SCtrain | SUPERCOMPUTING KNOWLEDGE PARTNERSHIP

Introduction to HPC I

Domitilla Brandoni, CINECA

d.brandoni@cineca.it

Univerza v Ljubljani

TU WIEN TECHNISCHE UNIVERSITÄT WIEN

CINECA

VSB TECHNICAL UNIVERSITY OF OSTRAVA | IT4INNOVATIONS NATIONAL SUPERCOMPUTING CENTER

Co-funded by the Erasmus+ Programme of the European Union

This project has been funded with support from the European Commission.
This publication [communication] reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

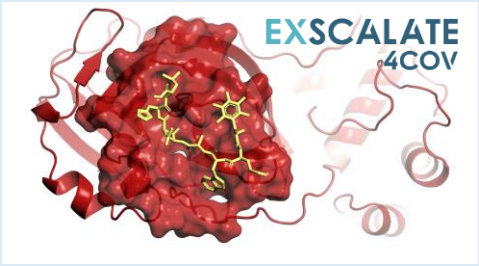
The ability of processing a huge amount of data

and performing complex calculation at high speed

1

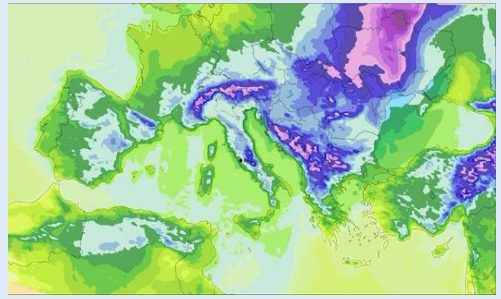
Problem dimensionality

Huge amount of data



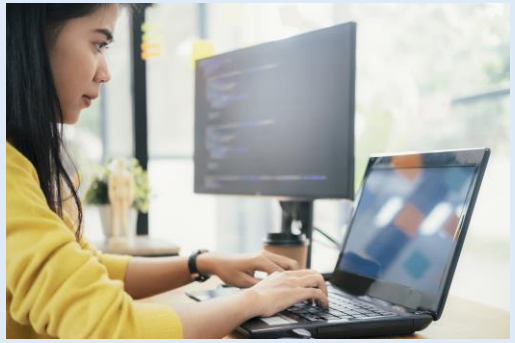
2

Time to solution

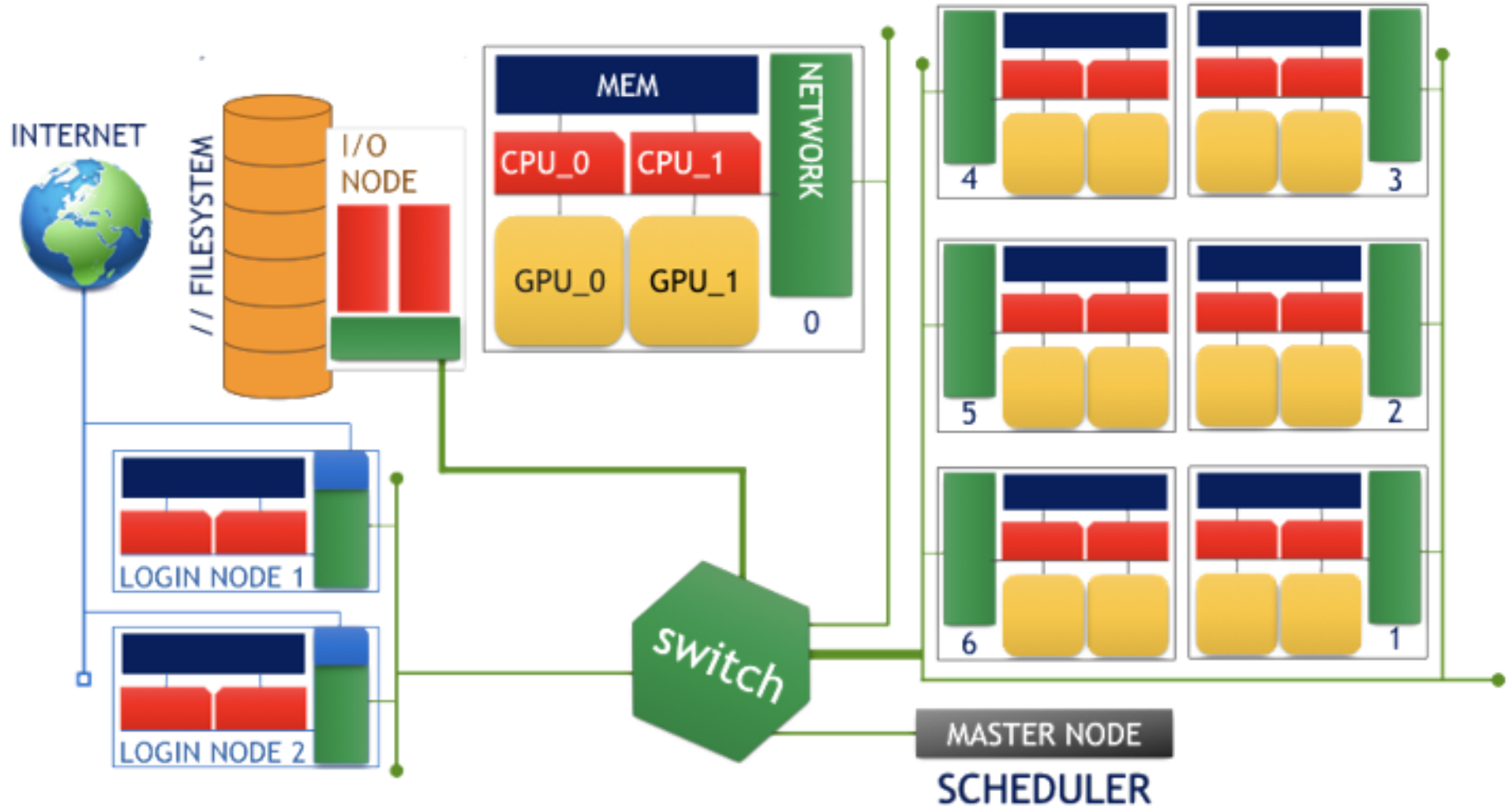



WHY?

HPC and supercomputers



Supercomputer architecture

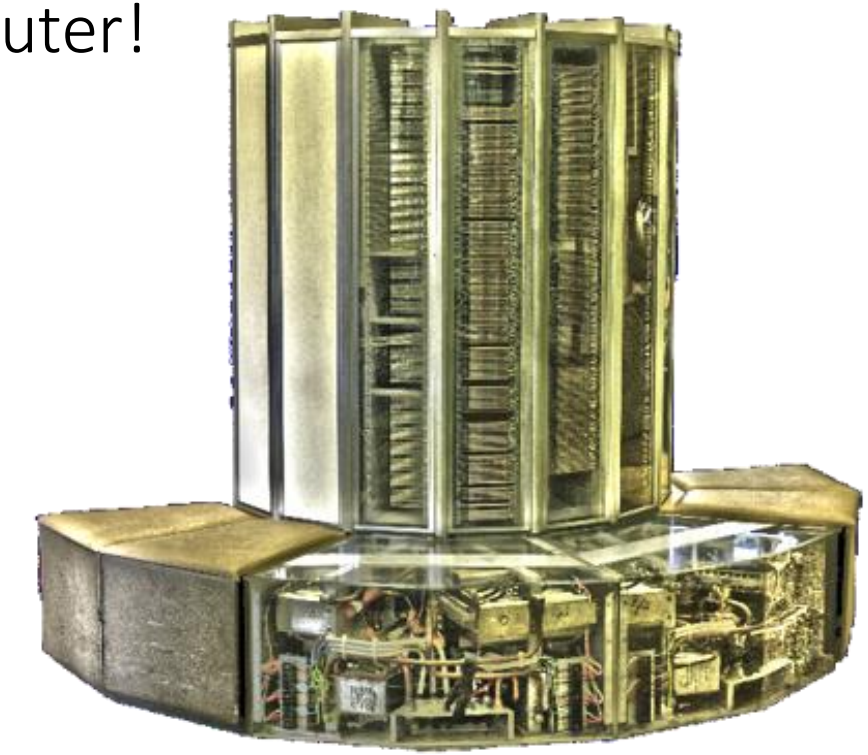


Questions?

HPC evolution

How it started

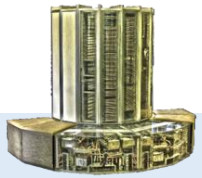
- 1976 –1982 the most powerful supercomputer!
- 135 MegaFLOPS
- nuclear weapons design
- 80 MHz
- 8.000.000 $



CRAY I

How it's going

1970



Cray I

i. 135 MegaFLOPS
ii. 80 MHz
iii. 8.000.000 $

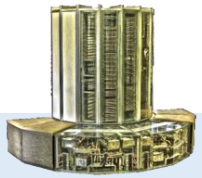
2022

Processor A8

i. 1,5 GigaFLOPS
ii. 1,4 GHz
iii. 439 € (iPad mini)

How it's going

1970



[Cray I]

i. 135 MegaFLOPS
ii. 80 MHz
iii. 8.000.000 $

2018

NVIDIA

Nvidia Tesla V100
i. 7 TeraFLOP
ii. ~5000 CUDA core
iii. 32 GB VRAM
iv. 10000€

intel

Intel Skylake Xeon 8160
i. 1.5 TeraFLOP
ii. 24 core
iii. 2.30 GHz
iv. 4700€

CINECA Clusters

Marconi
3188 nodes
Intel SKL 2x24 cores, 2.1 GHz
192 GB RAM/CN
Intel Omnipath

Marconi100
980 nodes
IBM POWER 9 2x16 cores (SMT 8)
4 NVIDIA Volta V100
256 GB RAM/CN
Mellanox EDR

ADA CLOUD
77 nodes
Intel Cascade Lake 8260 2x24 cores, 2.4 GHz
768 GB RAM
100Gbs Ethernet interconnection

GALILEO 100
528 nodes
Intel Cascade Lake 8260 2x24 cores, 2.4 GHz
384 GB RAM
100Gbs Infiniband interconnection

DGX
3 nodes
AMD Rome Cascade Lake 8260 2x32 cores,
2.6 GHz, 8 NVIDAI A100, NVLINK
980 GB RAM
100Gbs Ethernet interconnection

How to benchmark supercomputers

SCtrain | SUPERCOMPUTING KNOWLEDGE PARTNERSHIP

LINPACK

solve a linear system A x = b with a dense random matrix A

8	**HPC5** - PowerEdge C4140, Xeon Gold 6252 24C 2.1GHz, NVIDIA Tesla V100, Mellanox HDR Infiniband, **Dell EMC** Eni S.p.A. **Italy**	669,760	35,450.0	51,720.8	2,252
9	**Frontera** - Dell C6420, Xeon Platinum 8280 28C 2.7GHz, Mellanox InfiniBand HDR, **Dell EMC** Texas Advanced Computing Center/Univ. of Texas **United States**	448,448	23,516.4	38,745.9	
10	**Dammam-7** - Cray CS-Storm, Xeon Gold 6248 20C 2.5GHz, NVIDIA Tesla V100 SXM2, InfiniBand HDR 100, **HPE** Saudi Aramco **Saudi Arabia**	672,520	22,400.0	55,423.6	
11	**Marconi-100** - IBM Power System AC922, IBM POWER9 16C 3GHz, Nvidia Volta V100, Dual-rail Mellanox EDR Infiniband, **IBM** CINECA **Italy**	347,776	21,640.0	29,354.0	1,476

TOP 500 The List.

M100 on top500.org

June2021 — November2021 — June2022

14 — **18** — **21**

Coming soon... November 2022

High Performance Conjugate Gradient Method

solve a linear system A x = b with a sparse matrix A

Rank	TOP500 Rank	System	Cores	Rmax (TFlop/s)	HPCG (TFlop/s)
1	1	**Summit** - IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband , **IBM** DOE/SC/Oak Ridge National Laboratory United States	2,414,592	148,600.0	2925.75
2	2	**Sierra** - IBM Power System AC922, IBM POWER9 22C 3.1GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband , **IBM / NVIDIA / Mellanox** DOE/NNSA/LLNL United States	1,572,480	94,640.0	1795.67
3	7	**Trinity** - Cray XC40, Xeon E5-2698v3 16C 2.3GHz, Intel Xeon Phi 7250 68C 1.4GHz, Aries interconnect , **Cray/HPE** DOE/NNSA/LANL/SNL United States	979,072	20,158.7	546.12
4	8	**AI Bridging Cloud Infrastructure (ABCI)** - PRIMERGY CX2570 M4, Xeon Gold 6148 20C 2.4GHz, NVIDIA Tesla V100 SXM2, Infiniband EDR , **Fujitsu** National Institute of Advanced Industrial Science and Technology (AIST) Japan	391,680	19,880.0	508.85

Exascale race

SCtrain | SUPERCOMPUTING KNOWLEDGE PARTNERSHIP

1 000 000 000 000 000 000 FLOPS

Leonardo

Leonardo – Pre-exascale HPC system

Based on Atos BullSequana XH2000 technology

13,000 GPUs based on NVIDIA Ampere architecture

3456 GPU nodes (Intel Ice Lake + Nvidia GPUs)

1536 GPP nodes (Intel Sapphire Rapids)

250 Petaflops – 120 P Byte

Computing racks 95% Direct Liquid Cooled

Warm water: Inlet temperature of 37 degrees, outlet 47 degrees

NVIDIA Mellanox HDR 200 interconnect

Dragonfly+ topology

1.11:1 (intra-cell)

0.8:1 globally

Applications

SCtrain | SUPERCOMPUTING KNOWLEDGE PARTNERSHIP

CUBISM-MPCF

NEMO

CODE SATURNE

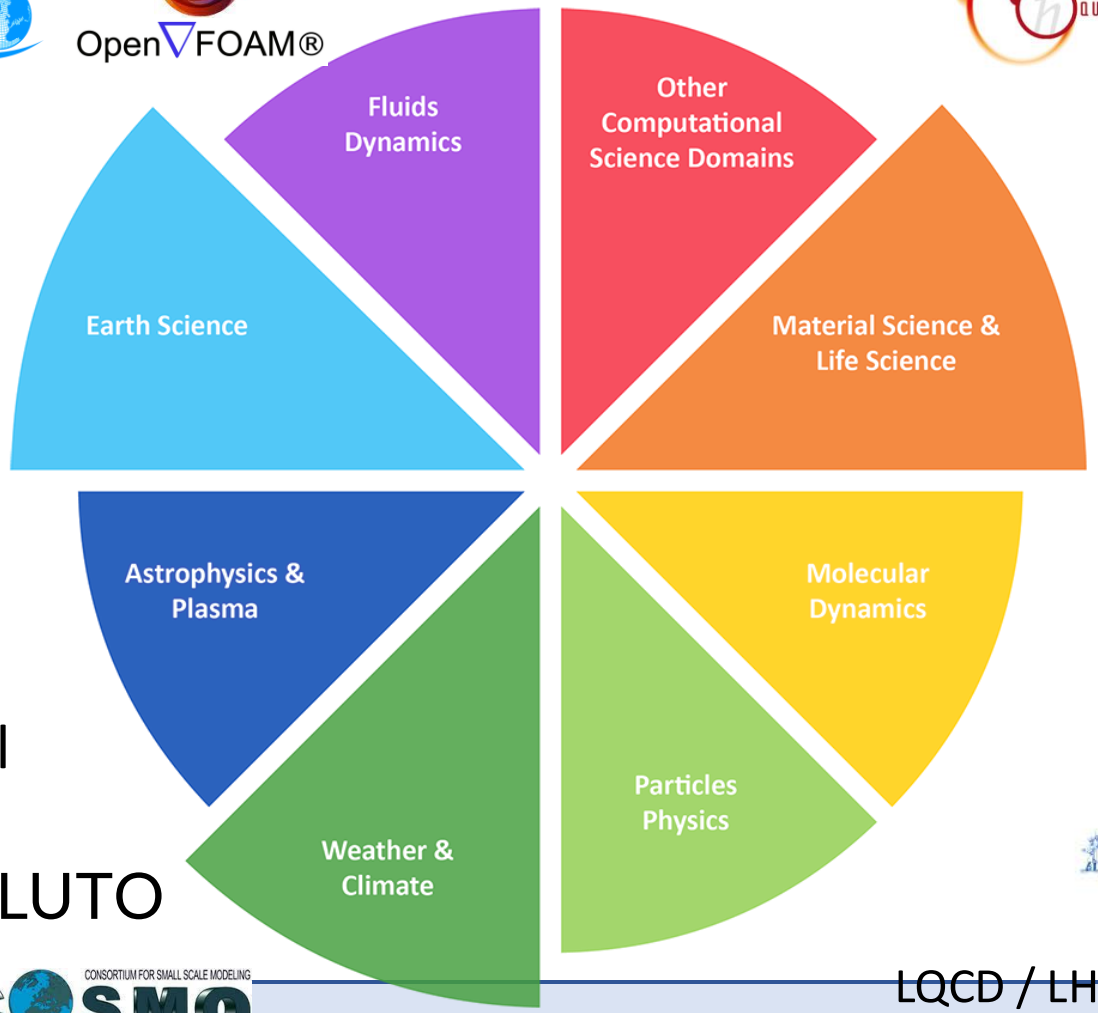
OpenFOAM®

QUANTUMESPRESSO

CP2K

SPECFEM 3D Globe

BigDFT



Gene

LAMMPS

GADGET

GROMACS
FAST. FLEXIBLE. FREE.

NAMD
Scalable Molecular Dynamics

Gysel

a5D

PLUTO

TINKER

CONSORTIUM FOR SMALL SCALE MODELING
COSMO

LQCD / LHC

MILC
qcd

IFS ECMWF


LEONARDO
CINECA

Tecnopolo di Bologna

1950s structure designed by Ing. Pier Luigi Nervi

ECMWF DC relocation

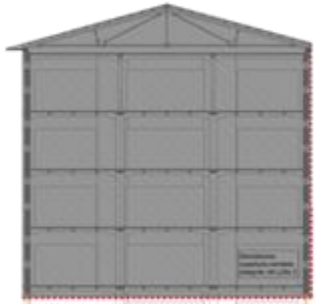
CINECA

INFN

Capannone Miscela C2 - LEONARDO

Capannone Miscela C2

Ballette building

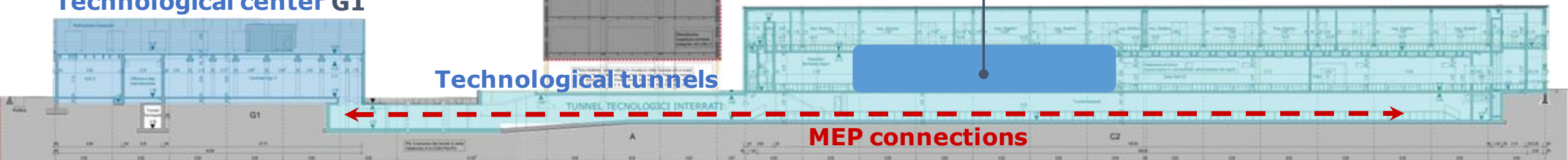


Data Hall Leonardo

Technological center G1

Technological tunnels

MEP connections



HELLENIC REPUBLIC
Ministry of Digital Governance

REPUBLIKA SLOVENIJA
REPUBLIC OF SLOVENIA
Ministrstvo za izobraževanje, znanost in šport
Ministry of education, science and sport

CINECA

MINISTRY OF
INNOVATION AND TECHNOLOGY

MINISTRY
OF EDUCATION, SCIENCE,
RESEARCH AND SPORT
OF THE SLOVAK REPUBLIC

Federal Ministry
Republic of Austria
Education, Science
and Research

Resources evaluation

Weather forecast

DATA

- 6 variables to predict (8 bytes each)
- Globe is divided in 50 Giga cells

TIME TO SOLUTION

- 1 timestep simulates 30 sec per cell
- 1000 ops for each timestep

Simulation coverage: 24 hours

Weather forecast

- 6 variables (8 bytes each)
- 50 Giga cells

- 30 sec per timestep
- 1000 ops/timestep
- 24 hours

Memory requirements:

(6 var) x (8 Byte) x (5 x 10^10 cells) ≅ 2 x 10^12 Byte = 2TB

Weather forecast

- 6 variables (8 bytes each)
- 50 Giga cells

- 30 sec per timestep
- 1000 ops/timestep
- 24 hours

Timesteps:
(24 * 60) min * 2 ts/min ≅ 3000 ts

Total ops:

(50 x 10^9 cells) x (1000 ops/ts) x (3000 ts) ≅ 150 x 10^15 = **150 Petaops**

Weather forecast

- 6 variables (8 bytes each)
- 50 Giga cells

- 30 sec per timestep
- 1000 ops/timestep
- 24 hours

Using a laptop of 1 TeraFLOPS

Time to solution:

150 10^15 ops / 10^12 op = 150x10^3 s ≅ 2 days (24 h to predict)

Questions?

User workflow

How to login

Switch on the laptop

Login on the desired user

ssh <username>@login.g100.cineca.it
ssh login.g100.cineca.it -l <username>

PASSWORD

Work environment

	env. variable	Dimension	Cluster access	user/project
Home	$HOME	50 GB	Local	User
Scratch	$CINECA_SCRATCH	Automatic clean	Local	User
Work	$WORK	1 TB	Local	Project
Dres	$DRES		Shared	Project
Tape	$TAPE		Shared	User

SCtrain | SUPERCOMPUTING KNOWLEDGE PARTNERSHIP

Practice together: login on G100

Transfer files

LOCAL

REMOTE

scp <path-file> <username>@login.g100.cineca.it:<remote-path>

scp <username>@login.g100.cineca.it:<path-file> <local-path>

Autentication required (ssh password)

The exact path of the files needs to be specified

Exercise

- Login on G100 (ssh)
- Check the path working directory (pwd)
- Create a directory SCtrain_introHPC in your home (mkdir)
- Create a new text file (touch/nano/vim)
- Copy the file on your laptop (scp)

Comando	Descrizione
cd	Change directory
mkdir	Make directory
cp	Copy a file (need to specify source and destination)
mv	Move a file (need to specify source and destination)
pwd	Show the path working directory
vim/nano	Text editor
hostname	Show the hostname
cat	Show text file content
ls	List all the files in a directory

Questions?

Work environment: SUDO

Module system

SCtrain | SUPERCOMPUTING KNOWLEDGE PARTNERSHIP

Modules are grouped in different profiles. If you need to load a module which is in a different profile than base, you need to load first the profile

Available profiles on the cluster

```
[dbrandon@login02 ~]$ modmap -profiles
advanced
archive
astro
base
bioinf
chem-phys
deeplrn
eng
lifesc
neurosc
profile
```

Check in which profile the needed module is

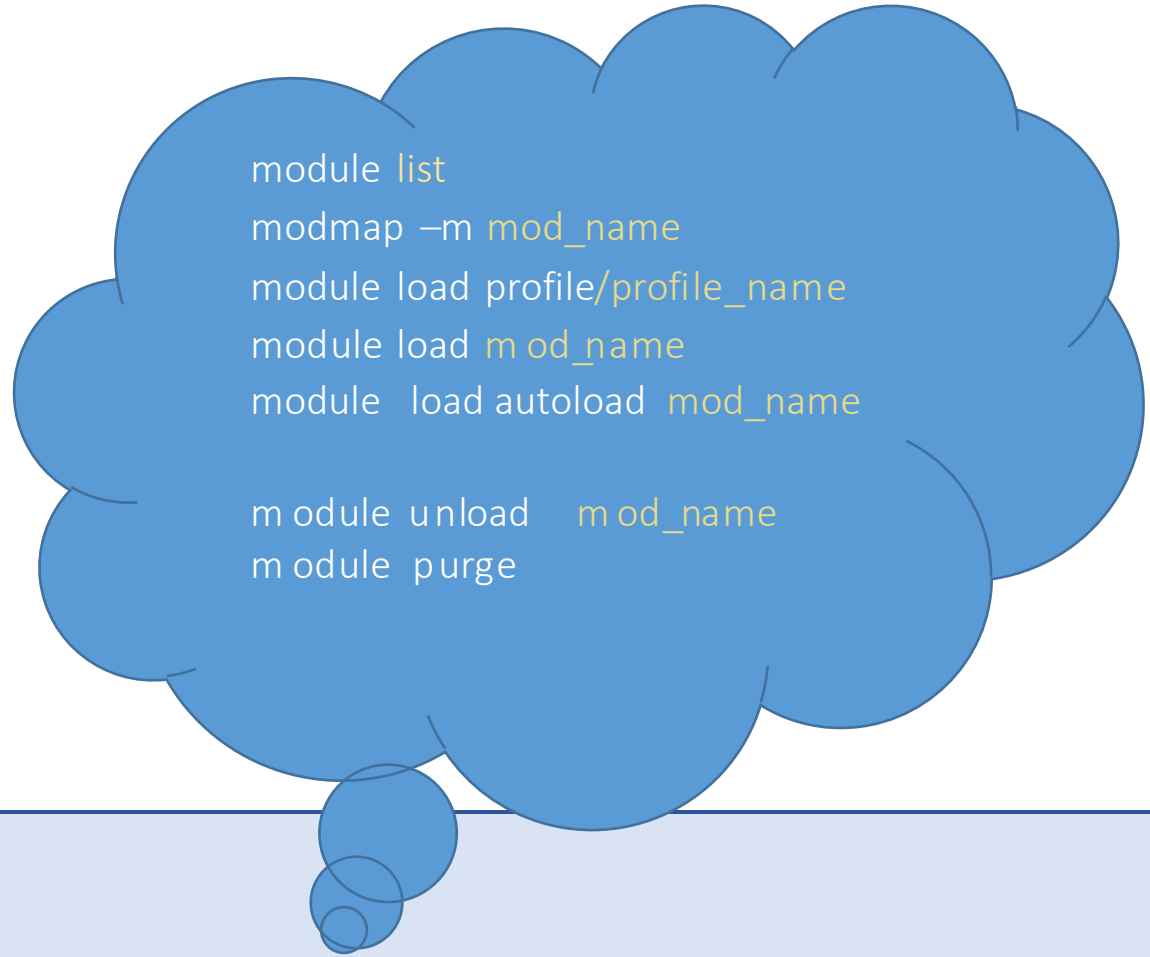
```
[dbrandon@login02 ~]$ modmap -m python
Profile: advanced
Profile: archive
Profile: astro
Profile: base
                python
                 3.8.12--gcc--10.2.0
                 3.8.12--gcc--8.4.1
                 3.8.12--intel--2021.4.0
                python
                 3.8.6--gcc--10.2.0
                 3.8.6--gcc--8.3.1
                 3.8.6--intel--2021.2.0
                 3.8.6--oneapi--2021.2.0
Profile: bioinf
Profile: chem-phys
Profile: deeplrn
Profile: eng
Profile: lifesc
Profile: neurosc
Profile: profile
```

SCtrain | SUPERCOMPUTING KNOWLEDGE PARTNERSHIP

Practice together: how to use modules

Exercise

1. Login on the cluster
2. Load openfoam/8.0
3. Check the currently loaded modules
4. Remove all the currently loaded modules



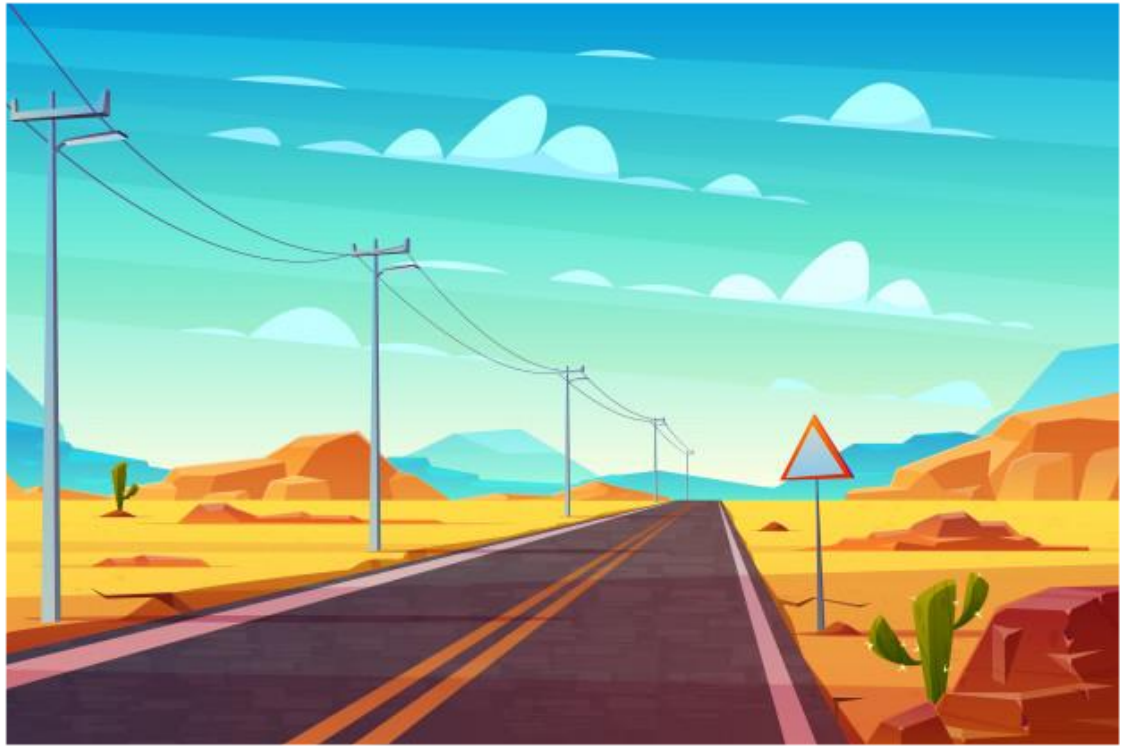
Module system

Command	Description
module load <module name>	Load a module
module load autoload <module name>	Load a module and its dependencies
module av	List of all available modules
module list	List of currently loaded modules
module unload <module name>	Unload a module
module purge	Unload all the currently loaded modules
module help <module name>	print out the help of the module
module show <module name>	print the environmental variable set by loading <module name>

Questions?

Scheduler

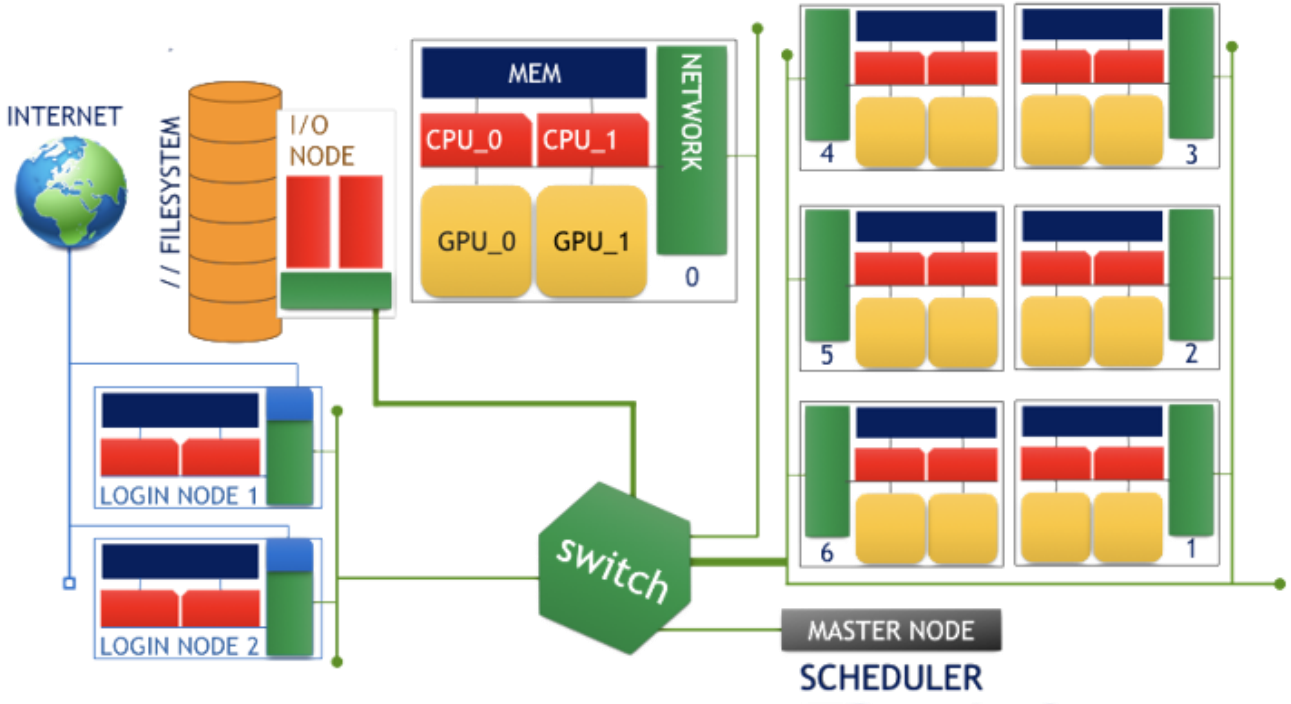
Scheduler



Scheduler

i. Login on the cluster
ii. You need to run the code on the compute nodes (processes on the login nodes have a time limit ~10 min)
iii. To work interactively on login nodes you can use RCM

ssh <username>@login.g100.cineca.it



Submit a job to the scheduler

Submit a job

Run code on compute nodes

- Login on the cluster
- Write the **jobscript**
- Submit the jobscript to the scheduler
- Wait for the job to be executed

- Allocates access to resources
- Manage job starting, executing and monitoring
- Manages queue of pending jobs

SLURM: Simple Linux Utility for Resource Management

Each jobscript can be divided in the following parts

- Set the shell for the script execution
- Slurm directives
- Load the modules
- Run the code

```
#!/bin/bash

#SBATCH --account=<account name>
#SBATCH --time=00:30:00
#SBATCH --job-name=<nome job>
#SBATCH --output=job.out
#SBATCH --error=job.err
#SBATCH --mem=100GB
#SBATCH --nodes=4
#SBATCH --ntasks-per-node=32
#SBATCH --ntasks=128
#SBATCH --partition=g100_usr_prod
#SBATCH --reservation=reservation_name

module load profile <modulo da caricare>
module load autoload <module da caricare>

<executable>
```

SCtrain | SUPERCOMPUTING KNOWLEDGE PARTNERSHIP

Practice together: how to submit a job

SLURM directives

- #!/bin/bash:
  Bash shell
- #SBATCH --account=<account name> -A <account name>
  Name of the account (different from username!)
- #SBATCH --time=00:30:00 -t 00:30:00
  Time limit for the job. If the job exceeds the time limit specified, it will be killed from SLURM
- #SBATCH --job-name=<job name> -J <job name>
  Name of the job
- #SBATCH --output=job.out -o job.out
  Name of the file where the standard output is written
- #SBATCH —reservation=<reservation name>
  Allocates the resources for the job from the specified reservation (reservation=set of reserved nodes for the account <account name>)

SLURM directives

- #SBATCH --error=job.err -e job.err
  Name of the file where the standard error is written
- #SBATCH --mem=100GB
  Allocated memory for each compute node
- #SBATCH --nodes=4 -N 4
  Number of nodes
- #SBATCH --ntasks-per-node=32
  Number of cpus per node
- #SBATCH --ntasks=128
  Number of cpus. The value of --ntask needs to be consistent with --nodes and --ntasks-per-node
- #SBATCH --partition=gll_usr_prod
  Partition (I.e. group of nodes) of the cluster for the resource allocation. On G100, the partition you will use is

SLURM directives

- #SBATCH --gres=gpu:4
  Number of gpus

- #SBATCH --mail-type=<mail_events> (NONE, BEGIN, END, FAIL, REQUEUE, ALL)
  Email notification. An email will be sent to you when something happens to your job, according to the keywords you specified (NONE, BEGIN,END, FAIL, REQUEUE, ALL)

- #SBATCH --mail-user=<user_list> (email address)
  Specifies the e-mail address for the keyword above

Other SBATCH directives can be found at https://slurm.schedmd.com/sbatch.html

Submit and monitor a job

- **sbatch <job name>**
  Your job will be submitted to the SLURM scheduler and executed when there will be nodes available (according to your priority and the partition you requested)

- **squeue -u <username>**
  Shows the list of all your scheduled jobs, along with their status (idle,running, closing, ...) Also, shows you the job id 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 is not starting and, if it is scheduled with top priority, you will get an estimated start time

- **scancel <job_id>**
  Remove the job

Exercise: write a jobscript

i. Login on G100

ii. Load profile/eng and openfoam/8.0

iii. Copy the folder
/cineca/prod/opt/applications/openfoam/8.0/intelmpi--oneapi-2021--binary/OpenFOAM-8.0/tutorials/incompressible/icoFoam/cavity/
in SCtrain_introHPC

cp –r /cineca/prod/opt/applications/openfoam/8.0/intelmpi--oneapi-2021--binary/OpenFOAM-8.0/tutorials/incompressible/icoFoam/cavity/ $HOME/SCtrain_introHPC

iv. Write the jobscript

v. Check the output of the job

```
#!/bin/bash

#SBATCH --account=<account>
#SBATCH --job-name=jobExmp
#SBATCH --output=job.out
#SBATCH --error=job.err
#SBATCH --time=00:30:00
#SBATCH --ntasks=1
#SBATCH --mem=1GB
#SBATCH --partition=gll_usr_prod
#SBATCH --qos=gll_qos_shared

module load profile/eng
module load openfoam/8.0

cd cavity/

blockMesh
icoFoam
```

Questions?

Submit an interactive job

Sometimes you may need to work interactively on compute nodes.

i. ALLOCATE THE RESOURCES

```
salloc -N nodes -n tasks  -A account  -t min  -p partition  --mem=1GB
```

ii. SUBMIT THE JOB

```
srun –n $SLURM_TASK <job name>
```

SCtrain | SUPERCOMPUTING KNOWLEDGE PARTNERSHIP

Practice together: how to submit an INTERACTIVE job