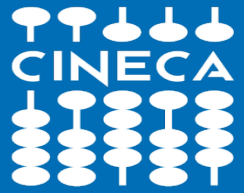
Introduction to the CINECA Marconi100 HPC system

TREX Hackathon III
March 06-08 2023

Diego Molinari
d.molinari@cineca.it

SuperComputing Applications and Innovations (SCAI) – High Performance Computing Dept



WELCOME TO CINECA

2022 OVERVIEW

HPC SYSTEMS

CINECA

CINECA enables world-class scientific research by operating and supporting leading-edge supercomputing technologies and by managing a state-of-the-art and effective environment for the different scientific communities.

LEONARDO | 2022

4992 nodes
Booster Module:
32 core per node
4 GPU NVidia Ampere custom
Data Centric Module:
56 cores per node
110 PB Storage
250 PFlops

SOON IN PRODUCTION

MARCONI | 2016

3188 nodes
48 cores per node
612 TB RAM
10 PFlops

MARCONI100 | 2020

980 nodes
32 cores per node
4 GPU Nvidia V100 per node
8 PB Storage
32 PFlops

DGX | 2021

3 nodes
128 cores per node
8 GPU NVIDIA A100 per node
100 TB Storage
15 PFlops

GALILEO100 | 2021

564 nodes
48 cores per node
2 GPU NVIDIA V100 per node
~22 PB Storage
2 PFlops

M100 Infrastructure: how to access

```
$ ssh -X username@login.m100.cineca.it
**********************************************************************
**
* Welcome to MARCONI100 Cluster /
*
*           IBM Power AC922 (Whiterspoon) -
*
*           Red Hat Enterprise Linux Server release 8.1 (Ootpa)
*
etc. etc.
```

- Short system description
- “In evidence” messages
- “Important messages” (changes of policies, maintenances, etc.)

Access by **public keys** (with the ssh keys generated on a local and **secure** environment, and protected via passphrase) is strongly recommended

IMPORTANT

Should the Hackathon activities (i.e., compilation) affect the M100 login nodes – it may happen – we will devote a login node to participants. Stay tuned!

M100 Infrastructure: how to access

$ ssh -X username@login.m100.cineca.it
****

You will receive personal username and password to connect to M100 for the Hackathon event:

Username: **a08traXX**
Password: **sent by email**
Account: **tra23_hackath** (needed to submit jobs to the queueing system: SLURM)

We also set up a set of reserved nodes for you. They are available using the following reservation:

Reservation: **s_tra_hackath** (valid from 2023-03-06 at 9:00 up to 2023-03-08 at 18:00)

There are 10 nodes in the reservation. If you realize you need more let us know.

At the first login, please **change the password** (you should be forced by the system to do that).
Password Policy
The new password has to be 10 characters long and contains at least 1 capital letter, 1 number, and 1 special character (!"#$%&'()*+,-./:;<=>?@[\]^_`{|}~)

Marconi100: the Power AC922 model

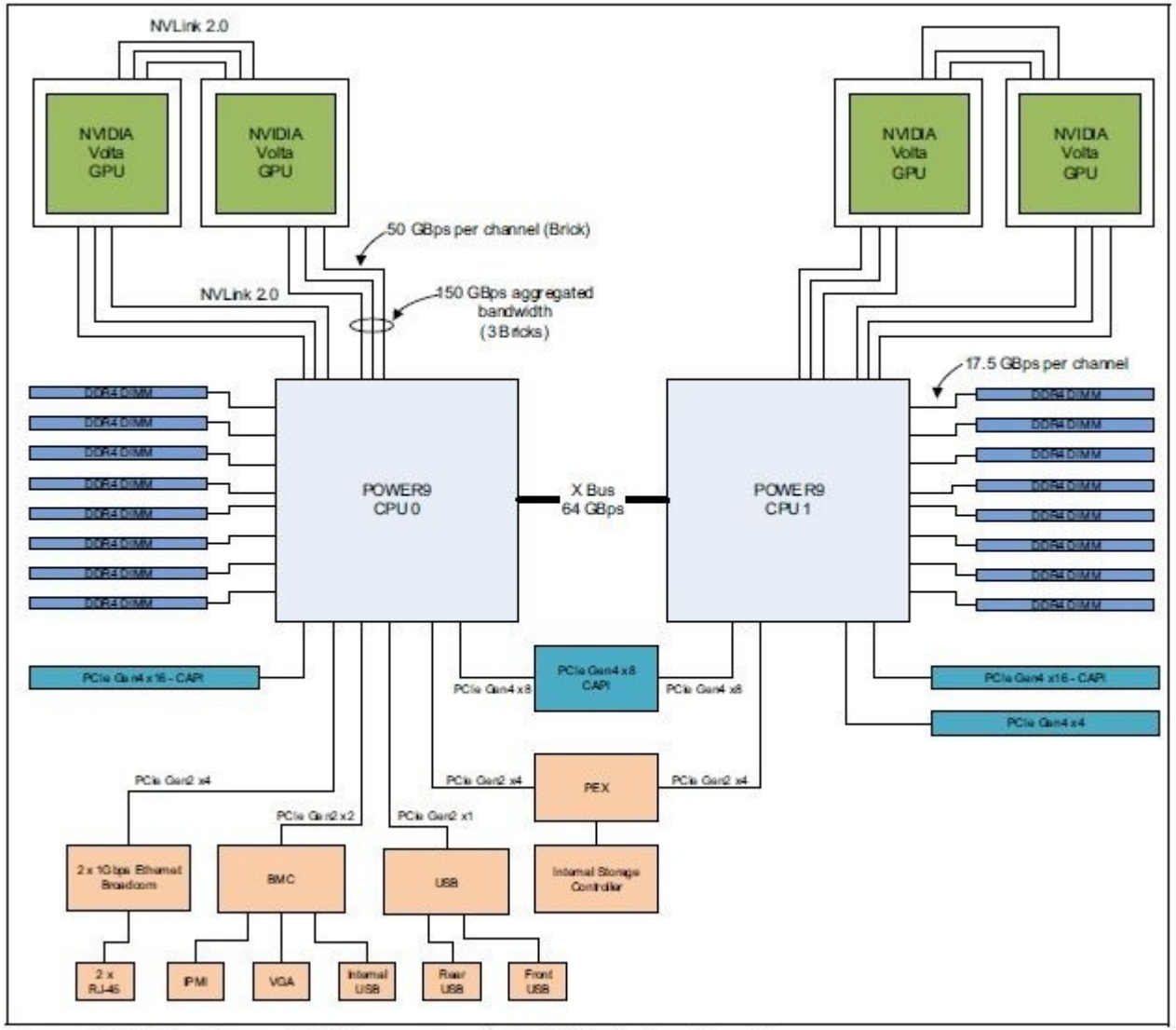
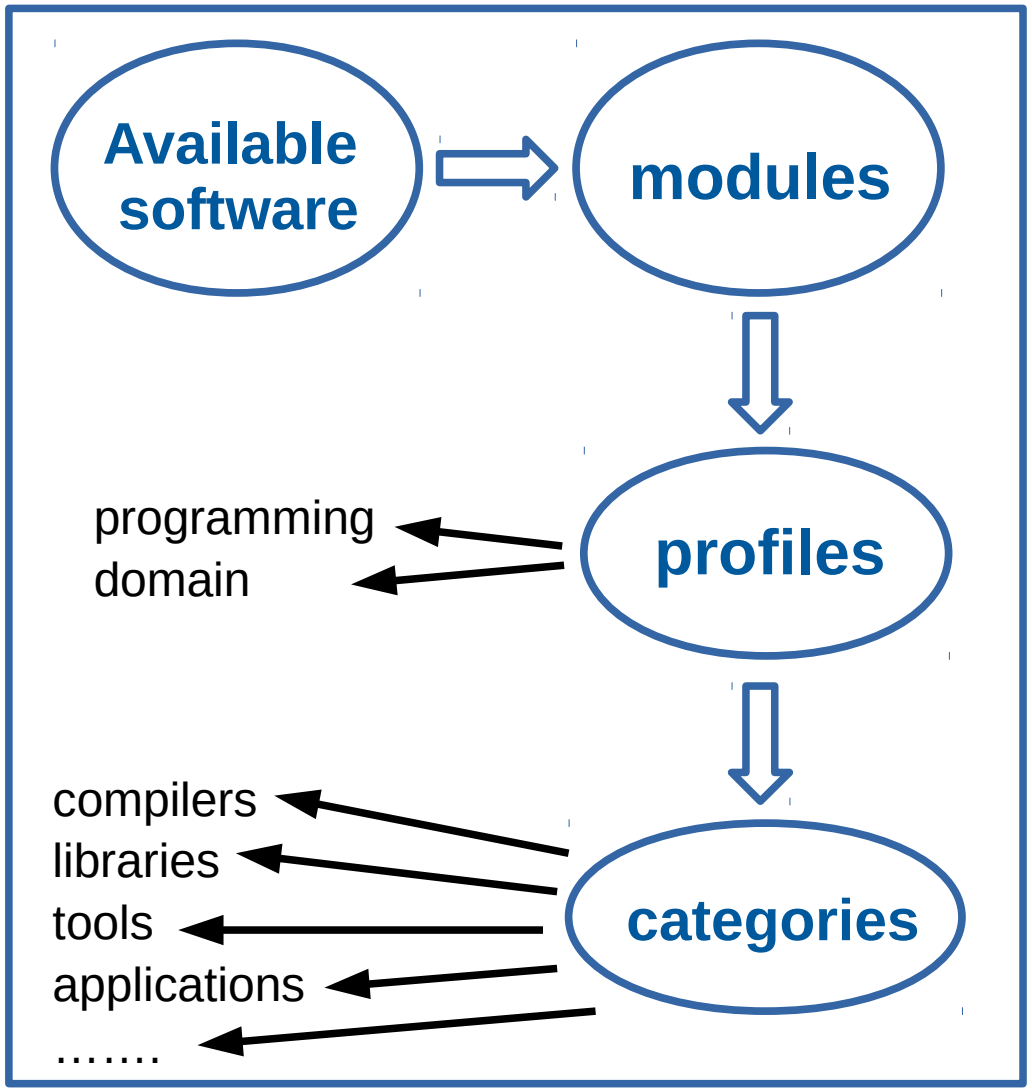


Figure 2-5 The Power AC922 server model GTH logical system diagram

- AC922 “Whiterspoon”
- **32 PFlops peak (9th on Top500 June 2020)**
- Nodes: 980 compute + 4 login nodes + 2 (for containers et all), 32 TFlops each
- Processors: 2x16 cores IBM 8335-GTG 2.6 (3.1) GHz
- Accelerators: **4xNVIDIA V100 GPUs**, Nvlink 2.0, 16GB
- RAM: 256 GB/node
- Local disk: 1.6TB NVMe
- Internal Network: Mellanox Infiniband EDR DragonFly+
- Disk Space: 8PB storage

M100 Module Software Environment



The available software is offered in a module environment

The modules are collected in different profiles and organized in functional categories

Profile types:

- Programming (**base**, advanced): compilation, debugging, profiling, libraries
- Domain (chem-phys, lifesc, …): production activities

The **base** profile is the default:

- automatically loaded after login
- contains basic modules for the programming activities

M100 Module Software Env: base

$ module av

---------- /cineca/prod/opt/modulefiles/profiles ----------
profile/advanced **profile/base** profile/chem-phys profile/bioinf profile/archive profile/candidate profile/deeplrn profile/lifesc
……..
---------- /cineca/prod/opt/modulefiles/**base**/environment ----------
autoload

---------- /cineca/prod/opt/modulefiles/**base**/libraries ----------
blas/3.8.0--gnu--8.4.0 szip/2.1.1--gnu--8.4.0 boost/1.76.0—spectrum_mpi--10.4.0--binary zlib/1.2.11--gnu--8.4.0
elsi/2.5.0--gnu--8.4.0 essl/6.2.1--binary ……..

---------- /cineca/prod/opt/modulefiles/**base**/compilers ----------
cuda/11.3 gnu/8.4.0 hpc-sdk/2022--binary python/3.7.7 python/3.8.2 spectrum_mpi/10.4.0--binary xl/16.1.1--binary

---------- /cineca/prod/opt/modulefiles/**base**/tools ----------
anaconda/2020.11 cmake/3.20.0 singularity/3.9.7 spack/0.14.2-prod

M100 Module Software Env: domains

“Domain” profiles:

```
---------------------------------------------------------- /cineca/prod/opt/modulefiles/profiles ----------------------------------------------------------
profile/advanced  profile/archive  profile/base  profile/candidate  profile/chem-phys  profile/deeplrn  profile/bioinf  profile/lifesc
```

To access a “domain” application, *e.g.* in the chemical physics scientific domain, you need to load the profile/chem-phys first:

```
$ module load profile/chem-phys
```

The profile chem-phys is **added** to the base profile

The domain profiles are all “additive”: you can load them together, adding them to the base profile

M100 Module Software Env:
autoload, modmap

Needing, *e.g.*, lammps?

```
$ module load profile/chem-phys
$ module load autoload lammps/22dec2022
$ module list
Currently Loaded Modulefiles:
 1) profile/base          4) spectrum_mpi/10.4.0--binary   7) cuda/11.0                      10) lammps/22dec2022
 2) profile/chem-phys     5) gnu/8.4.0                     8) lapack/3.9.0--gnu--8.4.0
 3) autoload              6) blas/3.8.0--gnu--8.4.0        9) fftw/3.3.8—spectrum_mpi—10.4.0
```

The **autoload** module takes care to load all the lammps dependencies

A better, easier way to know if an application is available on M100?

The **modmap** command!

```
$ modmap -m lammps
Profile: advanced
Profile: archive
Profile: base
Profile: chem-phys
                lammps
                 22dec2022
Profile: deeplrn
Profile: lifesc
```

modmap detects all the available profiles, categories, and modules
=> “map” of the available modules

modmap -h # command help

M100 Module Software Env:
spack

$ module load spack/0.14.2-prod

- setup-env.sh file is sourced, $SPACK_ROOT is initialized to /cineca/prod/opt/tools/spack/<vers>/none, spack command is added to your PATH, and some nice command line integration tools too.

- A folder is created into your default $WORK space ($USER/spack-<vers>) with the subfolders created and used by spack during the phase of a package installation:

  - sources cache: $WORK/$USER/spack-<vers>/cache
  - software installation root: $WORK/$USER/spack-<vers>/install
  - module files location: $WORK/$USER/spack-<vers>/modulefiles

- You can define different paths for cache, installation and modules directories (please refer to the spack guide to find out how to customize these paths)

- Some softwares installed with spack are already available as modules or as spack packages:

$ module load spack/0.14.2-prod	or	$ module load spack/0.14.2-prod
$ module av		$ spack find
$ module av <module _name>		$ spack find <package_name>

Can't you find the software you need? Use the "**spack**" environment

M100 Programming Environment

Available compilers in BASE profile:

IBM XL C/C++ and Fortran	16.1.1
gnu	8.4.0/10.3.0
hpc-sdk	2020/2021/2022
cuda	11.0 → 11.3

hwloc provides details about NUMA memory nodes, sockets, shared caches, cores and SMT, etc.

In addition:
- a gnu compiled **Open MPI 4.0.3** and **4.1.2** installations are available in **profile/advanced**
- more recent compilers (i.e. cuda/11.6 or hpc-sdk/2023 version 23.1) are available in **profile/candidate**

Available MPI environment in BASE profile:

IBM Spectrum MPI 10.4.0

- Based on Open MPI version 4.0.5, full MPI 3.2 standard
- FCA (hcoll) support (Mellanox Fabric Collective Accelerator on InfiniBand interconnect)
- Relies on hwloc to navigate the server hardware topology
- GPU support
  - NVIDIA GPUDirect RDMA
  - CUDA-aware MPI

Use **mpirun** (not srun, work in progress) to execute your MPI program
By default, GPUDirect support is disabled.
Run the "**mpirun -gpu**" command to enable it.

Use the --report-bindings option for an abbreviated image of the server's hardware and the binding of processes

M100 data areas

Login and Compute nodes

- $HOME (personal, under back-up, shared GSS over IB)
- $CINECA_SCRATCH (personal, no back-up, periodic cleaning for files older than 40 days, shared GSS)
- $WORK: common project area (/m100_work/tra23_hackath), 1 TB of quote, no back-up, available in the validity period of project + 6 months, shared GSS.
  Useful to share data among Project members (it is accessible by ALL Teams).
- /scratch_local (1TB NVMe), local to nodes, not writable on compute nodes

Compute nodes

- SLURM job TMPDIR: /scratch_local/slurm_job.<jobid> (1TB NVMe for the scratch_local), local to nodes, created by slurmd prolog at the start of the job and removed at the end of the job

M100 Environment

A RESERVATION ON **10 NODES** on the partition **m100_sys_test**
IS DEFINED from March, 6th up to March,, 8th

s_tra_hackath

In principle **1 node per participant**: PLEASE LET US KNOW IF YOUR ACTIVITY WOULD BENEFIT OF MORE THAN ONE NODE. We can increase the number of nodes in the reservation if needed.

THE RESERVATION IS AVAILABLE TO HACKATHON'S TRAINING USERNAMES

FORGOT IT? Write to us!

```
#!/bin/bash
...
#SBATCH -p m100_sys_test                  # defines the partition
#SBATCH -q qos_test                       # needed to access the m100_sys_test partition
#SBATCH --reservation=s_tra_hackath       # to access the reserved nodes
...
```

You can in principle use the production partition (m100_usr_prod), but you may end to wait because of a **long queue**.

M100 Production Environment

M100 is a general purpose system used by hundreds of users.

Compute nodes

- Production jobs must be submitted to M100 queueing system: batch jobs
- SLURM scheduler and resource manager
- Node sharing (but the allocated resources – cores, gpus, memory – are assigned in an exclusive way)

Login nodes

A responsible use of the login nodes is crucial to ensuure the effective use of the infrastructure and the access to the computing resources.

- Protect your credentials and access from “safe” posts; opt for ssh keys with passphrase
- Interactive runs on login nodes are strongly discouraged and should be limited to short test runs
  - Per user limits on cpu-time (10 minutes) and memory (1 GB)
  - Avoid running large parallel applications on the front-ends.
- The variable TMPDIR is defined for all users to /scratch_local
  You can re-define it to $CINECA_SCRATCH or other areas.
  PLEASE DO NOT SET it to /tmp → critical!

M100 Production Environment
SLURM specs and Accounting

Each node “exposes itself” as having

- 128 (virtual) cpus [32 physical cores with 4 Hts each]
- 4 GPUs
- 246000 MB of memory

It is possible to ask up to
- 128 ntasks-per-node (1 cpus-per-task)
- 1 ntask-per-node (128 cpus-per-task)
- Or any combination of ntasks-per-node * cpus-per-task ≤ 128

BUT

SLURM has been configured so to assign a physical core with its 4 Hts

Asking for –ntasks-per-node=1 and –cpus-pet-task=1 corresponds to ask for --cpus-per-task=4

The accounting considers:

- The requested number of physical cpus
- The requested number of GPUs
- The amount of memory

And calculates the number of equivalent cores taking the maximum among

- N physical cpus
- N GPUs * 8
- Memory / Memory-per-core

M100 Production Environment
Interactive batch jobs

In case you need to “interact” with your running job (tuning of input parameters, debugging, etc.)
And it needs more than 10 minutes, or many processes (not suitable on the login nodes)

“Interactive” SLURM batch job

- Ask for the needed resources (cores, gpus, memory, time) with srun or salloc
- The job is queued and scheduled as any other job, but, when executed, the standard input, output and error streams are connected to the reminal session from which srun or salloc were launched
- You can then run your application from the terminal

NON MPI programs (single process or multi-threaded programs using one or more GPUs)

$srun <options> --pty /bin/bash

The session starts on the compute node (look at the prompt!)

MPI programs using one or more GPUs

$salloc <options>

A new session is started on the login node
Remember to exit the session when you have finished.

M100 Production Environment
Interactive batch jobs for compilation

In case you need to **compile the code on a compute node**

“Interactive” SLURM batch job

- Ask for the minimal resources needed for the compilation with **salloc**
- When the job starts you can **ssh to the compute node** SLURM has granted you
- You can then compile your code
- Remember that you have set a **walltime** for the job

```
[...@login02] $ salloc -N1 -n1 -t 02:00:00 -A tra23_hackath
                       -p m100_sys_test -q qos_test --reservation=s_tra_hackath
salloc: Pending job allocation 6802903
salloc: job 6802903 queued and waiting for resources
salloc: job 6802903 has been allocated resources
salloc: Granted job allocation 6802903
salloc: Waiting for resource configuration
salloc: Nodes r206n11 are ready for job

[...@login02] $ ssh r206n11
[...@r206n11] $ nvcc …..

When you have finished remember to exit twice to stop the running job
[...@r206n11] $ exit
[...@login02] $ exit
salloc: Relinquishing job allocation 6802903
salloc: Job allocation 6802903 has been revoked.
[...@login02] $
```

M100 Production Environment
Non interactive batch jobs

As usual on HPC systems, the large production runs are executed in batch mode.
The user writes a list of the needed #SBATCH directives (resources, walltime, mail, jobname, etc. etc.) followed by the needed loading of modules, setting of variables, and launch of the executable.

```
#!/bin/bash
#SBATCH --nodes=1                 # Number of nodes
#SBATCH --ntasks-per-node=4       # Number of MPI ranks per node
#SBATCH --ntasks-per-socket=2     # Number of MPI ranks per socket
#SBATCH --cpus-per-task=32        # number of HW threads per task
#SBATCH --gres=gpu:4              # Number of requested gpus per node, can vary between 1 and 4
#SBATCH --mem=230000MB            # Memory per node
#SBATCH --time 00:30:00           # Walltime, format: HH:MM:SS (max 24 hours)
#SBATCH -A tra23_hackath
#SBATCH -p m100_sys_test
#SBATCH -q qos_test
#SBATCH --reservation=s_tra_hackath

module load profile/chem-phys
Module load autoload yambo/4.5

export OMP_NUM_THREADS=$SLURM_CPUS_PER_TASK

mpirun –map-by socket:PE=8 –rank-by core -np ${SLURM_NTASK} yambo -F yambo.in -J yambo.out
```

M100 Production Environment
Profiling your code with Nvidia Nsight system

In the previous edition, the usage of the Nvidia profiler caused several compute node crashes due to the usage of the /tmp area of the node by the profiler itself.
In order to avoid such a problem, we suggest to **modify the sbatch script** as in the following example.

```
#!/bin/bash
#SBATCH directives
#SBATCH --exclusive        #to avoid superposition of profiling jobs on the same node

module load hpc-sdk
module load ……

export OMP_NUM_THREADS=$SLURM_CPUS_PER_TASK

rm -rf /tmp/nvidia
ln -s $TMPDIR /tmp/nvidia
mpirun ..... nsys profile ....
rm -rf /tmp/nvidia
```

In the worst case the node does not crash when filling the /tmp area, but the job simply stop producing output, but still running with a consequence lost of time and cpu-hours.

We kindly ask to follow these suggestions

IMPORTANT:
on M100 Nvidia Nsight system **GUI is not supported**.
Please run the profiler via **command line**, then you can download the .qdrep result on your **local PC for visualization**

M100 HELP!

- We will be around for any kind of support needed!
- Ask superc@cineca.it (during and after hackathon. PLEASE: mention TREX Hackathon 2023 in the subject)
- Online guide: https://wiki.u-gov.it/confluence/display/SCAIUS/UG3.2%3A+MARCONI100+UserGuide

ENJOY TREX Hackathon III @ CINECA

from CINECA User Support Team!