Consortium des Équipements de Calcul Intensif

Efficient data storage on the CECI clusters

Ariel Lozano

CÉCI HPC Training
5 Dec 2023

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

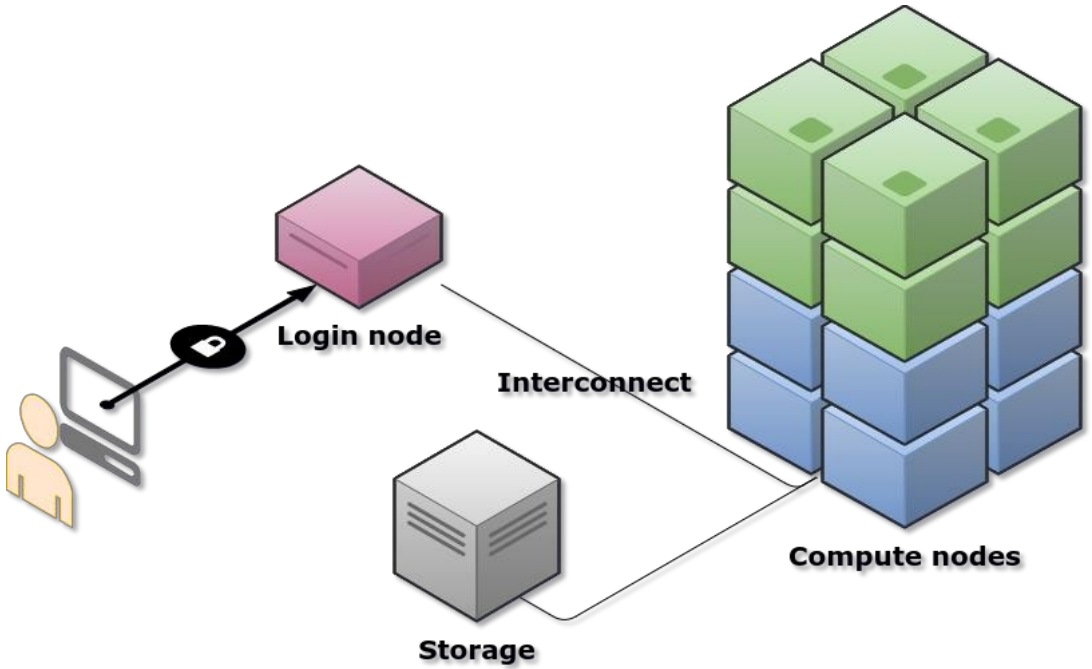
Some context

- Nowadays the best performant **units** of long term storage provides ~2 GB/s of sequential read/write. This can go 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)

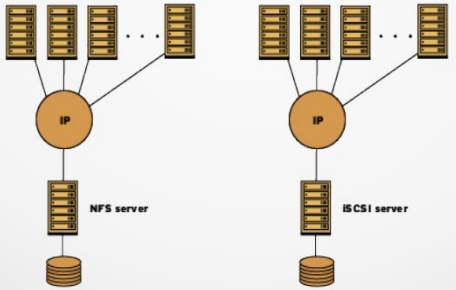
- A computer ‘cluster’ is a group of **linked** computers working together closely, so that in many respects they form a single computer
- Corollary: Access to **most** of the different storage solutions happens via the network

Previous: Network storage solutions

Network filesystem

One source many consumers

NAS: ex. NFS SAN: ex. GFS2



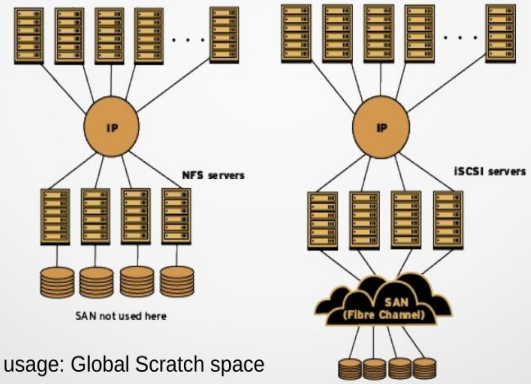
Typical usage: Home directories, Mass storage

10

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

Parallel / distributed filesystem

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



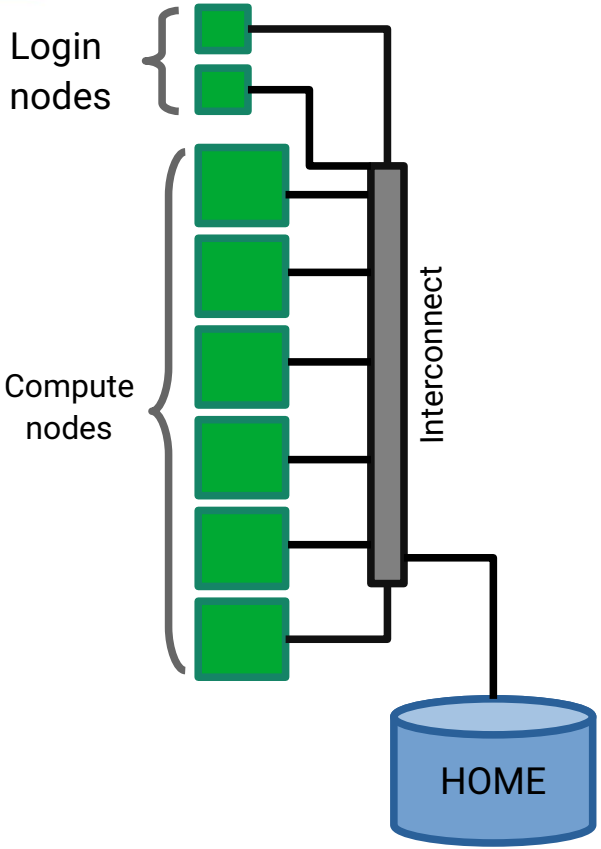
Typical usage: Global Scratch space

11

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

Damien François, "Introduction to data storage and access"

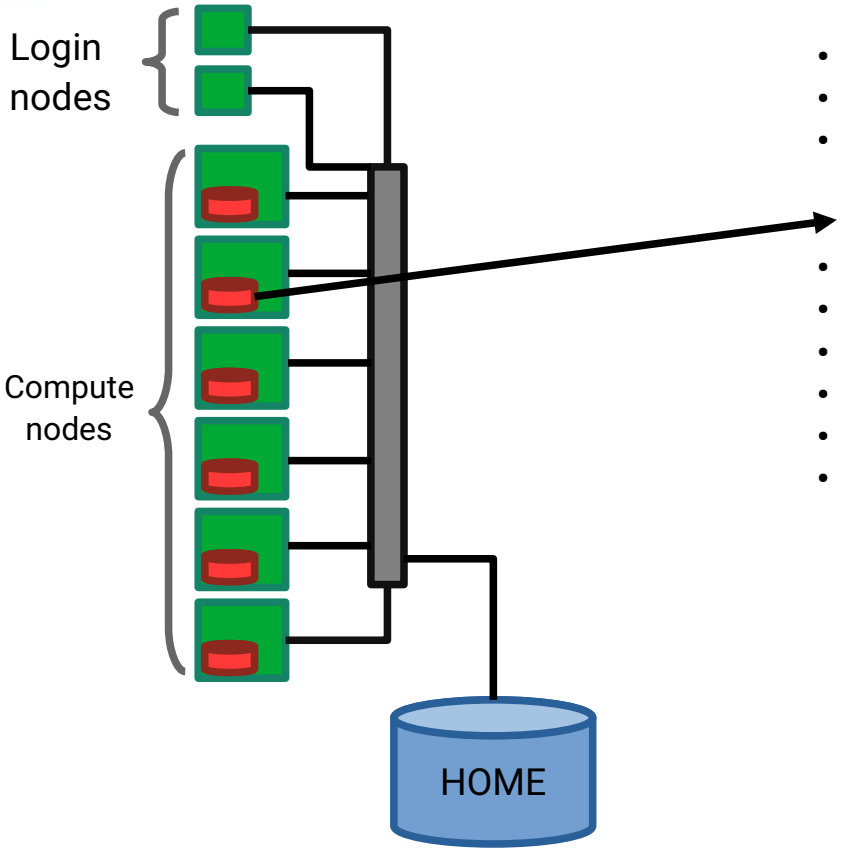
Storages on CECI clusters



`$HOME`

- Area where you land on login
- Network storage accessible by all compute nodes and login
- Single NFS server for all clients

Storages on CECI clusters



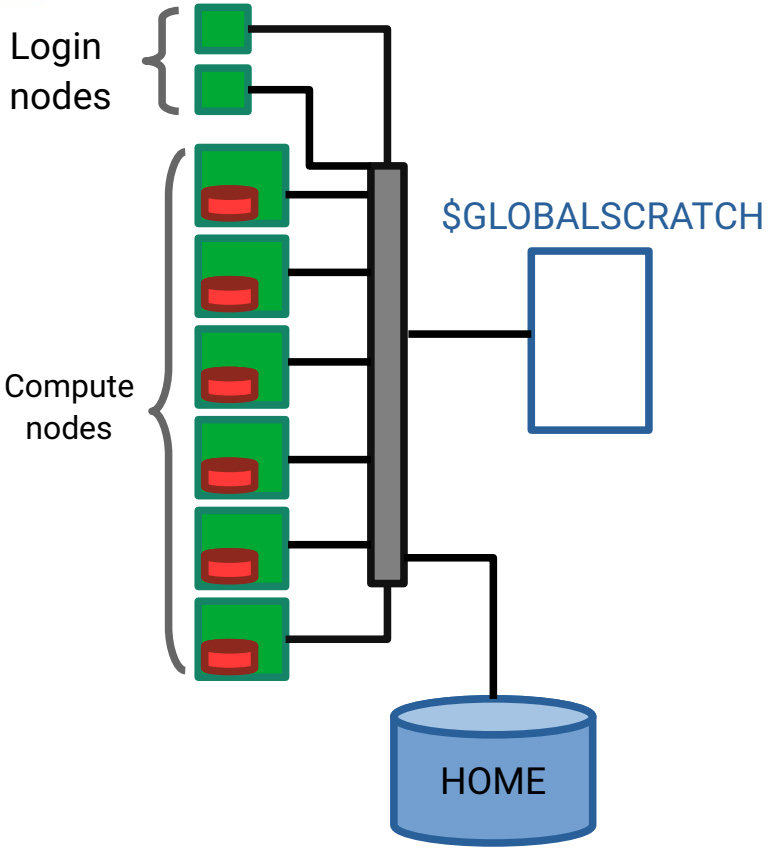
`$HOME`

- Area where you land on login
- Network storage accessible by all compute nodes and login
- Single NFS server for all clients

`$LOCALSCRATCH`

- Internal high performance storage inside each compute node
- Accessible **only** from the compute node running your job
- Shared with other jobs running on the same node
- Usage only temporary during the job runtime
- Serial jobs or Parallel SMP (openmp, python multiprocessing, ...)
- Data is **automatically purged** when job ends

Storages on CECI clusters



`$HOME`

- Area where you land on login
- Network storage accessible by all compute nodes and login
- Single NFS server for all clients

`$LOCALSCRATCH`

- Internal high performance storage inside each compute node
- Accessible **only** from the compute node running your job
- Shared with other jobs running on the same node
- Usage only temporary during the job runtime
- Serial jobs or Parallel SMP (openmp, python multiprocessing, ...)
- Data is **automatically purged** when job ends

`$GLOBALSCRATCH`

- Implemented via different setups but usually a parallel filesystem
- Network storage accessible by all compute nodes and login
- Can be composed of a single or multiple storage sources
- All jobs but **only option** for **multinode-parallel** jobs (big MPI jobs)
- Data there stays persistently (but all is removed in yearly maintenances)
- Please cleanup from time to time

Storages on CECI clusters

How much data can we handle on each area?

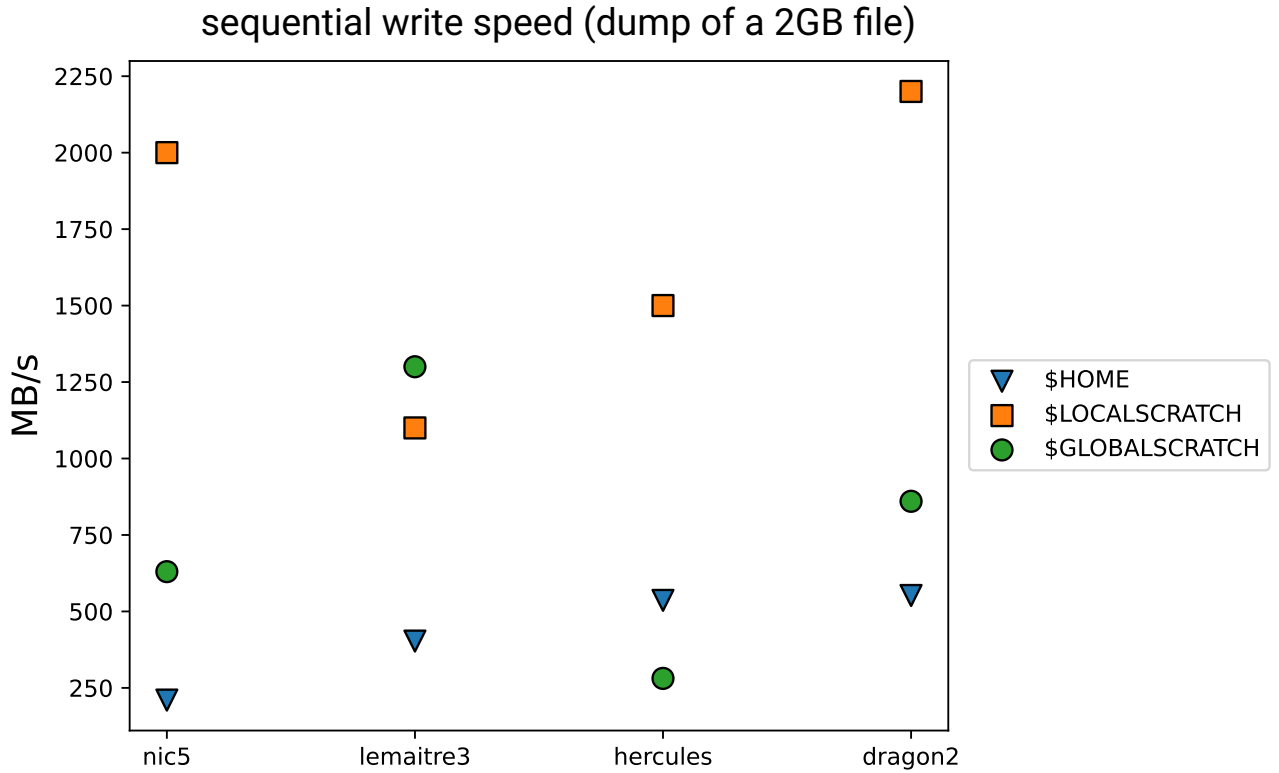
Cluster	$HOME (nfs server)	$LOCALSCRATCH (hardware support)	$GLOBALSCRATCH (fstype)
NIC5	100 GB	370GB (local SSD)	5TB (520TB beegfs)
Lemaitre3	100 GB	200GB (local SSD)	XX[1] (415TB beegfs)
Dragon2	40 GB	3TB (HDD RAID0)	XX[1] (52TB beegfs)
Hercules	200 GB	her2-w065..096: 1TB her2-w099..126: 4TB her2-w127..128: 8TB (HDD RAID0)	400GB soft / 4TB hard (nfs server)

More info: https://support.ceci-hpc.be/doc/_contents/ManagingFiles/Storage.html#quota-label

[1] No limits enforced but remember resources are shared among all ~800 CECI users!!

Storages on CECI clusters

What performance can I expect?



- Note this is not a definitive benchmark of the filesystems
- Different I/O patterns on your job can produce very diverse results

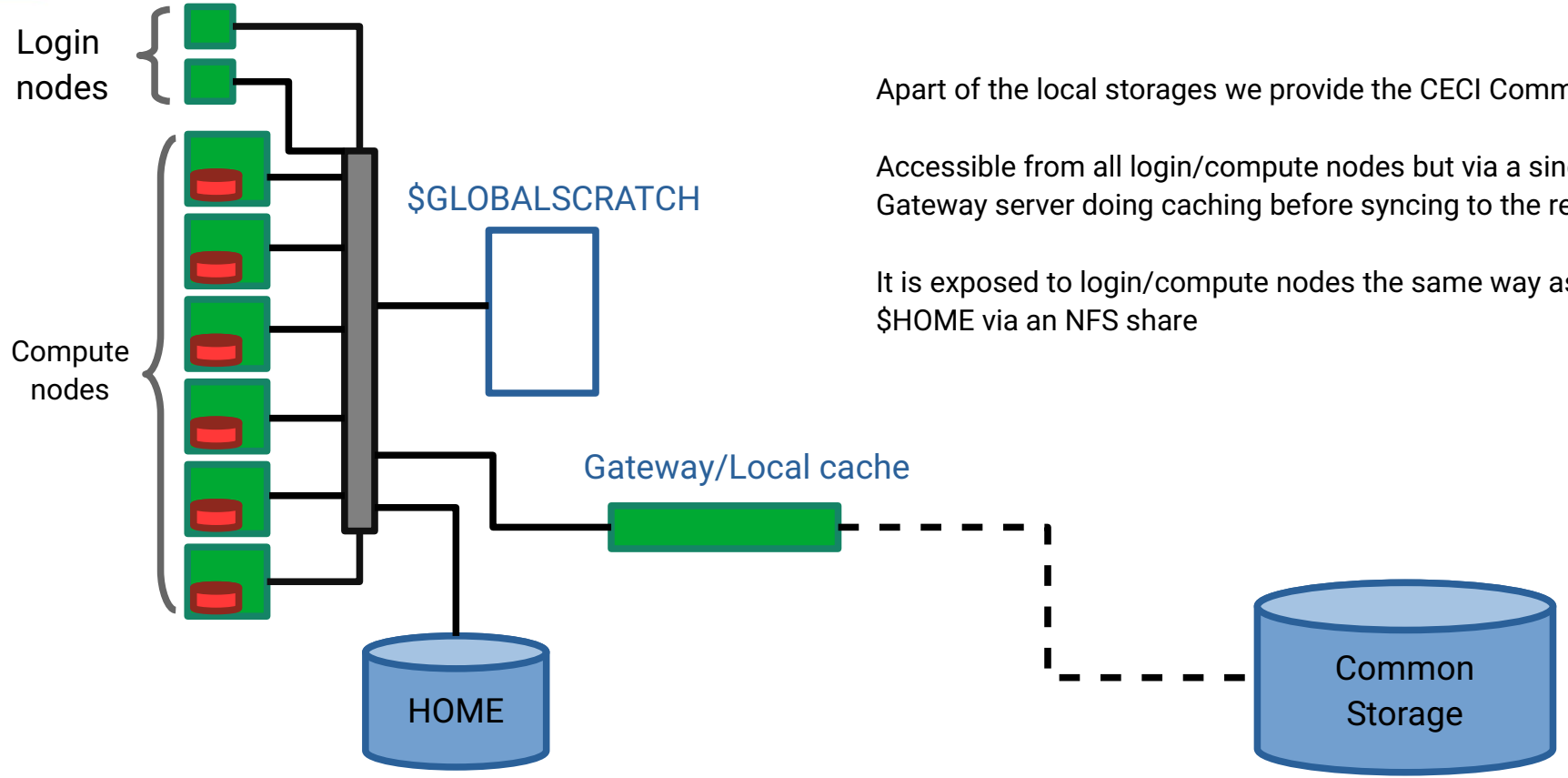
Storages on CECI clusters

Wrap-up

- $LOCALSCRATCH is in average an order of magnitude above all other solutions
- If all your input/output data volume for your job fits on $LOCALSCRATCH **use it** your code will speed up just for doing that
- Still $GLOBALSCRATCH is there to be used, is the only choice for parallel multi-node jobs
- $GLOBALSCRATCH can also be used to store big input data, or recover useful output for a job working on $LOCALSCRATCH
- You actually have no reason to use $HOME for job I/O. The space available there is smaller than on each SCRATCH areas and the performance is the lowest

CECI Common storage

external remote storage accesible by all clusters



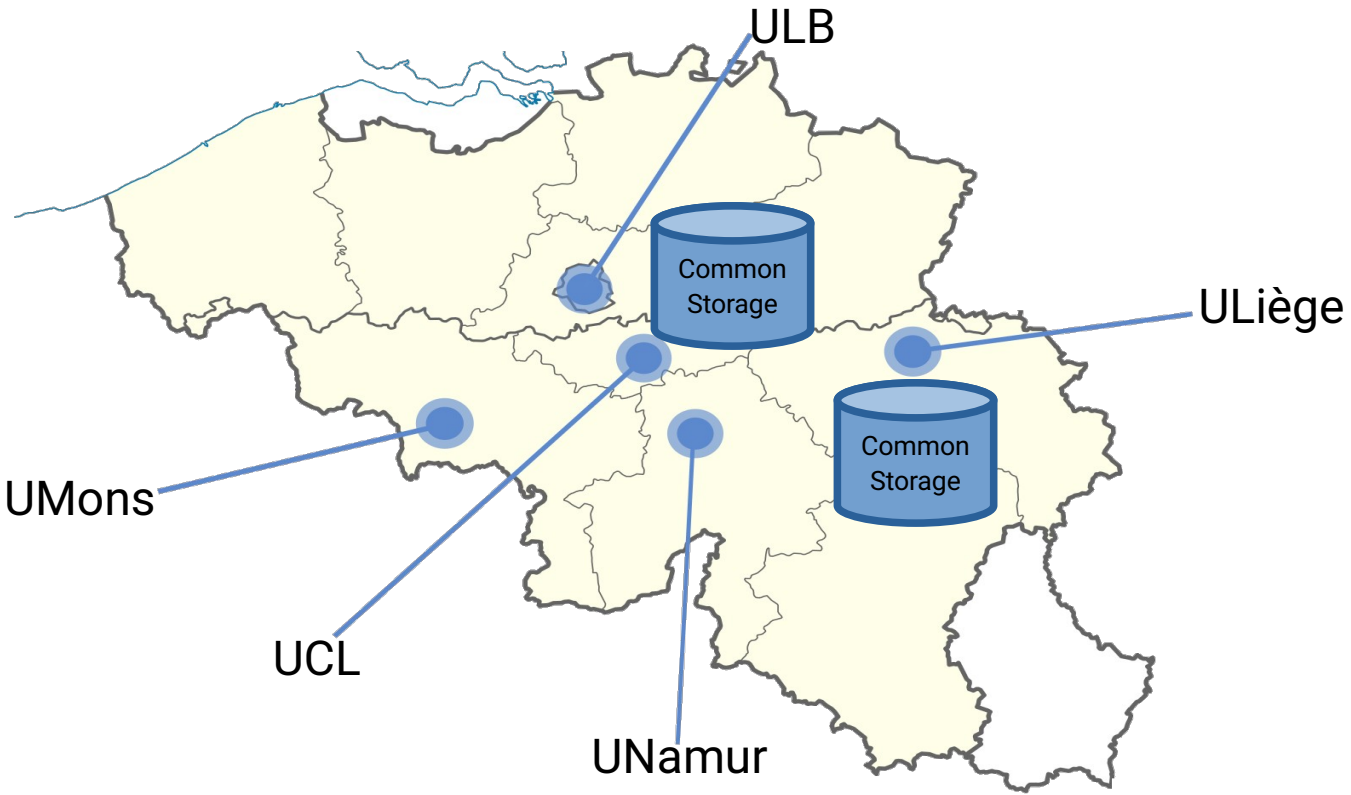
Apart of the local storages we provide the CECI Common Storage

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 data interconnect among the 5 sites for this solution

CECI Common storage

external remote storage accesible by all clusters

`/CECI/home`
- Personal directory path defined via `$CECIHOME` variable
- Quota of 100GB

`/CECI/trsf`
- Personal directory path defined via `$CECITRSF` variable
- Meant only for **temporary** copying from one cluster to another
- Data here is subject to be purged every 6 months
- Quota of 1TB soft 10TB hard

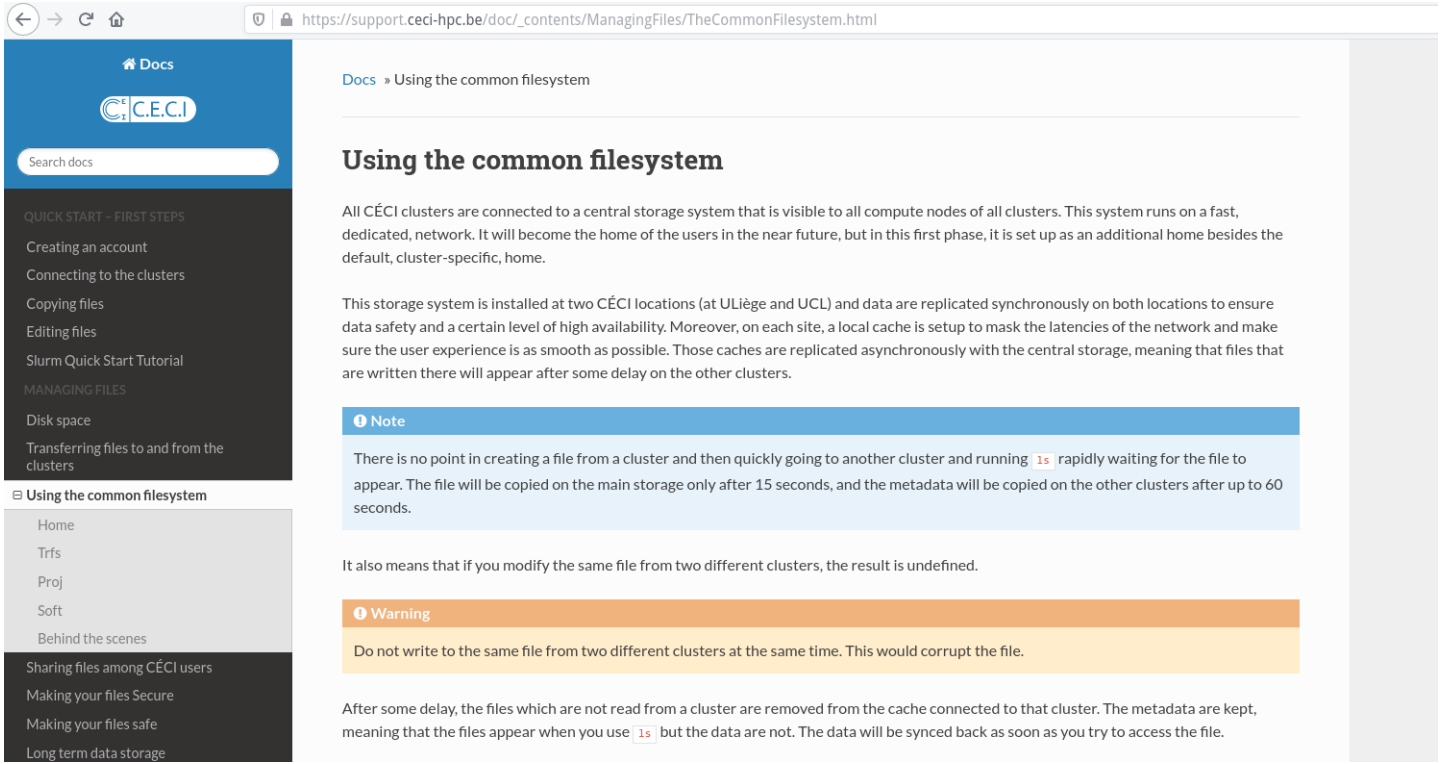
`/CECI/proj`
- Area where a team with a project can get a common folder for sharing data
- Must be requested by a PI
- Quota decided according to the project's needs

`/CECI/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



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?

Making use of 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

Jobs submission

LOCALSCRATCH example

```
#!/bin/bash
#SBATCH --job-name=stest1
#SBATCH --time=00:15:00
#SBATCH --ntasks=1
#SBATCH --mem-per-cpu=1000
#SBATCH --partition=batch

module load releases/2021b
module load SciPy-bundle/2021.10-foss-2021b

# define some useful directory names
RUNDIR="${LOCALSCRATCH}"
SUBMITDIR="${SLURM_SUBMIT_DIR}"

# on this example we assume our input files are on the
# same directory where we run sbatch myjob.sh
cp -r ${SUBMITDIR}/input_1000.in ${RUNDIR}/input.in

# change to the run directory on the compute node
cd ${RUNDIR}

# execute your code
python ${HOME}/bin/mycode/mycode1.py

# recover your useful output
cp -r long.out ${SUBMITDIR}/
```

- Python code that requires just an input file “input.in”
- The code is located on a folder inside my $HOME
- The only useful output it will produce is “long.out”

Jobs submission

GLOBALSCRATCH example

```
$ mkdir -p $GLOBALSCRATCH/myjob1000
$ cp -r input_1000.in $GLOBALSCRATCH/myjob1000/input.in
$ cd $GLOBALSCRATCH/myjob1000

> i create my submission script myjob.sh as e.g. below

#!/bin/bash
#SBATCH --job-name=gtest1
#SBATCH --time=00:15:00
#SBATCH --ntasks=1
#SBATCH --mem-per-cpu=1000
#SBATCH --partition=batch

module load releases/2021b
module load SciPy-bundle/2021.10-foss-2021b

python ${HOME}/bin/mycode/mycode1.py

> and we can submit it directly from here
$ sbatch myjob.sh
```

We can prepare everything before inside $GLOBALSCRATCH, input and submission script, then submit from there without extra data movements.

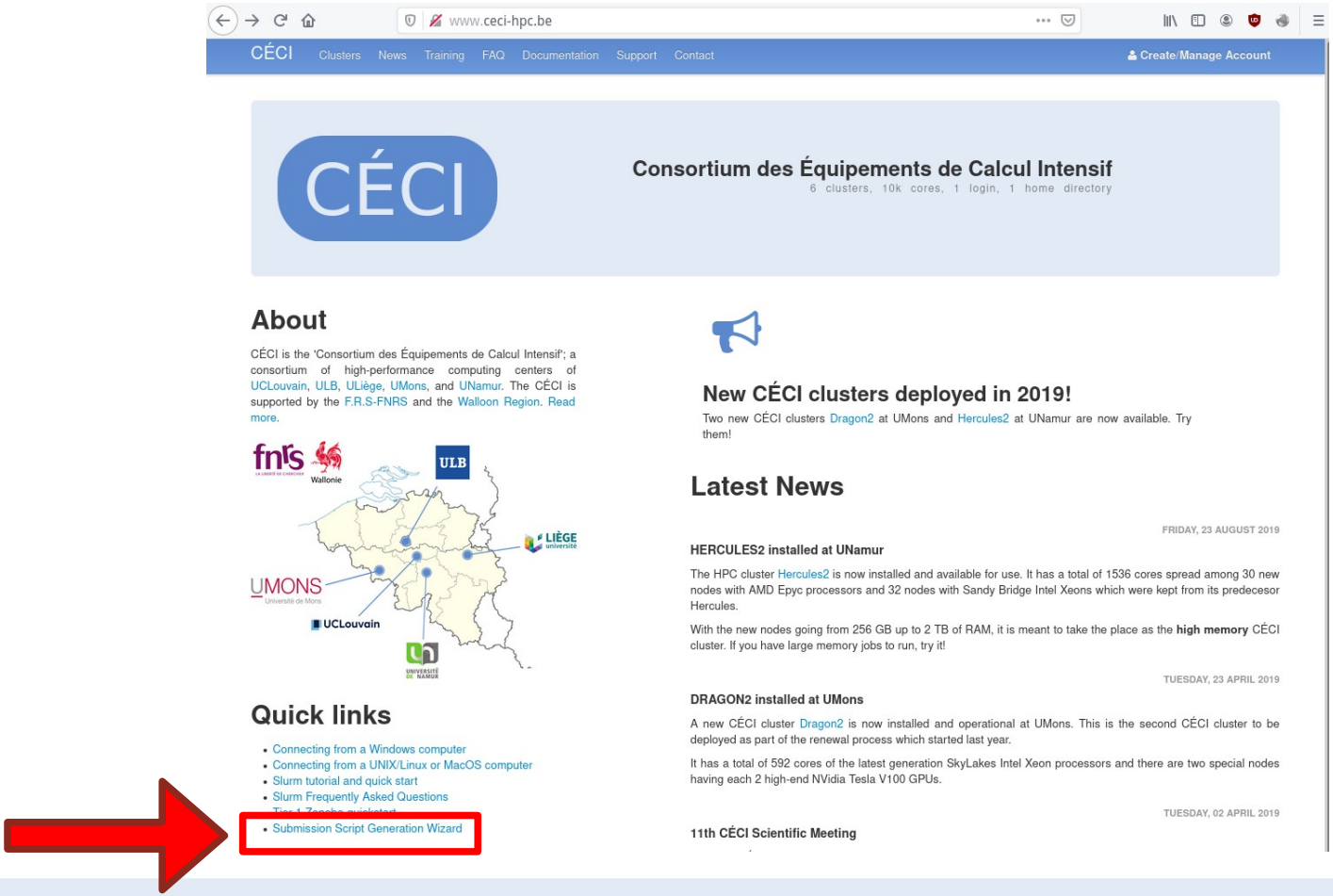
This is just an illustrative example !

This kind of approach must be avoided if all your input/output fits on LOCALSCRATCH

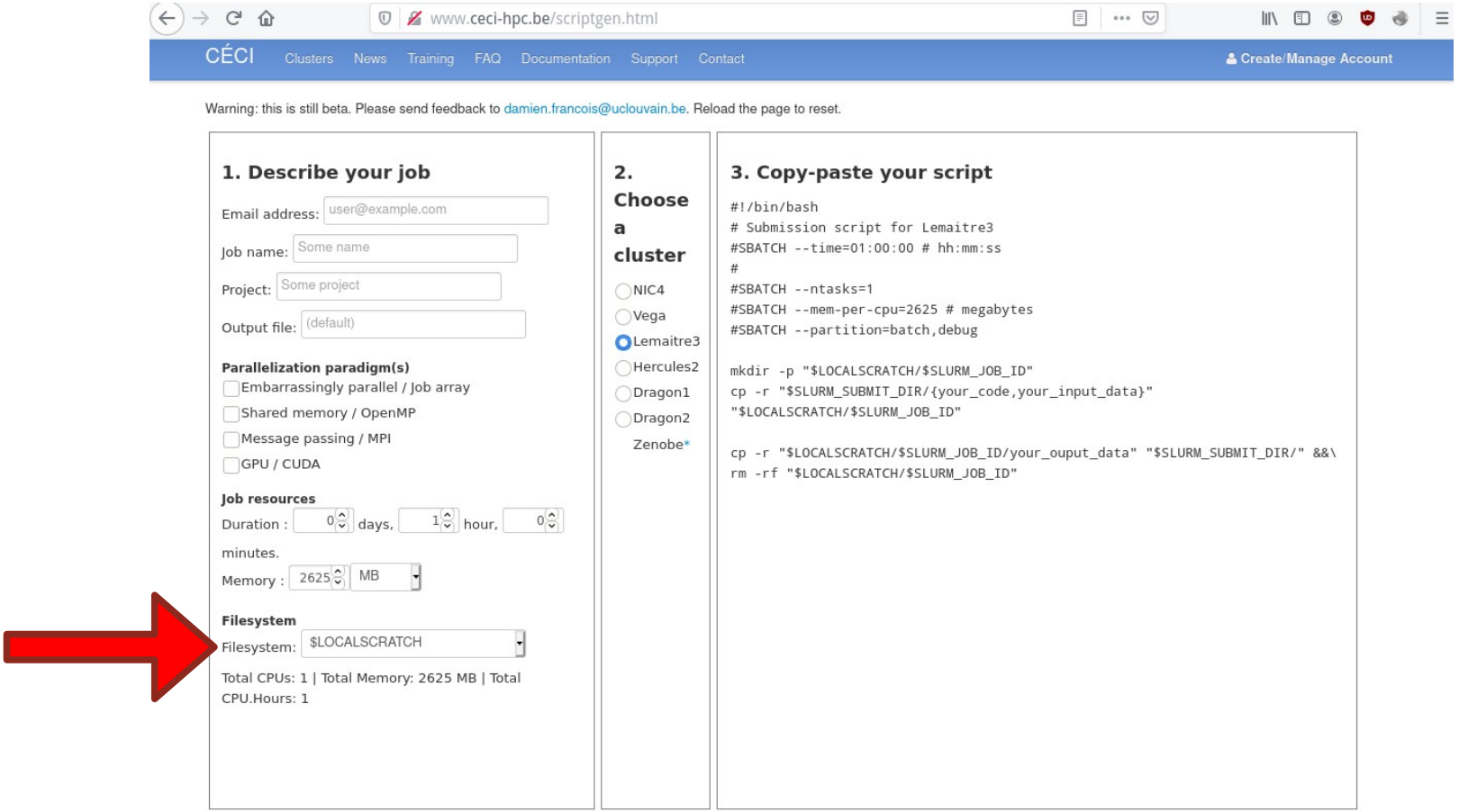
Use the previous approach on that case

Or also **consider a mix** if the input data is too big, but the produced output fits on LOCALSCRATCH

Jobs submission



Jobs submission



Examples ….

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

```
cat /CECI/proj/training/ceci_storages/README.md
```

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**