ABINIT School 2026

How to work on the INTI cluster of CEA: Environment and job submission

---

Content

First step: log on the *INTI* cluster with your username and password.
To do so, in a terminal window, enter : **ssh -Y username@inti.ocre.cea.fr**
You should use -Y and not -X for X11 forwarding as it seems to work better.
Your username should be of the form *stagXX* with XX between 01 and 38.

At any time, you can type the **man inti** *or* **machine.info** *command on the cluster to access the manual of the cluster.*

1. The "module" command

The *INTI* cluster uses Environment Modules allowing the dynamic modification of a user's environment via modulefiles.
To familiarize yourself with modules, you can first list the available modules with the command:

module avail

The modules that are being used right now in your environment are given by the command:

module list

You can purge your environment by unloading all the modules at any time with:

module purge

To load a specific module like abinit for instance, you just need to use the simple command:

module load abinit

All packages required for its execution (dependendies) will be automatically loaded.
You will then have abinit in your path (try it by typing for instance *abinit --version*).

For the purpose of the school we have created specific modules that are located in */ccc/work/cont999/formation/stag39/ABISCHOOL/ENV/MODULES.*
To be able to use these modules you first need to type

module use /ccc/work/cont999/formation/stag39/ABISCHOOL/ENV/MODULES

This will be done automatically when you connect to INTI. You can check that you indeed see the modules for the Abinit school by typing **module avail**
You should see a list of modules available including one called abischool/amd. This is the module that will be loaded by default if you enter

module load abischool

This will load several useful libraries and softwares for the school.

You might notice that there are different version (amd and nvidia) of this module. That is because we have access to two different partitions on the cluster

- The **rome** partition with nodes composed of 128 AMD Rome CPUs
- The **gh200-bxi** partition with nodes composed of 288 ARM Neoverse CPUs and 4 Nvidia GH200 GPUs

Logically, if you run on the GPU (gh200-bxi) partition, you should load the abischool/nvidia module.

Note that we have set up an environment variable called ABISCHOOL to directly access the directory /ccc/work/cont999/formation/stag39/ABISCHOOL where all the documents and other useful files for the school will be located.

2. Job submission

On such a cluster you **should not** directly execute your calculation by running a simple command like *abinit input.abi* or *mpirun -n 128 abinit input.abi* (for a parallel calculation).
All calculations should run on one or several computing nodes so you have to submit your "job" to a computing node.

Job submissions and resources allocation are managed by a specific environment.
Special commands prefixed by **ccc_** are provided to execute these operations (this is specific to the TGCC computing center).

First, you need to write a script which will define a set of directives about your job and how to execute it. Then submit your job on a given batch queue.
Here is a typical script which you can write in a file called e.g. job-ls.sh:

```
#!/bin/bash
#MSUB -r MyJob                    # Name of the job (to be changed)
#MSUB -n 50                       # Number of MPI processes to use
#MSUB -c 1                        # Number of tasks per process to use
#MSUB -T 600                      # Time limit in seconds
#MSUB -o ls_%I.o                  # Standard output. %I is the job id
#MSUB -e ls_%I.e                  # Error output.    %I is the job id
#MSUB -q rome                     # Partition name
#MSUB -E '--reservation=Formation-Abinit-cpu-20260202' # For the ABINIT school only

export OMP_NUM_THREADS=1          # Usually set to the same number as in the "MSUB -c" line above

ls > output                       # Execute a simple ls
```

Note that the first 9 lines are commented by #, so that they are not a part of the shell script. However the system still reads and understands them.

The first 7 lines starting by #MSUB define the

- The name of the job
- The number of MPI processes
- The number of tasks per process
- The time limit for the execution
- The standard output and standard error files
- The name of the partition (rome or gh200-bxi for us here)

Note: the total number of processors used is equal to the number of processes (-n) times the number of tasks per process (-c).

The last line is specific to the ABINIT school and tell the system to use the nodes reserved for the school.
The name of the reservation is **Formation-Abinit-cpu-202602XX** where 02XX is the date (ie. 0202 for the first day, 0203 for the second etc.).
The rest of the file contains usual commands in bash to launch the calculation

To submit this job, just run:

ccc_msub job-ls.sh

Various tools provides you with information about your job.

- **ccc_mpp** provides information about the execution and gives you the **BatchID**.
  Note: *job-ls* runs fast so it might be finished before you can see any information. You can add *-u your_username* to only see the information about your jobs.
- **ccc_mpeek BatchID** is useful to have more information about the job.
- **ccc_mdel BatchID** can be used to kill the job during execution.

After the job is executed, you will find two new files in the current directory corresponding to the standard output and error files.
You can have a look at these two files to see if the job executed correctly.

Note

One usually submits jobs directly from the login node where you are right after connecting to INTI.
However, in many cases during the school, you will instead have to connect directly to a computing node.
Then you can run your calculations directly since you already are on a node.
To do so, you first need to start an interactive session on a node, which you can do by running

ccc_mprun -K -X first -p rome -n 128 -c 1 -m work -E '--reservation=Formation-Abinit-cpu-202602XX'

As this is not such a simple command we created an alias to start the interactive session. So you should be able to start a session by simply typing

start_session

Once this is done you still need to connect to the associated node before being able to run your calculations.
You can do so by identifying the node on which the session is running for instance using ccc_mpp and then ssh to that node. Here again we defined a command to do it so you can connect by simply typing

connect

Try to run these command and make sure you can indeed connect to a node.
At the end you should first exit the connection to the node by typing

exit

and then close the interactive session by typing

exit

once more.

3. Filesystems

Working on a cluster is not so different from working on a desktop computer.
But since several processors are working at the same time, they might write a huge number of files on the working directory.
Specific *filesystems* have thus been developped for clusters and supercomputers, and it is important to think a little bit about how to correctly handle I/O.
You should not execute ABINIT and produce results in your HOME ($HOME) directory, instead you should do it within the work ($CCCWORKDIR) directory which uses a faster filesystem.

To access your work directory simply type

cd $CCCWORDKDIR

1.4. Trying ABINIT

Let us finally try to connect to a computing node and run a small Abinit calculation.

- First, let us start an interactive session and connect to a node

  start_session

  connect

- Then, we move to the work directory

  cd $CCCWORKDIR

- We can create a directory to run this small test in

  mkdir test_abi
  cd test_abi

- Then, we need to load the modules

  module load abischool

- We need some input files to run abinit.

  cp $ABISCHOOL/test/Al/* .

- Finally, we can run abinit

  abinit Al.abi

- And close the session by typing **exit** twice.
  Once to disconnect from the node and another time to close the interactive session.

Have nice tutorials !

For more information about INTI, type ***machine.info*** or ***man inti*** in a terminal.