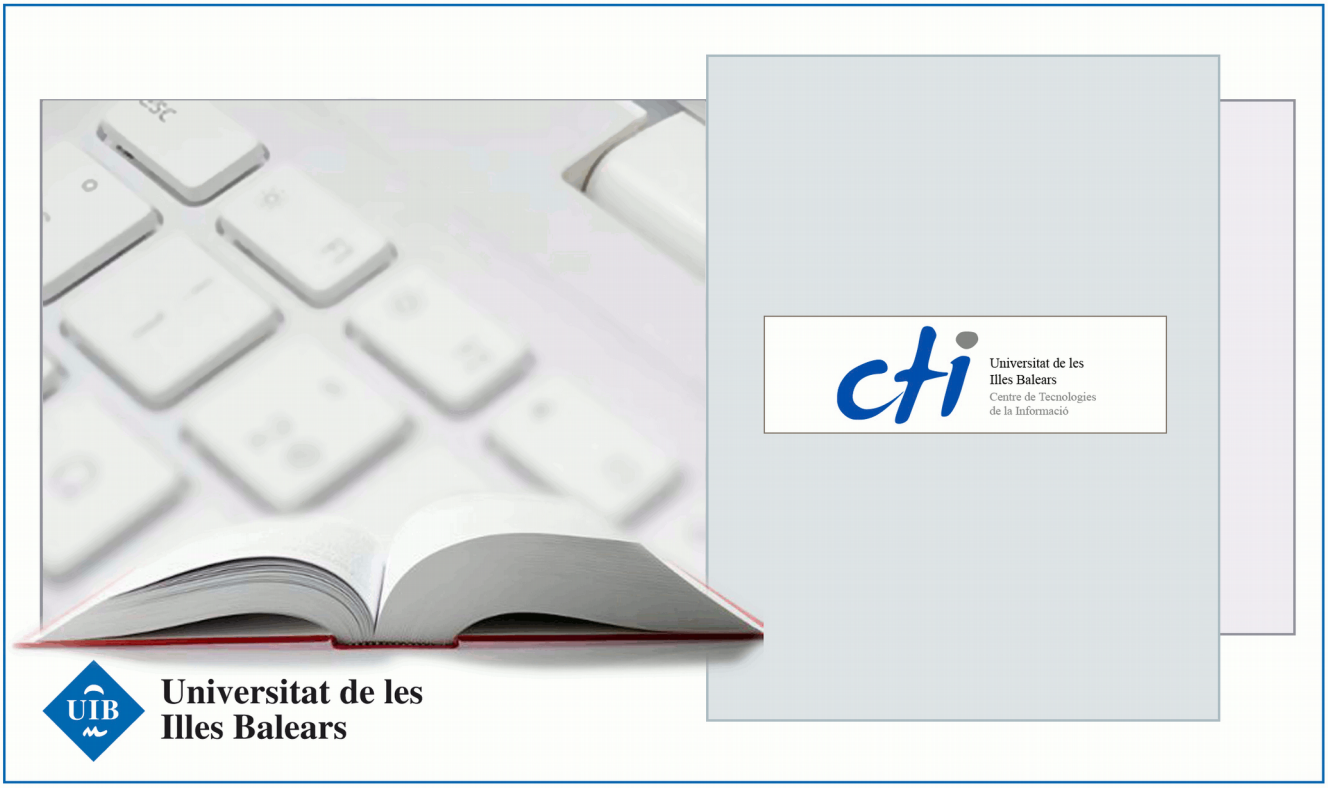
Manual d'usuari



Clúster de càlcul de la UIB

1 Introducció

Aquest manual proporciona una introducció al hardware que hi ha instal·lat al sistema operatiu i aplicacions, a com establir connexions, a com compilar i a com executar treballs d'investigació.

2 Hardware

El clúster està compost per 2 nodes de *login* i 46 nodes de computació. D'entre els nodes de computació n'hi ha 42 de *thin* i 4 de *fat*. Els nodes *fat* són nodes de computació amb majors prestacions de memòria que els *thin*.

Els nodes de computació comparteixen entre ells una xarxa d'altes prestacions anomenada *Infiniband* destinada a les comunicacions dels treballs en execució i a l'escriptura/lectura de dades.

Disposam d'un servidor d'emmagatzematge directament connectat als nodes de *login* que disposa de més de 40TB repartits en diferents particions.

3 Com accedir al clúster

Per tenir accés al clúster necessitau emplenar una sol·licitud disponible a UIBdigital. Una vegada heu activat el servei, hi podeu entrar amb un client *ssh*. Les credencials a emprar són les mateixes que a UIBdigital.

3.1 Accés ssh

Els entorns operatius tipus Linux i Mac ja tenen instal·lat per defecte un client *ssh* i el podeu emprar per a accedir al clúster. En els entorns operatius tipus Windows és necessari instal·lar un client *ssh*. Un dels mes populars és el *Putty*.

Per exemple, des d'un i-Mac amb el programa *Terminal*:

(ordinador_local)$ssh -l lrp111 foner.sint.uib.es

The authenticity of host 'foner.sint.uib.es (10.240.0.20)' can't be established.

RSA key fingerprint is b6:0a:2b:00:f1:06:06:a9:cb:34:33:a0:e7:2f:db:38.

Are you sure you want to continue connecting (yes/no)? yes

Warning: Permanently added 'foner.sint.uib.es,10.240.0.20' (RSA) to the list of known hosts.

lrp111@foner.sint.uib.es's password:

[lrp367@foner2 ~]$

Una vegada hi heu accedit, disposau d'un entorn *shell* de Linux. La *shell* per defecte és la *bash*, però l'usuari pot canviar-la.

3.2 Transferència de fitxers al clúster

Per a transferència de fitxers, ho podeu fer via *sftp* o *scp*. Als entorns Mac i Linux disposam d'eines *sftp* i *scp* instal·lades per defecte al nostre sistema.

En el cas de Windows, podeu fer servir el *Putty* o instal·lar una altra eina més orientada a transferència de fitxers. Un dels mes populars és *Filezilla*, que està disponible per a Mac, Windows i Linux si escau.

En el cas de comandes, si la transferència és des de l'ordinador personal cap al clúster, la comanda és:

(ordinador_local)$scp <fitxer_origen>
<nom_usuari>@foner.sint.uib.es:<directori_destí>

En cas contrari, si és del clúster cap a l'ordinador central és:

(ordinador_local)$ scp <nom_usuari>@foner.sint.uib.es:<camí_fitxer>

<directori_local>

Un exemple és transferir el fitxer local *prova.dat* cap al clúster foner:
(ordinador_local)$scp prova.dat lrp367@foner2.sint.uib.es:/home/lrp367/
lrp367@foner2.sint.uib.es's password:

4 Entorn de treball

4.1 General

El sistema disposa de dos nodes de *login* que proporcionen tots els serveis necessaris perquè es pugui treballar amb el clúster. Aquests nodes es troben en configuració mode *failover*. És a dir, en cas de fallada d'algun dels nodes, els serveis es traspassen al node operatiu per tal d'oferir màxima disponibilitat al servei. Aquests nodes estan orientats per fer-hi l'accés, compilar codi i llançar treballs al planificador de treballs. L'accés als nodes de computació no està permès.

4.2 Estructura de directoris

En haver iniciat sessió per *ssh* al clúster, us trobareu dins d'un entorn GNU/Linux amb l'estructura de directoris habitual. A continuació es descriu quina funció tenen:

- */data/bin:* Aquí s'hi destinen els programes d'usuari. Són aplicacions que poden fer servir tots els investigadors i no tenen cap restricció de llicència.

- */data/scratch:* Directori on emmagatzemar els fitxers temporals de les aplicacions de càlcul. Aquest espai presenta un cicle de vida d'**una setmana**. És a dir, els fitxers amb data de modificació superior a una setmana seran automàticament eliminats.

- */data/exemples_slurm:* Directori on es troben exemples d'execució de *shell scripts* de l'SLURM.

- */data/documentacio:* Documentació vària sobre el clúster, com ara el manual d'usuari.

4.3 Mòduls

Al clúster disposau del paquet GNU Modules, que permet carregar entorns d'execució (exportant les variables d'entorn corresponents) per a cada programa.

4.3.1 Ús

És necessari que, per a l'execució dels vostres treballs, mireu els que estan disponibles amb la comanda:

$ module avail

i després seleccioneu el que necessitau amb:

$ module load <nom_modul>

Podeu consultar informació del mòdul amb la comanda

$ module help <nom_modul>

o

$ module whatis <nom_modul>

Una vegada hàgiu acabat de treballar amb el programa en concret executau:

$ module unload <nom_modul>

Per exemple, per carregar el mòdul *foner-modules/g09* executau:

$ module load foner-modules/g09

Recordau que alguns mòduls són incompatibles entre si, i que per carregar-ne un haureu de fer un *unload* de l'altre.

5 Execució de treballs

5.1 Tipus de treballs

Distingim 2 tipus de treballs que es poden executar al clúster:

- Paral·lel intranode: És aquell càlcul que només permet distribuir feina entre els processadors de dins un mateix node. Degut a que l'escalabilitat està limitada a un node és habitual que aquests treballs tinguin majors necessitats de memòria. L'estàndard més comú es diu *OpenMP*.
- Paral·lel internode: És aquell càlcul que permet executar subtasques a més d'un node. La comunicació entre processadors es fa sempre a través de la xarxa d'altes prestacions *Infiniband*. L'estàndard més comú en aquest cas es MPI.

Normalment, si un treball s'executa de forma paral·lela internode, també ho permet de forma intranode, donant lloc a una execució híbrida (internode + intranode).

5.2 Planificador de treballs

El planificador de treballs que s'empra al clúster foner és l'*Slurm*. Aquest programari assisteix a l'investigador a l'hora de llançar els seus treballs de la següent manera:

- Reserva recursos (nodes de computació) per a una durada determinada.
- Proporciona un entorn per iniciar, executar i fer seguiment de la seva feina (normalment un treball que s'executa en paral·lel) en els nodes que se li han assignat.
- És l'entitat que gestiona els recursos en cas de contenció posant els treballs pendents en una cua d'espera.
- Per obtenir mes informació, podeu anar a la plana principal del l'Slurm.
- Des del punt de vista d'usuari, l'Slurm presenta les següents parts:
  - Node: Màquina física on els processos d'usuari s'executen.
  - Partició: Agrupació lògica de nodes on s'executen els treballs. Aquesta agrupació es pot considerar una cua de treball. Cada partició presenta una sèrie de restriccions com poden ser límit de temps, usuaris permesos etc.
  - Treball (job): Reserva de recursos per un temps determinat a un usuari.
  - Subtreball (Job step): És el conjunt de tasques que s'executen dins un treball.

És molt important executar sempre els treballs fent servir el planificador SLURM i no executar-los mai directament a cap dels nodes. Els treballs que s'executin sense emprar el gestor de cues poden provocar un mal funcionament del sistema i poden ser cancel·lats.

5.3 Particions del sistema

L'entorn disposa de tres particions:

- Partició *thin*: dedicada als treballs que requereixin comunicació internode o híbrida. No es poden executar treballs en una sola màquina, sinó que mínim se'n requereixen dues. Té un temps màxim d'execució per tasca de 3 dies. Els nodes seleccionats per a aquesta partició són els nodes anomenats *thin* a la secció d'introducció.
- Partició *fat*: dedicada als treballs que requereixin comunicació intranodes (a una sola màquina o treballs en sèrie). No podem demanar més d'un node en aquesta partició. Té un temps màxim d'execució per tasca de 3 dies, però si s'especifica amb el paràmetre *–time* pot arribar fins a 6 dies. El conjunt de nodes per a aquesta partició són els anomenats *fat* a la secció d'introducció.
- Partició *test*: dedicada a compilacions, tests i proves. Té un temps màxim d'execució per tasca de 10 minuts, però si s'especifica amb el paràmetre *–time* pot arribar fins a 1 hora. Aquesta és la partició per defecte.

Podeu consultar aquesta i més informació sobre les particions amb la comanda:

$scontrol show partitions

5.4 Llançar treballs

El gestor *Slurm* permet 2 formes de treballar al clúster: interactiva i en diferit. A continuació s'explica cada cas.

5.4.1 Llançar treballs de forma interactiva

En primer lloc, se sol·licita la reserva de recursos necessaris amb les comandes d'*Slurm*. Llavors, quan els recursos estan disponibles, el sistema proporciona l'entorn perquè se'n pugui fer ús. Aquesta forma de treballar és útil per fer proves i tests.

A continuació se'n descriu un exemple:

$srun -p thin -N2 -n15 –time=100 ./wchem

La comanda anterior sol·licita la reserva de 2 nodes de la partició *thin* i un total de 15 cpus per un temps de 100 minuts. Una vegada es disposa d'aquests

recursos, s'executa la comanda *./wchem*. La sortida i errors d'aquesta execució estan vinculats a la sessió de l'usuari.

Hi ha més opcions i paràmetres a part dels esmentats. Per a més informació, consultau la documentació de l'*Slurm*.

5.4.2 Llançar treballs de forma diferida

En primer lloc, l'investigador prepara un script amb directives que l'*Slurm* interpreta especificant la reserva de recursos i la tasca que ha de dur a terme el planificador. Després, l'investigador sol·licita l'execució del treball amb l'*script* que ha preparat. La diferència amb el cas anterior és que el sistema no proporciona un entorn interactiu, sinó que presenta l'evolució de la feina dins fitxers de sortida i error. Aquesta és la forma recomanada per posar feina al planificador.

A continuació se'n mostra un exemple:

$sbatch wrf_exec.sh

La comanda anterior sol·licita al gestor l'execució en diferit del *shell script* wrf_exec.sh. En aquest cas, el planificador interpreta les directives #SBATCH incloses dins el *shell script*. D'aquesta manera, no és necessari mantenir la connexió al clúster per a la visualització dels resultats. A continuació, es mostra un exemple de *shell script* que s'ha d'adaptar en cada cas:

wrf_exec.sh:

```
#!/bin/bash
#SBATCH --partition=thin
#Coa(partició) on volem executar el treball
#SBATCH –output=WRFJobName-%j.out
#Fitxer d'output. MOLT IMPORTANT tenir controlada la sortida del programa si no la volem perdre!
#SBATCH --error=WRFJobName-%j.err
#Fitxer d'error
#SBATCH --nodes=5
#Número de nodes
#SBATCH --ntasks-per-node=2
 #Número de tasques(o processadors) que emprarem a cada node.
source /opt/modules/3.2.10/Modules/init/bash
```

```
#Carregam l'entorn dels modules
module load foner-modules/wrf
# Carregam el programa que volem


cd NETCDF_WORDIR
srun ./wrf.exe
#Execució del programa en paral·lel
```

En l'exemple anterior, se sol·licita disposar de 5 nodes i 2 tasques per node a la partició *thin*. També s'especifica on s'han de posar la sortida i els errors dels treballs. Acte seguit, es carrega l'entorn necessari per permetre l'execució del treball i finalment s'executa.

Hi ha més opcions i paràmetres a part dels esmentats. Per a més informació, consultau la documentació de l'*Slurm*.

És important adequar bé el número de processadors que se sol·licitaran al gestor de cues amb el número processos que emprarà el programa.

5.5 Control i seguiment de l'execució

A continuació s'exposen les comandes bàsiques de control necessàries per a l'usuari.

Comanda	Descripció
scontrol show job [JobID]	Presenta una descripció de l'estat en què es troba el *jobid*
scontrol show partition <nom_particio>	Mostra informació sobre la cua/partició passada. Si no hi posam <nom_partició>, les mostrarà totes.
scontrol show node <nom_node>	Comprova l'estat d'un node. Si no hi posam <nom_node> els mostrarà tots.
scontrol update jobid=<id_tasca> Nice=[0..10000]	Disminueix la prioritat dels nostres treballs. 0 significa no modificar la prioritat i 10000 assignar-li'n la mínima.
scontrol requeue <id_tasca>	Reencua el treball passat.
smap [-i 1]	Mostra de forma gràfica l'assignació de nodes a treballs. L'opció -i 1 indica que s'actualitzarà la informació cada segon.
sinfo -n <nodes> o sinfo -p <particio>	Mostra l'estat d'uns nodes o particions concrets.
sinfo -T	Mostra les reserves del sistema.
scancel [<JobID> | --user=<nom_usuari>]	Cancel·la el treball amb ID JobID. Amb l'opció –user=<nom_usuari> cancel·la tots els treballs de l'usuari passat.
squeue	Mostra l'estat de les cues del sistema.

Consultau més informació amb el *man* de cada comanda.

6 Compilació d'una aplicació

En cas que sigui necessari executar el vostre codi, el sistema disposa d'eines per compilar. Normalment, per compilar eines per a càlcul científic necessitau:

1. Compiladors: El clúster disposa dels compiladors GNU i els Intel. Per obtenir el màxim rendiment de l'executable és recomanable en principi sempre fer ús dels compiladors Intel si és possible. La utilització d'un compilador o un altre es fa amb la càrrega respectiva del seu "module".
2. Llibreries *OpenMP*: És l'estàndard per implementar paral·lelització intranode. Sol estar lligada al conjunt d'eines emprades. Per tant, si s'utilitzen compiladors Intel es fan servir les implementacions d'Intel i, si es fan servir els compiladors GNU s'empren les implementacions GNU.
3. Llibreries MPI: És l'estàndard per implementar paral·lelització internode. Sol estar lligada també al conjunt d'eines emprades. Per tant, si s'utilitzen compiladors Intel, es fan servir les implementacions d'Intel i si es fan servir els compiladors GNU, s'empren les implementacions GNU. També, ja que hi ha programari que empra algun compilador amb una implementació MPI diferent a la que hi ha per defecte, s'han incorporat al clúster diferents implementacions, i estan disponibles amb el seu respectiu "module".
4. Llibreries de caire científic: Són llibreries que les aplicacions empren per executar-se. També disposen del seu "module" per carregar-les a l'entorn.

Per compilar programes en alguna implementació MPI, primer heu de carregar l'entorn pertinent, i llavors compilar. Per exemple, per compilar un programa amb l'entorn de *bullx-compilers*, primer heu de fer:

```
$ module load foner-modules/bullx-compilers
$ mpicc <nom_fitxer.c>
```

Per altre banda, si es requereix l'ús d'*OpenMp*, ho fareu posant un paràmetre al compilador.

```
$gcc -fopenmp <nom_fitxer.c>
```

Consultau el manual del compilador per a més informació.

7 Contacte

Suport: suport.hpc@uib.es, extensió 2582