- **Remember to backup your useful data somewhere else**