CINECA High-Performance Computing System: Leonardo

Muhammad Mohsin - DIBRIS, University of Genoa, Italy

Vega Research Laboratories s.r.l.

Introduction to CINECA

- CINECA is Italy's largest supercomputing center and a **key player in European HPC**.
- **Center of excellence in the Italian and European ecosystem** for supercomputing technologies development for frontier applications
- Established in 1967, it provides **computational resources for academia and industry**.

Overview of HPC

- High-Performance Computing (HPC) utilizes powerful computers to solve complex problems.
- It enables simulations and data analysis at scales beyond traditional computing.
- It provides parallel computing capabilities for data-driven technological solutions.

What is Leonardo?

- Leonardo is one of Europe’s most advanced supercomputers, designed to support a wide range of scientific research.
- European Computational Excellence and Collaborative Ecosystem
- It is designed to support large-scale simulations and data-intensive applications.
- The system aims to enhance computational capabilities for researchers across various disciplines.

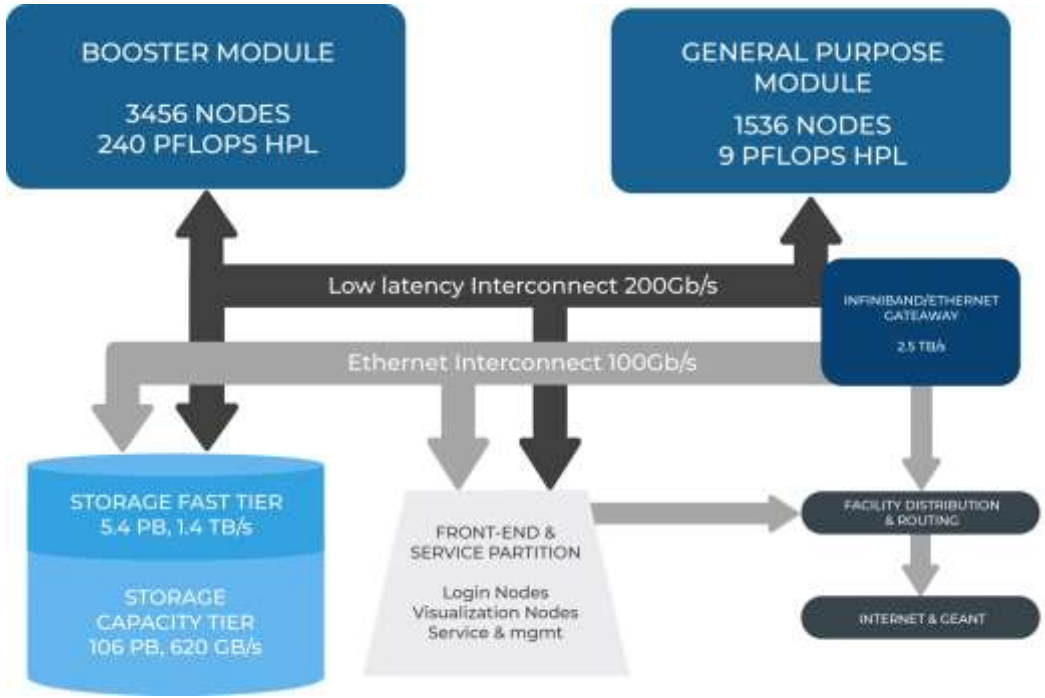
Technical Specifications

- The architecture of Leonardo is based on a hybrid model combining CPU and GPU computing.
- Leonardo boasts a peak performance of over 250 petaflops, making it a formidable computing resource.
- It utilizes cutting-edge technologies, including NVIDIA GPUs and AMD EPYC processors.

Architecture of Leonardo

- The architecture of Leonardo is based on a hybrid model combining CPU and GPU computing.
- It features a massive memory bandwidth to handle complex computations and large datasets.
- The system is organized in a modular fashion, allowing for easy upgrades and maintenance.

Architecture of Leonardo



Booster Module (Atos BullSequana X2135)

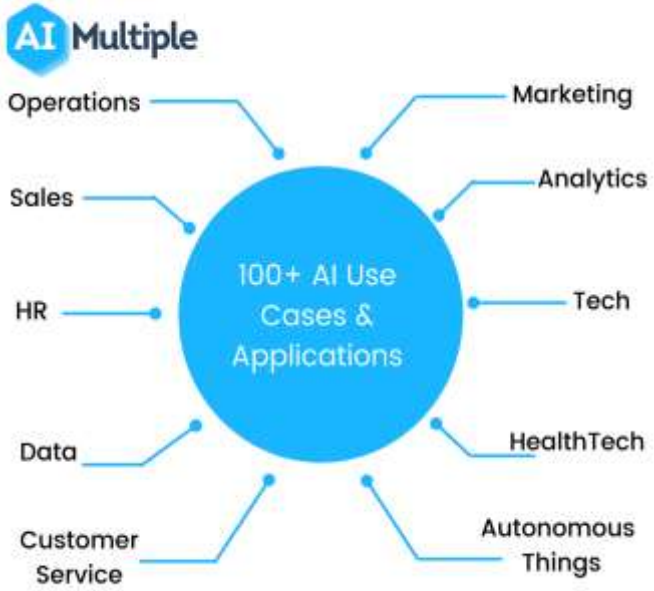
- **Model:** BullSequana X2135 “Da Vinci” single node GPU Blade
- **Nodes:** 3456 (each like a powerful computer)
- **Processors:** single socket 32-core Intel Xeon Platinum 8358 CPU, 2.60GHz (Ice Lake)
- **Cores:** 110592
- **RAM:** 8x 64 GB DDR4 3200 MHz (512 GB)
- **Accelerators:** 4x NVIDIA custom Ampere A100 GPU 64GB HBM2e, NVLink 3.0 (200GB/s)
- **Network:** 2 x dual port HDR100 per node (400Gbps/node)

Data-Centric General Purpose (DCGP) Module

- **Model:** BullSequana X2140 three-node CPU Blade
- **Nodes**: 1536
- **Processors**: Intel Saphire Rapids 2×56 cores, 2.0 GHz
- **Cores:** 172032 (112 cores/node)
- **RAM:** (48×32) GB DDR5 4800 MHz
- **Network:** 3xNvidia HDR cards 1x100Gb/s

Applications in Research

- Leonardo supports a wide range of research applications across disciplines.
- It enables researchers to conduct simulations that require significant computational power and data processing.
- Different software including libraries, tools, and compilers are compatible for instance CUDA.



Collaboration Opportunities

- CINECA encourages collaborations with universities and research institutions.
- Leonardo provides access to computational resources for joint projects.

CALL ISCRA B – STANDARD PROJECTS

- medium-sized projects (from 50,000 to 250,000 GPU hours)

CALL ISCRA C – SMALL RESEARCH, EXPERIMENTATION, AND DEVELOPMENT PROJECTS

- up to 8000 GPU hours on Marconi100 and up to 100000 core hours on Galileo100

CALL ISCRA D – DATA STORAGE

- From a few dozen to a few hundred TB for 2-3 years

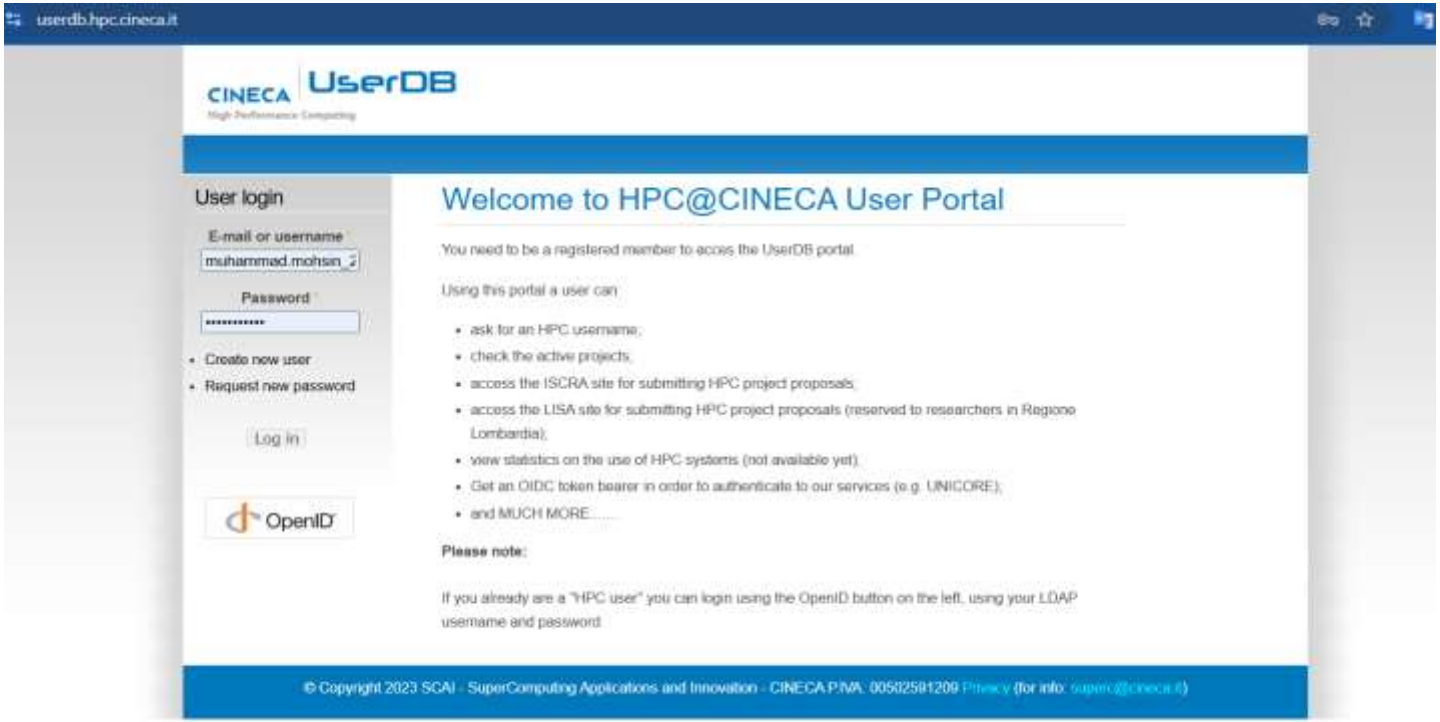
Creating an Account and Accessing HPC Shared Resources

Leonardo Cluster

Step 1: Account Creation

- Go to the Cineca portal: **https://userdb.hpc.cineca.it/**
- Register using your institutional credentials or request access through your PI or project manager.
- Once your account is verified, you will receive login credentials to access the HPC resources.

Example


userdb.hpc.cineca.it
CINECA UserDB
User login
E-mail or username
muhammad.mohsin
Password
Create new user
Request new password
Log in
OpenID
Welcome to HPC@CINECA User Portal
You need to be a registered member to acces the UserDB portal
Using this portal a user can
ask for an HPC username;
check the active projects;
access the ISCRA site for submitting HPC project proposals;
access the LISA site for submitting HPC project proposals (reserved to researchers in Regione Lombardia);
view statistics on the use of HPC systems (not available yet);
Get an OIDC token bearer in order to authenticate to our services (e.g. UNICORE);
and MUCH MORE.....
Please note:
If you already are a "HPC user" you can login using the OpenID button on the left, using your LDAP username and password
© Copyright 2023 SCAI - SuperComputing Applications and Innovation - CINECA P.IVA: 00502591209 Privacy (for info:

Step 2: Accessing the Server

- Use SSH to log into the HPC environment:
  ssh yourusername@login.leonardo.cineca.it
- Ensure you have valid credentials and are added to the correct project with appropriate resource allocations.
- You can manage file transfers via SCP or SFTP tools.

Example

- Data Storage architecture
  - $HOME: permanent/backed up, user specific, local
  - $WORK: permanent, project specific, local
  - $FAST: permanent, project specific, local (LEONARDO ONLY)
  - $CINECA_SCRATCH: temporary , user specific, local
  - $TMPDIR: temporary, user specific, local
  - $DRES: permanent, shared (among platforms and projects)
  - Backup policies
  - Environment variables
    - No valid Data Center license found
  - Summary
  - What to use when...
- Monitoring the occupancy
  - No valid Data Center license found
  - No valid Data Center license found
  - No valid Data Center license found
  - No valid Data Center license found
- File permissions
- Pointing $WORK to a different project: the chprj command
- Endianness
- Managing your data

Example

Data Storage architecture

All HPC systems share the same logical disk structure and file systems definition.

The available storage areas can be

- **temporary** (data are cancelled after a given period);
- **permanent** (data are never cancelled or cancelled only a few months after the "end" of the project);

they can also be

- **user specific** (each username has a different data area);
- **project specific** (accessible by all users within the same project).

Finally, the storage areas can be:

- **Local** (specific for each system);
- **Shared** (the same area can be accessed by all HPC systems)

The available data areas are defined through predefined "environment variables": $HOME, $CINECA_SCRATCH, $WORK.

Important: It is the user's responsibility to backup your important data. We only guarantee a daily backup of data in the $HOME area.

Step 3: Managing Resources

- Once logged in, check your current project and resource allocation by running the command:
- **module load cineca-tools**
- **saldo -b**
- This will show your project balance, CPU hours used, and other resource information.
- Make sure to manage your usage to avoid overconsumption of allocated resources.

SLURM Scheduler on HPC Systems

- **Simple Linux Utility for Resource Management**
- Ensures fair resource access in shared HPC environments by managing job scheduling.
- **Interactive Mode**: Used for data movement, code development, compilations, and brief test runs.
- Limits: **Max 10 minutes of CPU time**; **free of charge** under the current billing policy.

SLURM Scheduler on HPC Systems

Batch Mode:

- For long production runs.
- It requires a shell script specifying operations, executed once resources are allocated.
- Store all data in **$WORK** or **$CINECA_SCRATCH** for compute node access.

Step 4: Submitting Jobs

```
#!/bin/bash
#SBATCH --job-name= example_job        # Job name
#SBATCH --output=output.log # Output log file
#SBATCH --error=error.log   # Error log file
#SBATCH --time=24:00:00                # Time limit (24 hours)
#SBATCH --partition=boost_usr_prod      # Use one of the available partitions
#SBATCH --gres=gpu:2                   # Request two  GPU
#SBATCH --ntasks=1                     # Number of tasks
#SBATCH --cpus-per-task=16             # CPU cores per task
```

Step 4: Submitting Jobs

Jobs are submitted to the HPC queue using SLURM. A sample job script is shown below:

#SBATCH --mem=100GB # Memory per node

#SBATCH --nodes=1

#SBATCH --account=iscrc_weee-ai # Request 2 nodes

#module load python/3.11.10

#module load cuda/12.4

Activate your virtual environment

source /leonardo/home/userexternal/mmohsin0/virtualenvironment/bin/activate

srun ./your_application #python3 test.py

Submit the job using: **sbatch job_script.sh**

Check the status of current jobs using: **squeue -u yourusername**

Step 5: Monitoring and Managing Jobs

- Use the following commands to manage your jobs:
- **squeue:** to check running and queued jobs
- **scancel:** to cancel a job
- **sacct:** to view job accounting information
- You can also monitor CPU and memory usage to optimize job performance and resource utilization.

Thank you
Any Question?