Introduction to Leonardo Data Centric and General Purpose (DCGP)

February 18th, 2025

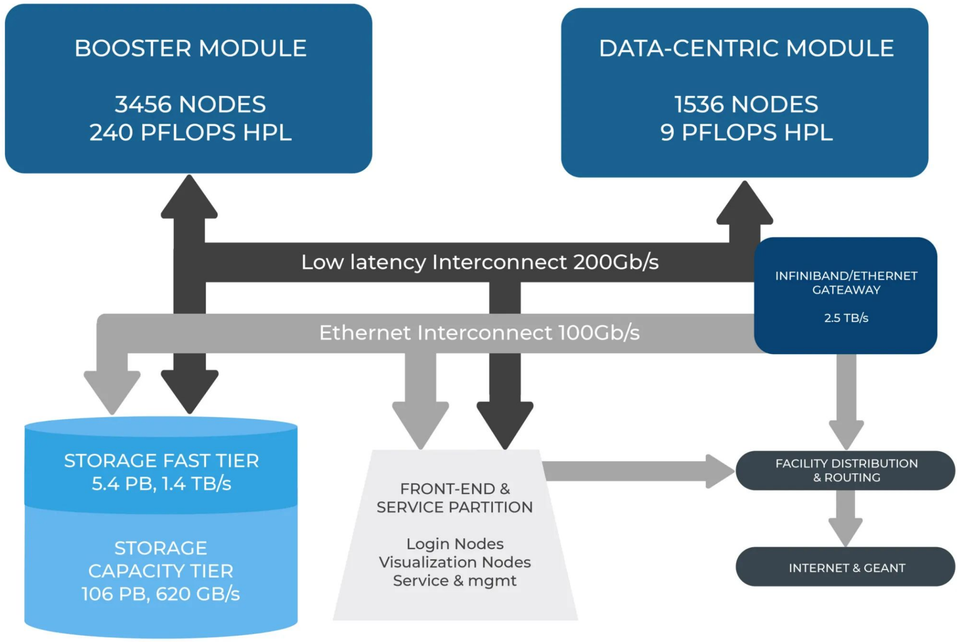
Caterina Caravita
c.caravita@cineca.it

CINECA - High Performance Computing Department

Outline

- ➢ **Leonardo infrastructure**
- ➢ Access HPC resources and filesystems
- ➢ Software environment
- ➢ Programming environment
- ➢ Production environment
- ➢ Final remarks

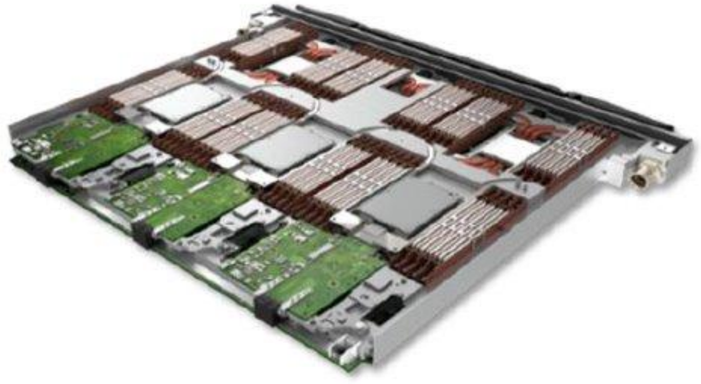
Leonardo infrastructure and login nodes



Atos BullSequana X430-E6

- ➢ Processors (dual-socket): **2x CPU Intel *Whitley* ICP06, 32 cores Intel *Ice Lake*** (64 cores/node), **2.4 GHz**
- ➢ RAM: 512 (16x32) GB RAM DDR4 3200 MHz
- ➢ Disk: 14 TB HDD
- ➢ **NO GPUs**

Data Centric and General Purpose (CPU-only) partition

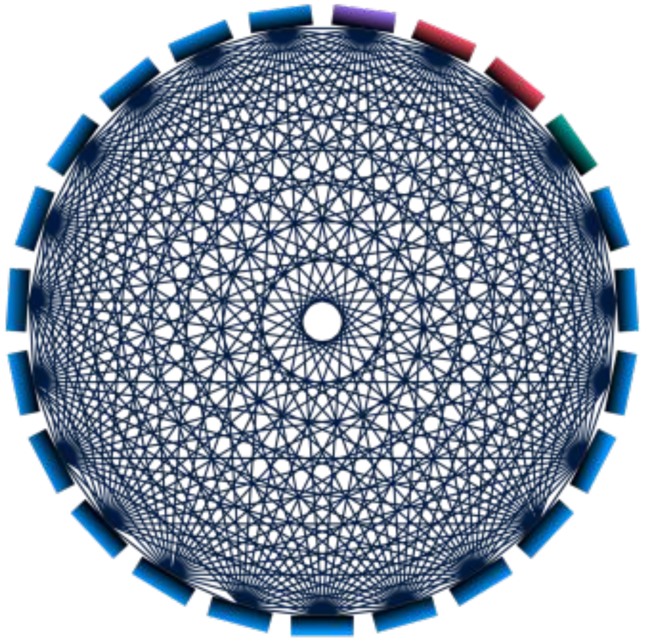


BullSequana X2140 three-node CPU Blade

- ➢ 1536 nodes: **lrdn[1537-4992]**
- ➢ Processors (dual-socket): **2x CPU Intel *Xeon* 8480+, 56 cores Intel *Sapphire Rapids*** (**112 cores/node**), **3.8 GHz** (turbo enabled)
- ➢ RAM: 512 (16 x 32) GB DDR5 4800 MHz
- ➢ **Disk: 1x SSD 3.84 TB M.2 NVMe**
- ➢ Internal network: PCIe Gen5,
  1x port HDR100 100Gb/s network interface

Peak performance: about 13 PFlops

Inter-node network topology



Booster Module nodes
I/O cell
Data-Centric cells
Hybrid cell (Booster + Data-Centric nodes)

Dragonfly+ topology

based on **Nvidia Mellanox Infiniband HDR, bidirectional bandwidth of 200 Gb/s** (shared between Leonardo Booster and DCGP)

- ➢ All nodes are divided into cells
- ➢ Non-blocking, two-layer **Fat Tree** within the cells
- ➢ All to all connection between cells

- ➢ **Adaptive routing algorithm**: SLURM will take care of the "best"-possible node allocations

Storage

Fast Tier

5.4 PB, 1.4 TB/s

NVMe storage (SSD disks)

➢ HOME, PUBLIC, FAST SCRATCH

Capacity Tier

106 PB, read 744 GB/s - write 620 GB/s

HDD disks

➢ WORK, LARGE SCRATCH, DRES

Outline

LEONARDO
EuroHPC
CINECA
Founded by the European Union
L E O N A R D O

Become a new HPC user

- **Register on the UserDB Portal:** https://userdb.hpc.cineca.it/

- **Get associated to an active account**

  → Principal Investigator (PI): we create the account and set you as PI on the UserDB

  → Collaborator: ask your PI to associate you to the account on the UserDB

- **Request the "HPC Access" on UserDB**

  → You will receive two mails:

  one with your HPC username, and one to set an HPC password and configure the 2FA

https://wiki.u-gov.it/confluence/display/SCAIUS/Get+Started

Access to Leonardo

The access to CINECA HPC systems requires a **two-factor authentication (2FA)**.

First time

- **Activate the 2FA**: authenticate on our **Identity Provider** at https://sso.hpc.cineca.it using your HPC username and password.
  → You will need an **app to generate authentication codes** (e.g. Google Authenticator)

- **Install and configure the smallstep client** (depending on your OS)

Any access to the cluster

- **Request the ssh certificate** to our Identity Provider via the smallstep client from your local shell.
  → A web page will open on the browser and you will be asked to insert a One-Time Password (OTP) from the app
  → ***Valid for 12 hours***

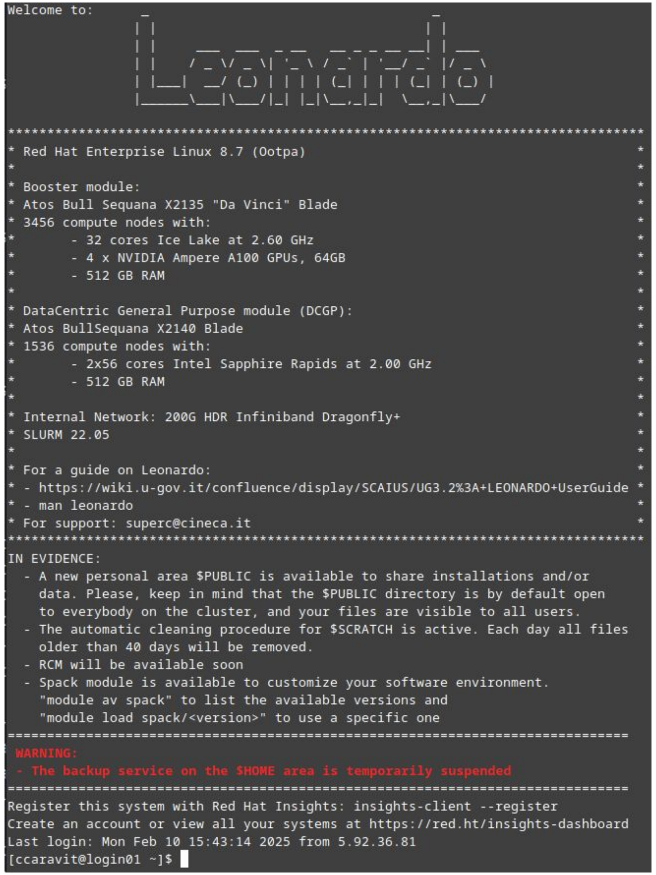
- **Access to the cluster via ssh:**

```
$ ssh <username>@login.leonardo.cineca.it
```

https://wiki.u-gov.it/confluence/display/SCAIUS/2%3A+Access+to+the+Systems

Slides, June 7th, 2023

Access to Leonardo



```
Welcome to:      _                                  _
                | |                                | |
                | |     ___  ___  _ __   __ _ _ __ __| | ___
                | |    / _ \/ _ \| '_ \ / _` | '__/ _` |/ _ \
                | |___|  __/ (_) | | | | (_| | | | (_| | (_) |
                |______\___|\___/|_| |_|\__,_|_|  \__,_|\___/

*********************************************************************************
* Red Hat Enterprise Linux 8.7 (Ootpa)                                          *
*                                                                               *
* Booster module:                                                               *
* Atos Bull Sequana X2135 "Da Vinci" Blade                                      *
* 3456 compute nodes with:                                                      *
*       - 32 cores Ice Lake at 2.60 GHz                                         *
*       - 4 x NVIDIA Ampere A100 GPUs, 64GB                                     *
*       - 512 GB RAM                                                            *
*                                                                               *
* DataCentric General Purpose module (DCGP):                                    *
* Atos BullSequana X2140 Blade                                                  *
* 1536 compute nodes with:                                                      *
*       - 2x56 cores Intel Sapphire Rapids at 2.00 GHz                          *
*       - 512 GB RAM                                                            *
*                                                                               *
* Internal Network: 200G HDR Infiniband Dragonfly+                              *
* SLURM 22.05                                                                   *
*                                                                               *
* For a guide on Leonardo:                                                      *
* - https://wiki.u-gov.it/confluence/display/SCAIUS/UG3.2%3A+LEONARDO+UserGuide *
* - man leonardo                                                                *
* For support: superc@cineca.it                                                 *
*********************************************************************************
IN EVIDENCE:
  - A new personal area $PUBLIC is available to share installations and/or
    data. Please, keep in mind that the $PUBLIC directory is by default open
    to everybody on the cluster, and your files are visible to all users.
  - The automatic cleaning procedure for $SCRATCH is active. Each day all files
    older than 40 days will be removed.
  - RCM will be available soon
  - Spack module is available to customize your software environment.
    "module av spack" to list the available versions and
    "module load spack/<version>" to use a specific one
===============================================================================
 WARNING:
 - The backup service on the $HOME area is temporarily suspended
===============================================================================
Register this system with Red Hat Insights: insights-client --register
Create an account or view all your systems at https://red.ht/insights-dashboard
Last login: Mon Feb 10 15:43:14 2025 from 5.92.36.81
[ccaravit@login01 ~]$
```

```
$ ssh <username>@login.leonardo.cineca.it
```

Motto of the day

➔ **Short system description**

➔ **“In evidence” messages**

➔ **“Important” messages**

(e.g. scheduled maintenances)

Filesystems

$HOME

- 50 GB per user
- user specific
- permanent
- daily backup (<u>soon</u>)

$PUBLIC

- 50 GB per user, <u>only on Leonardo</u>
- user specific (permissions **755**)
- permanent
- **no** backup

$SCRATCH

- no quota
- user specific
- temporary (data removed after 40 days)
- **no** backup

Filesystems

$HOME

- 50 GB per user
- user specific
- permanent
- daily backup (<u>soon</u>)

$PUBLIC

- 50 GB per user, <u>only on Leonardo</u>
- user specific (permissions **755**)
- permanent
- **no** backup

$SCRATCH

- no quota
- user specific
- temporary (data removed after 40 days)
- **no** backup

$WORK

- quota per account (default 1 TB)
- account specific
- permanent
- **no** backup

$FAST

- similar to $WORK
- **fast I/O**
- <u>only on Leonardo</u>

Filesystems

$HOME

- 50 GB per user
- user specific
- permanent
- daily backup (soon)

$PUBLIC

- 50 GB per user, only on Leonardo
- user specific (permissions **755**)
- permanent
- **no** backup

$SCRATCH

- no quota
- user specific
- temporary (data removed after 40 days)
- **no** backup

$WORK

- quota per account (default 1 TB)
- account specific
- permanent
- **no** backup

$FAST

- similar to $WORK
- **fast I/O**
- only on Leonardo

$TMPDIR

- local on nodes
- job specific
- **fast I/O**

DRES

- long storage on demand
- shared among accounts and platforms (not Leonardo)

All the filesystems are based on **Lustre**
→ Check your areas, disk usage and quota: **$ cindata**

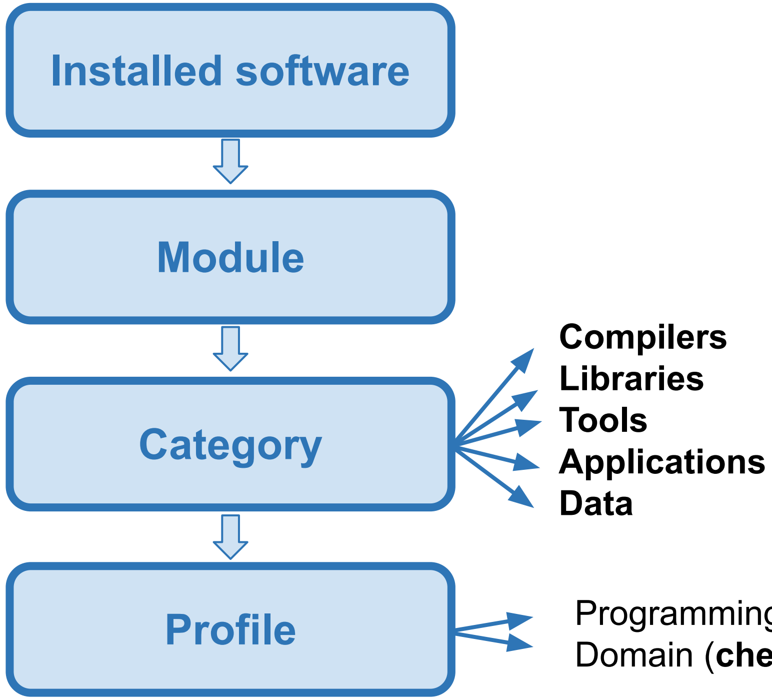
https://wiki.u-gov.it/confluence/display/SCAIUS/LEONARDO+User+Guide#LEONARDOUserGuide-DisksandFilesystems

Outline

- ➢ Leonardo infrastructure
- ➢ Access HPC resources and filesystems
- ➢ **Software environment**
- ➢ Programming environment
- ➢ Production environment
- ➢ Final remarks

Module environment

Any available software is offered on the clusters in a module environment.
The modules are organized in functional categories and collected in different profiles.

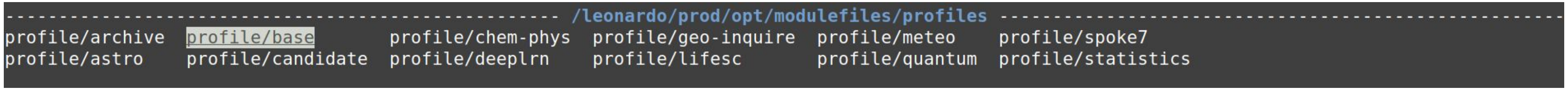


Base is the default profile:
automatically loaded after login,
containing basic modules
for programming activities

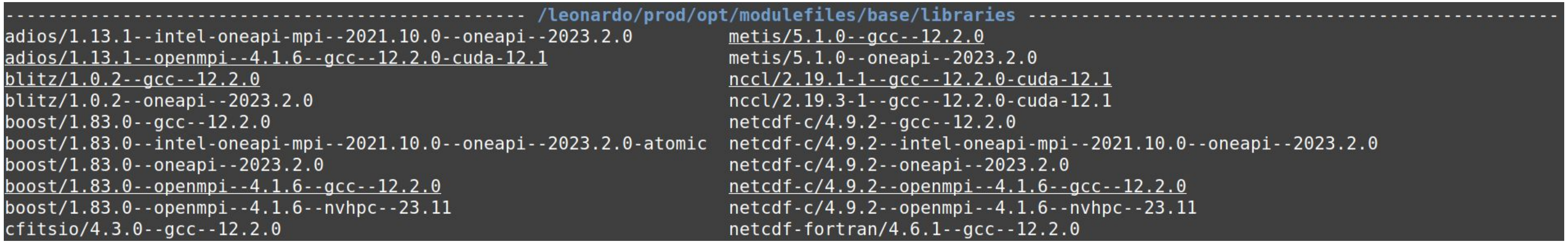
Programming (**base**): compilation, profiling, debugging…
Domain (**chem-phys, astro, bioinf…**): production activities

Module environment

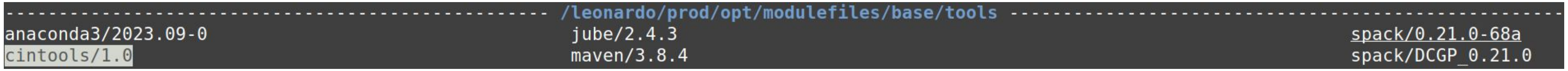
$ module avail



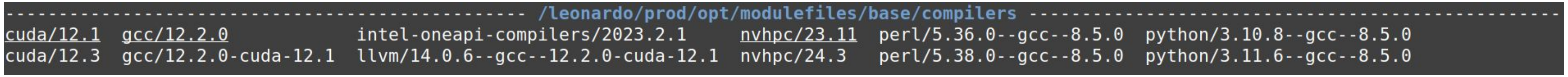
```
----------------------------------------------------- /leonardo/prod/opt/modulefiles/profiles -----------------------------------------------------
profile/archive  profile/base       profile/chem-phys  profile/geo-inquire  profile/meteo    profile/spoke7
profile/astro    profile/candidate  profile/deeplrn    profile/lifesc       profile/quantum  profile/statistics
```



```
--------------------------------------------------- /leonardo/prod/opt/modulefiles/base/libraries ---------------------------------------------------
adios/1.13.1--intel-oneapi-mpi--2021.10.0--oneapi--2023.2.0         metis/5.1.0--gcc--12.2.0
adios/1.13.1--openmpi--4.1.6--gcc--12.2.0-cuda-12.1                 metis/5.1.0--oneapi--2023.2.0
blitz/1.0.2--gcc--12.2.0                                            nccl/2.19.1-1--gcc--12.2.0-cuda-12.1
blitz/1.0.2--oneapi--2023.2.0                                       nccl/2.19.3-1--gcc--12.2.0-cuda-12.1
boost/1.83.0--gcc--12.2.0                                           netcdf-c/4.9.2--gcc--12.2.0
boost/1.83.0--intel-oneapi-mpi--2021.10.0--oneapi--2023.2.0-atomic  netcdf-c/4.9.2--intel-oneapi-mpi--2021.10.0--oneapi--2023.2.0
boost/1.83.0--oneapi--2023.2.0                                      netcdf-c/4.9.2--oneapi--2023.2.0
boost/1.83.0--openmpi--4.1.6--gcc--12.2.0                           netcdf-c/4.9.2--openmpi--4.1.6--gcc--12.2.0
boost/1.83.0--openmpi--4.1.6--nvhpc--23.11                          netcdf-c/4.9.2--openmpi--4.1.6--nvhpc--23.11
cfitsio/4.3.0--gcc--12.2.0                                          netcdf-fortran/4.6.1--gcc--12.2.0
```



```
---------------------------------------------------- /leonardo/prod/opt/modulefiles/base/tools ----------------------------------------------------
anaconda3/2023.09-0                                    jube/2.4.3                                                          spack/0.21.0-68a
cintools/1.0                                           maven/3.8.4                                                         spack/DCGP_0.21.0
```



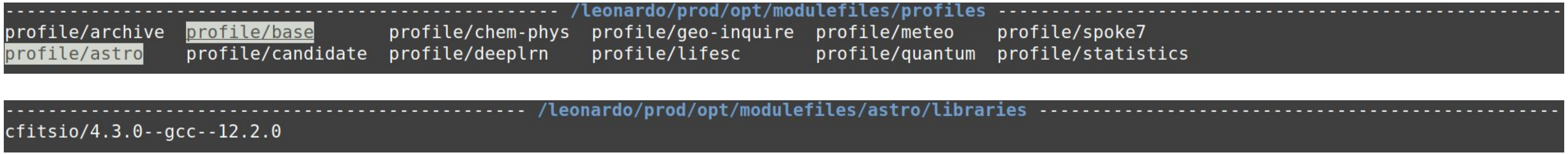
```
--------------------------------------------------- /leonardo/prod/opt/modulefiles/base/compilers ---------------------------------------------------
cuda/12.1  gcc/12.2.0            intel-oneapi-compilers/2023.2.1    nvhpc/23.11  perl/5.36.0--gcc--8.5.0  python/3.10.8--gcc--8.5.0
cuda/12.3  gcc/12.2.0-cuda-12.1  llvm/14.0.6--gcc--12.2.0-cuda-12.1  nvhpc/24.3   perl/5.38.0--gcc--8.5.0  python/3.11.6--gcc--8.5.0
```

Almost all the modules on Leonardo have been installed with **Spack**, and they report the Spack package name.

Module environment

$ module load profile/astro
$ module avail

Loaded profiles
are **added** to the environment



```
---------------------------------------------------- /leonardo/prod/opt/modulefiles/profiles ----------------------------------------------------
profile/archive   profile/base       profile/chem-phys  profile/geo-inquire  profile/meteo     profile/spoke7
profile/astro     profile/candidate  profile/deeplrn    profile/lifesc       profile/quantum   profile/statistics
```

```
------------------------------------------------- /leonardo/prod/opt/modulefiles/astro/libraries -------------------------------------------------
cfitsio/4.3.0--gcc--12.2.0
```

$ module show <module_name>/<version> → Print information about the module, such as dependencies, paths

$ module help <module_name>/<version> → Print the help of the software, its brief description and examples of the use

Module environment

$ modmap -m <module_name> → Detect all profiles, categories and modules available (e.g. different releases)

$ module load <profile>

$ module load <module_name>/<version> → **all the dependencies are automatically loaded; we recommend to specify the module version!**

$ module list → List all the profiles and modules loaded so far

You will find **modules compiled to support GPUs and modules suitable only for CPUs**.
You can check the compiler in the full name of the module, where the version is specified (e.g. gromacs/2022.3--intel-oneapi-mpi--2021.10.0--oneapi–2023.2.0).
Remind that modules compiled with nvhpc, cuda should be used only on the Booster partition, while modules compiled with gcc, intel, oneapi are suitable for running on the DGCP partition.

Important!

Install new software

In case you don't find a software, you can choose to install it by yourself.

- **Install without sudo permissions**
- **Install with conda/pip in conda/virtual env**
  - ➢ **Note:** the official conda repository is no more reachable from Cineca clusters.

    You can rely on conda-forge repository, e.g.

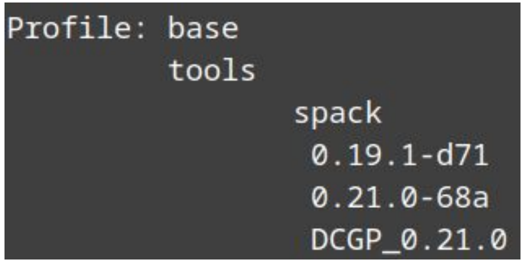
    ```
    $ module load anaconda3/2023.09-0
    $ conda create -y -c conda-forge -n <env_name> --override-channels
    ```
- **Install with Spack**

Write to superc@cineca.it if you need guidance on the installation or if you want to request a new module.

Install with Spack

“Spack” environment provided by the **package manager Spack** and available as **modules**.

$ modmap -m spack



```
Profile: base
         tools
                spack
                 0.19.1-d71
                 0.21.0-68a
                 DCGP_0.21.0
```

for installation of Spack packages for Leonardo Booster (based on **nvhpc** or **gcc** compilers)

for installation of Spack packages for Leonardo DCGP (based on **intel**, **oneapi** or **gcc** compilers)

A new spack version 0.22.2 will be available soon, only one for Booster and DCGP, together with a new software stack

Install with Spack

Load the suitable module for the partition (Booster or DCGP) you will work on.

$ module load spack/<version>

- ➢ **setup-env.sh** file is sourced
- ➢ **$SPACK_ROOT** is initialized
- ➢ **spack command** is added to your PATH, and some nice command line integration tools as well
- ➢ Folder **/spack-<version>** is created into your **$PUBLIC** area (on Leonardo, and $WORK on the other clusters) and it contains some subfolders created and used by spack during the phase of the packages installation:
  - sources cache: **/cache**
  - software installation root: **/install**
  - modulefiles location: **/modules**
  - user scope: **/user_cache**

https://wiki.u-gov.it/confluence/display/SCAIUS/5%3A+User+Environment+Customization

Install with Spack

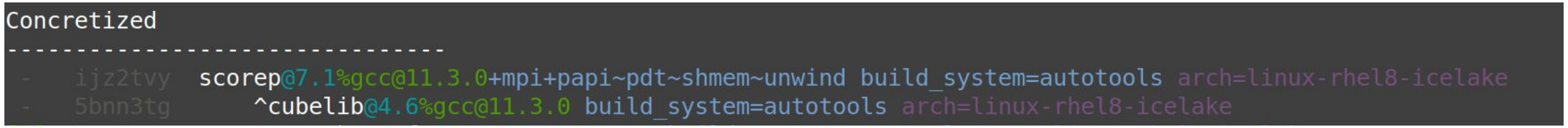
Some fundamental Spack commands

$ spack list <package_name> → Check if the package is available for installation with Spack

$ spack info <package_name> → Show available versions, building variants and dependencies

$ spack spec -Il <package_name> → Show version, compiler, dependencies, building variants with which the package will be installed (-Il for installation status and *hash*)
→ options can be specified

e.g. $ spack spec -Il scorep



```
Concretized
--------------------------------
 -   ijz2tvy  scorep@7.1%gcc@11.3.0+mpi+papi~pdt~shmem~unwind build_system=autotools arch=linux-rhel8-icelake
 -   5bnn3tg      ^cubelib@4.6%gcc@11.3.0 build_system=autotools arch=linux-rhel8-icelake
```

$ spack install <package_name> → Install the package
→ options as spec command

$ spack load <package_name> → Load the package installed to use it (you can also create a **module**)

Outline

LEONARDO
EuroHPC
CINECA
Founded by the European Union
L E O N A R D O

Programming environment

Compilers and MPI libraries are available as modules in profile/base.
Use the ones suitable for the architecture:
on Leonardo DCGP, Intel oneapi compilers and libraries are recommended.

Check with commands
modmap -m,
module av,
module show,
module help,
and **man**

Compilers

- ➢ **GCC** (GNU compilers: gcc, g++, gfortran)
- ➢ **NVHPC** (ex hpc-sdk, ex PGI + CUDA → NVIDIA compilers: nvc, nvc++, nvcc, nvfortran)
- ➢ **CUDA**
- ➢ **INTEL ONEAPI** (**Intel** compilers: icc, icpc, ifort. **Oneapi** compilers: icx, icpx, ifx) → **no** Nvidia GPU support

MPI libraries

- ➢ **OpenMPI** (GNU/NVHPC compilers)
- ➢ **Intel Oneapi MPI** (**Intel** compilers) → **no** CUDA-aware

https://wiki.u-gov.it/confluence/display/SCAIUS/DCGP+Section#DCGPSection-Programmingenvironment

https://wiki.u-gov.it/confluence/display/SCAIUS/DCGP+Section#DCGPSection-MPIenvironment

Update of the software stack

A new spack version will be available soon, only one for Booster and DCGP, together with a new software stack.

Latest versions

- ➢ **Spack 0.21.0** (different modules for Booster and DCGP)
- ➢ Software stack mainly compiled with gcc 12.2, cuda 12.1, nvhpc 23.11, **intel-oneapi-compilers 2023.2.1** (intel 2021.10.0, **oneapi 2023.2.0**)
- ➢ MPI libraries: openmpi 4.1.6, **intel-oneapi-mpi 2021.10.0**

New versions (soon)

- ➢ **Spack 0.22.2** (same module for Booster and DCGP)
- ➢ Software stack mainly compiled with gcc 12.2, cuda 12.2, nvhpc 24.5, **intel-oneapi-compilers 2024.1.0** (intel 2021.10.0, **oneapi 2024.1.0**)
- ➢ MPI libraries: openmpi 4.1.6, hpcx-mpi 2.19, **intel-oneapi-mpi 2021.12.1**

Outline

- ➢ Leonardo infrastructure
- ➢ Access HPC resources and filesystems
- ➢ Software environment
- ➢ Programming environment
- ➢ **Production environment**
- ➢ Final remarks

Login and compute nodes

CINECA HPC clusters are shared among many users, so **a responsible use is crucial!**

Login nodes

- ➢ Interactive runs on login nodes are strongly discouraged and should be limited to short test runs → ***10 minutes cpu-time limit***
- ➢ Avoid running large and parallel applications on login nodes
- ➢ ***No GPUs on login nodes***

Compute nodes

- ➢ Long production jobs should be submitted on compute nodes using the **scheduler** → **SLURM**
- ➢ Jobs can be submitted in two main ways: via **batch mode** and via **interactive mode**
- ➢ **Nodes shared**, but the allocated resources (cores, RAM, $TMPDIR) are assigned in an exclusive way

Resources per node

Each node → max resources you can request per node

- ➢ 112 cores (**cpus**) ⟶ **ntasks-per-node * cpus-per-tasks ≤ 112**
- ➢ 494000 MB of RAM (**memory**)
- ➢ 3 TB of temporary local memory on $TMPDIR (**gres=tmpfs**)

⇨ The **accounting** considers

- the requested number of CPUs
- the requested memory on RAM
- the requested memory on $TMPDIR

and calculates the **number of equivalent cores** → it takes the **maximum** among

- number of cpus
- memory / memory-per-core (= requested memory / memory-per-node * cores-per-node)
- tmpfs / tmpfs-per-core (= requested tmpfs / tmpfs-per-node * cores-per-node)

Eurofusion resources

Serial partition → **lrd_all_serial** (default, free)

2 dedicated *login-type* nodes

- max 4 physical cores (hyperthreading x2: max 8 virtual cpus)
- max walltime: 4 h

Production partition → **dcgp_fua_prod**

258 compute nodes dinamically allocated

- max 16 nodes
- max walltime: 24 h

Big production QOS: **dcgp_qos_fuabprod**

- min 17 full nodes - max 64 nodes
- max walltime: 24 h

Debug partition → **dcgp_fua_dbg**

2 compute nodes dinamically allocated

- max 2 nodes
- max walltime: 10 min

Eurofusion resources

Additional options

Low priority qos: **qos_fualowprio**

- max 16 nodes
- max walltime: 8 h
- automatically added to the active accounts with exhausted budget (free, zero queue priority)

Low priority account: **FUA38_LOWPRIO_0**

- for active projects with non-exhausted budget, after request to superc@cineca.it
- you also need to add the qos qos_fualowprio

Special qos: **qos_special**

- if needed more than 64 nodes and/or 24h
- after request to superc@cineca.it and EF approval

Submit jobs with SLURM

Batch mode

- Write a batch script like the example
- Launch the batch script
  $ sbatch [options] start.sh
- The job is queued and scheduled

shell →

#SBATCH directives →
(also contracted syntax,
e.g. -N for --nodes)

Loading modules and setting variables →

Launch executable →
(for parallel applications, use **srun** or **mpirun**)

```
#!/bin/bash

#SBATCH --nodes=1                          # nodes
#SBATCH --ntasks-per-node=4                # tasks per node
#SBATCH --cpus-per-task=8                  # cores per task
#SBATCH --mem=494000MB                     # memory on RAM
#SBATCH --gres=tmpfs:200GB                 # memory on $TMPDIR
#SBATCH --time=1:00:00                     # time limit (d-hh:mm:ss)
#SBATCH --account=<account_name>           # account
#SBATCH --partition=<partition_name>       # partition name
#SBATCH --qos=<qos_name>                   # quality of service

module load <module_name>

 srun my_application
```

Submit jobs with SLURM

Interactive mode

- Ask for the needed resources with the same **SLURM directives** with srun or salloc
- The job is queued and scheduled but, when executed, the standard input, output, and error streams are connected to the **terminal session** from which srun or salloc were launched
- **Run your application from that prompt**
- Exit from the terminal session: **$ exit**

Non MPI programs

```
$ srun -N 1 --ntasks-per-node=8  --cpus-per-task=4 -t 01:00:00
-p <partition_name> -A <account_name> --pty /bin/bash
```

The session starts on the compute node: **[username@lrdn4553 ~]$**

Also **MPI programs**

```
$ salloc -N 1 --ntasks-per-node=8  --cpus-per-task=4 -t 01:00:00
-p <partition_name> -A <account_name>
```

A new session starts on the login node: **[username@login02 ~]$**

Submit jobs with SLURM

Only on Leonardo “diskful” nodes, it’s possible to increase the space of the **$TMPDIR** area.
Remind that the area is **local to nodes**, and **job specific** (i.e. “temporary”): created at the begging of a job and deleted at its end, and accessible only by the user who launched the job.

Specify the space on $TMPDIR=/tmp (default=10GB):

```
#SBATCH --gres=tmpfs:200GB
```

on the local disks on **lrd_all_serial** nodes (max 1 TB) and **dcgp_usr_prod** compute nodes (max 3 TB).

It is possible to use the $TMPDIR=/scratch_local space also on the **login** nodes (14 TB shared among users, remove your files once they are not requested anymore).

On the diskless **boost_usr_prod** compute nodes, the $TMPDIR=/tmp area is hosted on the RAM, with a fixed size of 10 GB (no increase is allowed, and the gres=tmpfs resource is disabled).

Remind that for the DCGP jobs the requested amount of gres=tmpfs resource contributes to the consumed budget, changing the number of accounted equivalent core hours.

https://wiki.u-gov.it/confluence/display/SCAIUS/LEONARDO+User+Guide#LEONARDOUserGuide-DisksandFilesystems

Submit jobs with SLURM

#SBATCH --account=<account_name> or **-A <account_name>**

Specifies the account with a **budget** of core-hours available to run jobs.

Note that the account name changed from Marconi to Leonardo DCGP, with the addition of a final "**_0**"

Remind also that, on Leonardo, you can check the status of your accounts with

$ saldo -b $\longrightarrow$ **Leonardo Booster**

$ saldo -b --dcgp $\longrightarrow$ **Leonardo DCGP**

Accounts defined on Booster can only be used on **Booster partitions** (boost_fua_prod, boost_fua_dbg), and accounts defined on DCGP can only be used on **DCGP partitions** (dcgp_fua_prod, dcgp_fua_dbg)**.**

Monitor your jobs with SLURM

$ squeue -u <username> or **$ squeue --me**

Shows the list of all your scheduled jobs, along with their status (pending, running, closing, …).
Also, shows you the **jobID** required for other SLURM commands.

$ scontrol show job <job_id>

Provides a long list of informations for the job requested.
In particular, if your job isn't running yet, you'll be notified about the reason it has not started yet and, if it is scheduled with top priority, you will get an **estimated start time**.

$ scancel <job_id>

Removes the job (queued or running) from the scheduled job list by killing it.

$ sinfo (e.g. **$ sinfo -o "%10D %a %20F %P"**)

Provides information about SLURM nodes and partitions.

$ sacct <options> <job_id> (e.g. **$ sacct -Bj <job_id>**)

Displays accounting data for all jobs and job steps in the SLURM job accounting log or SLURM database.

Outline

- ➢ Leonardo infrastructure
- ➢ Access HPC resources and filesystems
- ➢ Software environment
- ➢ Programming environment
- ➢ Production environment
- ➢ **Final remarks**

Final remarks

- ★ **Login nodes** should only be used for installation (connection to external network!), compilation, and small tests. **No** GPUs on login nodes!
- ★ Consider to use **Leonardo Booster for applications on GPUs** and **Leonardo DCGP for applications only on CPUs.**
- ★ Remind to check your accounts budget with "**saldo -b --dcgp**" on Leonardo DCGP.
- ★ Recommended **compilers** are gcc and Nvidia compilers (nvhpc, cuda) for Leonardo Booster, and gcc and Intel (intel, oneapi) for Leonardo DCGP.
- ★ Rely on the already available **software stack**, tested and optimized for the cluster architecture, and on **Spack** for autonomously installing additional software.

https://wiki.u-gov.it/confluence/display/SCAIUS/HPC+at+CINECA%3A+User+Documentation

Write to superc@cineca.it in case of need!

LEONARDO
EuroHPC Joint Undertaking
CINECA
Founded by the European Union
L E O N A R D

Thank you

Q&A

1.

Q: As far as I know, EUROfusion users are accounted with node hours instead of core hours. Can you please explain how are we accounted in Leonardo DCGP?

A: Both on Marconi and Leonardo the accounting is in terms of **core-hours**.
The difference is that on Marconi the nodes were allocated in exclusive way by default, so even if you requested less than an entire node, you consumed as the entire node. On Leonardo instead (both Booster and DCGP), the nodes are in principle shared with other users, they are not allocated in exclusive way by default (you can specify "**--exclusive**"), so **you consume the core-hours equivalent to the requested resources**.

<u>See slide 28.</u>
For example, if you request 56 cores per one hour on dcgp_usr_fuaprod partition, you only consume 56 core-hours.
If you request only 2 cores, but half of the space per node in tmpfs, i.e. 1.5 TB, you still consume as 56 cores (half of the node).

Q&A

2.

Q: What best to use for I/O? Fast scratch, scratch or tmpdir?

A: It depends on your application. The **WORK** and **SCRATCH** areas (on HDD) can always be used, but if your application take relevant advantage from a higher I/O performance, you can use the **FAST** area (on SSD). **TMPIDIR** also assure a fast communication because it is hosted on disk which is local to the node. Moreover, in case of DCGP compute nodes, the local disk is also SSD (while on login-type nodes the local disk is HDD, and on Booster compute nodes there is no local disk and TMPDIR space is hosted on RAM). Remind that $TMPDIR directory is created at the beginning of the job by the SLURM prologue, and deleted at the end of the job by the SLURM epilogue.

Q&A

3.

Q: What to know for porting from Marconi to Leonardo a production code that has been running for years without problems on Marconi and other computers?

A: From the point of view of the data transfer from Marconi to Leonardo, we suggest to exploit the **datamover** service: https://wiki.u-gov.it/confluence/display/SCAIUS/Datamover
From the point of view of the compilation and execution of the code, we remind that Marconi and Leonardo DCGP have both Intel processors. On Leonardo DCGP we suggest to rely on **Intel/Oneapi compilers**, both for the compilation and for the libraries/applications/tools you need. In case you face errors and you need some help in finding the right software stack, write to superc@cineca.it.

Q&A

4.

Q: Can I use openmpi if oneapi does not work for me?

A: Yes, you can. In general, **GNU compilers** are suitable also for the DCGP partition. We suggest to use **Intel/Oneapi** if possible, because they may offer better performance, since they are optimized for Intel architecture (CPUs in this case).

Q&A

5.

Q: What is the difference between asking for ram and for tmpdir?

A: **$TMPDIR** space represents a proper storage space (similarly to WORK, SCRATCH, FAST, HOME), to be used only during the job.

Q&A

6.

Q: Is there a default for “$SBATCH --gres=tmpfs:” ?
If no tmpdir space is needed, will it not be accounted for?

A: The **default** is 10 GB.
The resource tmpfs is **accounted** as well, but beeing the 10 GB less than 1/112 of the total amount of tmpfs on DCGP compute nodes (which is 3TB), if you ask for 1 core for one hour, for example, you will be carged for 1 core-hour. Otherwise, if you ask for 3 TB and 1 core, this will be the same of asking for the full nodes, meaning that you will be accounted for 112 core-hours.

Q&A

7.

Q: In case if we need more RAM memory per process, in the past we were able to request half of the MPI processes/node and use the complete node RAM, how is this handled in Leonardo?

A: This is possible on Leonardo as well. You can request indeed, for example, half of the cores per node (56 cores on DCGP compute nodes) and the whole RAM of the node (494000 MB). The **accounting** will consider that you are using the entire node, so the equivalent of 112 cores will be accounted.

Q&A

8.

Q: How do I ask all the memory?

A: You can explicitly request the **maximum RAM**, for example
#SBATCH --mem=494000
(MB) on DCGP compute nodes, or
#SBATCH --mem=0
which means the maximum possible on the node.