**System Overview**

Hpc64 is a centrally administered computing cluster dedicated to research computing, composed of servers owned by various labs in the Division of Science. The system is accessible at hpc64.brandeis.edu to users with accounts. Upon login the users land on the "login node", and from there they can organize their files, compile the software and prepare the runs, and then interact with the job scheduler to submit the calculations to the compute nodes.

NOTE: the login node must not be used to run directly calculations, but only to submit calculation to the scheduler.

There is an HPC Advisory Committee that oversees the cluster. System management, application and user support and users training falls to Francesco Pontiggia.

If you have a technical question or problem that cannot be addressed by the documentation below, or you need HPC related advise, please open a ticket using the form at this link: [Open a Ticket](#).

The cluster runs 64-bit Rocks/RHEL linux and is composed by an heterogeneous collection of Intel Xeon based hardware ranging from 2 sockets (8 cores/node) Dell PowerEdge 1950 to the more recent 4 sockets (32 cores/node) M820Blade servers.

The system consists of 161 compute nodes (for a total of 1732 cores) and additional 16 nodes (240 cpu cores) which also feature a total of 62 Nvidia GPUs (for a total of 115392 Cuda cores).

The DDR3 RAM on the nodes ranges from 8GB (1GB/CPU core) for many of the older Dell1950 servers to 128GB (4GB/CPU core) for the recent Dell M820Blade servers.

---

**Accessing the System**

- Accounts
- Login
- Connecting from Off Campus:
  - Displaying graphics: X2go and X11 forwarding.

**File Systems**

- Home File System
  - Some considerations about I/O in your applications.
    - Writing binary unformatted data
    - Binary vs formatted ASCII I/O in C
- Scratch Space

**Computing Environment**

- Linux shell and environment
- Software Modules
- Available Applications

**Running Calculations**

- Policy and queues
  - Resource sharing policy
  - List of queues in the system
• Managing Jobs
  • Submitting a batch job: qsub
    • qsub typical use cases
      • How to run a simple serial job?
      • How to use job relocation upon suspension
      • Output and Error files
      • How to run a parallel job?
      • Large Memory Jobs
      • Job Array, large number of (nearly)identical jobs?
      • Job Dependencies
      • Soft requests
      • Local scratch space
  • Interactive sessions: qrsh
  • Monitor your jobs: qstat
    • qstat typical use cases
      • How to check my jobs?
      • How to see all the jobs in the queueing system?
      • How to see the general status of the queues to see if there are slots available?
      • How can I get a list with the status of the nodes associated to my queue?
      • How can I see who is running in my queue?
      • How can I see who is running on my node?
      • How to get additional information on memory and cpu usage of my job?
      • How to get additional information on a node?
  • Deleting your jobs: qdel
  • Modify a pending job: qalter
  • Suspending jobs: qmod
  • Tips on how to run some popular applications
    • How to run Matlab jobs

HPC Advisory Committee