CLUSTER DI CALCOLO DI OAS-BOLOGNA E SOFTWARE SCIENTIFICO DISPONIBILE

Autori

Fulvio Gianotti <fulvio.gianotti@inaf.it>

Alessandro Tacchini <alessandro.tacchini@inaf.it>

Adriano De rosa <adriano.derosa@inaf.it>

Daniela Paoletti <daniela.paoletti@inaf.it>

Eleonora Torresi <eleonora.torresi@inaf.it>

Coautori (ordine Alfabetico)

Andrea Bulgarelli <andrea.bulgarelli@inaf.it>

Vito Conforti <vito.conforti@inaf.it>

Mauro Dadina <mauro.dadina@inaf.it>

Giovanni De Cesare <giovanni.decesare@inaf.it>

Fabio Finelli <fabio.finelli@inaf.it>

Valentina Fioretti <valentina.fioretti@inaf.it>

Enrico Franceschi <enrico.franceschi@inaf.it>

Paola Grandi <paola.grandi@inaf.it>

Valerio Pastore <valerio.pastore@inaf.it>

Tutti di OAS-INAF Via Gobetti, 101 40129 Bologna

ABSTRACT

All'interno dell' Osservatorio di Astrofisica e Scienza dello Spazio di Bologna (OAS), da sempre abbiamo avuto la necessità di strumenti di calcolo potenti e condivisi su cui gli scienziati possano svolgere le elaborazioni sui dati scientifici di Progetti e Missioni.

Il Cluster OAS nasce proprio per soddisfare questa esigenza ed è ospitato nel Centro di Calcolo del plesso OAS presso il CNR di Bologna. Tutti gli afferenti a OAS possono chiedere di essere registrati

nel sistema di autenticazione LDAP del Cluster e in questo modo accedere via Internet ai nodi di login da dove possono sottomettere le loro elaborazioni.
Ci sono due modalità di usare il cluster: quella interattiva e quella batch più classica per un Cluster. L'accesso interattivo consente di lanciare in tempo reale le elaborazioni anche in maniera grafica tramite opportuni programmi si console virtuale, questo tipo di elaborazione viene svolta direttamente nei nodi di login. La modalità batch sfrutta il meccanismo a code, slurm, per sottomettere i lavori in maniera organizzata ai più potenti nodi di calcolo che non sono direttamente utilizzabili dagli utenti.
Il cluster inoltre fornisce spazio di archiviazione condiviso organizzato in HOME per i dati personali degli utenti, DATA per i risultati delle elaborazioni PROGRAMMI per la memorizzazione dei moduli di elaborazione e LUSTRE per il calcolo parallelo.
I vantaggi di avere un Cluster sono che gli utenti trovano i programmi e compilatori di cui necessitano già installati nelle principali versioni e hanno a disposizione una buona potenza di calcolo e spazio di storage per poter lavorare molto più agevolmente rispetto ai propri computer personali.
Il Cluster OAS non ha una potenza paragonabile ai grandi Cluster Commerciali e di Ricerca, ma essendo ritagliato sulle esigenze degli afferenti a OAS risponde bene alle esigenze dell'istituto e può servire come Nave Scuola per poter poi accedere a strutture più grandi qualora fosse necessario.
Nel documento saranno illustrate nel dettaglio le caratteristiche Hardware e Software del Cluster, compresi tutti i principali Software scientifici installati di cui si spiega brevemente l'utilizzo.

List Of Images

List Of Tables

1 INTRODUZIONE

OAS-Bologna è dotato di un centro di calcolo all'avanguardia necessario per le molteplici attività di ricerca che vengono svolte all'interno dell'Istituto. Tale cluster, situato nel plesso CNR, è accessibile dagli utenti dell'OAS facendo il login sui server 'front-end' login01-login02-login03 con il proprio username e password. Per ottenere tali credenziali è sufficiente spedire una E-Mail a sid.oas@inaf.it con il proprio Nome e Cognome ed eventualmente una breve descrizione del perchè l'utente necessita di usare il Cluster e di che SW necessita per le sue attività.
Su tale cluster sono installate librerie pubbliche come GCC, librerie MPI, librerie di calcolo scientifico o di accesso ai dati scientifici (come CFitsIO e HEALPix).
Molti dei software pubblici per l'analisi dei dati scientifici sono correttamente installati sul cluster e pronti ad essere utilizzati dagli utenti. In questo modo ogni utente può ridurre e analizzare i dati provenienti da differenti satelliti senza dover necessariamente installare sulla propria macchina il software appropriato, ma semplicemente inizializzando le variabili d'ambiente per una specifica analisi. Questo permette di risparmiare tempo e garantisce l'utilizzo di un software, e dei relativi file di calibrazione, sempre aggiornati.

Il presente documento è strutturato come segue:

- in Sezione 2 viene presentato lo stato dell'arte del cluster di calcolo (architettura generale, caratteristiche tecniche e sistemi operativi dei nodi di login, modalità di accesso ai nodi);

- la Sezione 3 è dedicata al software scientifico presente sul cluster di calcolo. In particolare, i pacchetti di riduzione e analisi dati installati sono elencati e descritti insieme con l'ambiente di sviluppo del cluster.

2 CLUSTER DI CALCOLO

Di seguito verrà presentato lo stato dell'arte dell'infrastruttura hardware e software del centro di calcolo di OAS-Bologna. Esso è dotato di un cluster di calcolo dedicato all'analisi scientifica, allo *storage* dei dati e allo sviluppo software, e di un cluster di macchine virtuali dedicato all'implementazione dei servizi come il servizio di autenticazione (LDAP), un server di posta per i servizi interni (allarmi, etc.), webmail, workstation virtuali, cloud storage, server di progetto. Tali macchine virtuali sono relative a progetti e/o task specifici, in gestione al centro di calcolo.

2.1 Architettura generale del cluster di calcolo

Il cluster di calcolo è costituito da:

- 3 nodi di login le cui caratteristiche sono descritte in 2.3. Caratteristiche tecniche e sistemi operativi dei nodi di login:
  - login01.oas.inaf.it
  - login02.oas.inaf.it
  - login02.oas.inaf.it ←- Disponibile a breve

A questi nodi è possibile accedere da Internet come descritto nel paragrafo 2.6. Modalità di accesso ai nodi di login. Una volta effettuato l'accesso a un nodo di login sarà possibile accedere ai nodi di calcolo utilizzando il gestore delle code come descritto in 4 Utilizzo del Cluster di Calcolo.
I nodi di calcolo sono nella rete nascosta 192.168.210.0/24

- 3 nodi di calcolo con 64 core ciascuno (gestiti da uno *scheduler* "IBM loadlever"), tutte le caratteristiche in 2.4. Caratteristiche tecniche e sistemi operativi dei nodi di Calcolo;
  - node01 -> 192.168.210.1
  - node02 -> 192.168.210.2
  - node04 -> 192.168.210.4
  - node05 -> 192.168.210.5
  - sbuccia -> 192.168.210.6

- nodi di *storage,* le cui caratteristiche sono descritte in 2.5. Caratteristiche tecniche e sistemi operativi dei nodi di Storage
  - storage01 -> 192.168.210.34
  - storage02 -> 192.168.210.33
  - storage03 -> 192.168.210.36

A questi server non si accede direttamente, ma il loro spazio disco viene reso disponibile ai nodi di login e ai nodi di calcolo come descritto in 2.2. Organizzazione dello spazio disco nel Cluster

Questi server sono tutti rappresentati in **Figura 1**. I nodi sono collegati tramite rete ethernet a 1Gbit/s e con rete Infiniband a 20Gbit/s. Le parti tratteggiate in rosso rappresentano quelle parti che non sono ancora state migrate e/o hanno problemi HW in via di soluzione.

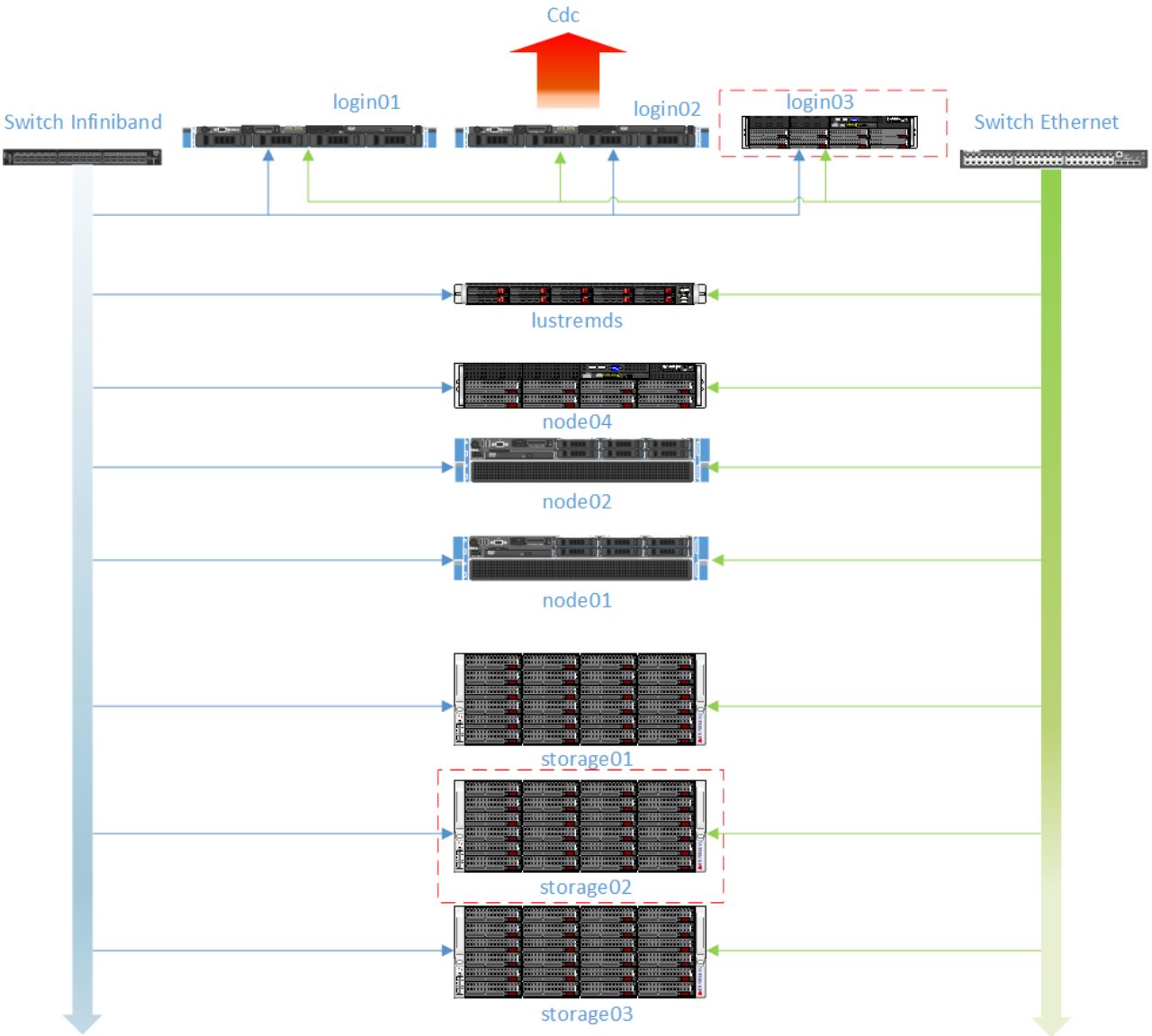

Figure 1. Architettura del Cluster

2.2 Organizzazione dello spazio disco nel Cluster

Il Cluster OAS, oltre che mettere a disposizione degli utenti le risorse di calcolo schematizzate sopra fornisce anche lo spazio disco per memorizzare i propri dati, immagini, documenti, risultati delle elaborazioni ecc..
Lo spazio disco è organizzato in diverse sezioni:

- ***/home*** montata via NFS da tutti i server del cluster. E' lo spazio dedicato per i dati e i settaggi utente. Sono 15TB disponibili in totale. Questo è uno spazio disco molto importante, perché quando si fa login nel cluster ci si trova in questa directory. Per esempio l'utente con username *gianotti* dopo il login si troverà in */home/gianotti.* Di questo spazio è garantito il backup ed è già attivo. Non è da utilizzare direttamente per far girare i calcoli scientifici, ma eventualmente può essere l'area dove spostare i risultati più importanti.
- ***/prod_oasbo*** sono circa 20TB di spazio dove sono installati i programmi che girano sul cluster. Montato via NFS.
- ***/blasco*** è il punto di accesso al filesystem LUSTRE utilizzato dal nuovo sistema basato su CentOS7.x. *blasco* garantisce 130TB (che sarà presto espanso a 170TB). Questo spazio disco è utilizzato per il calcolo scientifico. In */blasco/users/* ogni utente ha uno spazio da

utilizzare per i dati da elaborare e i risultati. Questo FileSystem è visibile da tutti i nodi di login e di calcolo. Su questo filesystem non vengono fatti backup.

- ***/data*** filesystem di 22TB dedicato ai risultati e di cui sarà garantito il backup - questo per garantire l'integrità dei risultati in caso di indisponibilità del filesystem LUSTRE. Montato via NFS. Di questo spazio disco saranno fatti i backup appena possibile.
- ***/mission*** filesystem dedicato ai dati delle Missioni di 44TB ed e' montato via NFS.

2.3 Caratteristiche tecniche e sistemi operativi dei nodi di login

I nodi di login si differenziano tra loro in base all'utilizzo che l'utente ne vuole fare, in modo da poter distribuire il carico di lavoro il più equamente possibile tra i nodi. I nodi di login sono basati su su CentOS7 e LUSTRE. Il nodo login03 è ancora basato su CentOS6.9 , ma sarà anche lui passato a CentoS7.9 (quindi, al momento, si consiglia di non utilizzarlo).

Nodo	**Caratteristiche tecniche**	**Sistema operativo Linux** *(RedHat like)*
login01	8 core amd , 32GB ram	CentOS 7.9
login02	8 core amd, 32GB ram	CentOS 7.9
login03 (in upgrade)	16 core Intel Xeon, 18GB ram	CentOS 6.9

Table 1. Caratteristiche Nodi di LOGIN

2.4 Caratteristiche tecniche e sistemi operativi dei nodi di Calcolo

I nodi di calcolo sono quelli in cui viene svolta la parte di calcolo e sono controllati dallo scheduler.

Nodo	**Caratteristiche tecniche**	**Sistema operativo Linux** *(RedHat like)*
node01	64 core amd, 128GB ram	CentOS 7.9
node02	64 core amd, 128GB ram	CentOS 7.9
node04	64 core amd, 256GB ram	CentOS 7.9
node05	40 core Intel, 128GB ram	CentOS 7.9
sbuccia	112 core Intel 128GB ram	CentOS 7.9

Table 2. Caratteristiche NODI di Calcolo

2.5 Caratteristiche tecniche e sistemi operativi dei nodi di Storage

I nodi storage sono i nodi dove vengono memorizzati i dati, che li rendono poi disponibili agli altri nodi sotto forma di filesystem LUSTRE o NFS. I Filesystem vengono montati nei nodi di login e di Calcolo e sono accessibili in ben definiti mount point elencati sotto:

- /Blasco LUSTRE
- /home NFS
- /data NFS
- /prod_oasbo NFS
- /mission NFS

Nodo	Caratteristiche tecniche	Sistema operativo Linux *(RedHat like)*	Storage Netto - Filesystem
nashome	16 Core intel xeon 6GB ram	CentoS 7.x	15TB /home
storage01	8 core intel xeon, 16GB ram	CentOS 6.x	48 TB
storage02 (In riparazione	8 core intel xeon, 16GB ram	CentOS 6.x	30 TB
storage03	32 core amd, 64GB ram	CentOS 7.x	99TB /Blasco

Table 3. Caratteristiche NODI di Storage

2.6 Modalità di accesso ai nodi di login

E' possibile effettuare l'accesso ai nodi di login in tre modalità:

1) SSH senza X forwarding

ssh username@login01.oas.inaf.it
ssh username@login02.oas.inaf.it
ssh username@login03.oas.inaf.it

in questa modalità l'utente non può utilizzare la finestra grafica, ma solo un terminale. Nei sistemi Linux e macOS il terminale è un programma che è sempre presente di default. Per windows un ottimo programma che si può utilizzare gratuitamente è PUTTY o MobaXTerm che fornisce in aggiunta un'interfaccia unix per windows utile ad esempio per fare backup etc.

2) SSH con X forwarding

ssh -X username@login01.oas.inaf.it
ssh -X username@login02.oas.inaf.it
ssh -X username@login03.oas.inaf.it

In alcuni sistemi tipo MacOS invece di -X occorre usare -Y che permette un forwarding sicuro della grafica X11. Quindi se si vede che -X non funziona si può utilizzre -Y a lato pratico sono equivalenti.

in questa modalità l'utente può utilizzare la finestra grafica se nel suo PC ha installato un opportuno client X:

- Per sistemi Linux il Client X è presente di default in tutte le distribuzioni
- Per i sistemi macOS il client consigliato è XQUARTZ
- Per i sistemi Windows il client consigliato è XMING o MobaXTerm

2.7 Modalità di accesso ai nodi di calcolo

Benchè sia al momento possibile accedere tramite SSH ai nodi di calcolo, questa è una pratica da non seguire, perchè lanciare dei processi di calcolo direttamente facendo SSH fa si che questi non siano considerati dallo scheduler e quindi si rischia di sovraccaricare i nodi di calcolo senza controllo.
Per accedere correttamente ai nodi di calcolo occorre usare la seguente istruzione dai nodi di login:

srun --x11 --nodes=1 --ntasks-per-node=1 --time=01:00:00 -p large --pty bash -i <Comando>

Dove:

- *--x11* serve a redirigere l'output grafico verso l'utente
- *--nodes==#* numero di nodi su cui girerà il comando
- *--ntasks-per-node=#* numero di task per nodo
- *--time=hh:mm:ss* tempo di esecuzione
- *-p large* partizione su cui si intende chiedere le risorse
- *--pty bash -i* serve per scegliere la shell

Per esempio:

per richiedere una sessione interattiva di 5 ore sulla partizione large con due processori occorre utilizzare il seguente comando:

srun --x11 --nodes=1 --ntasks-per-node=2 --time=05:00:00 -p large --pty bash

3 SOFTWARE SCIENTIFICO

OAS-Bologna è coinvolto in numerose missioni da Terra e dallo spazio, che operano nelle diverse bande dello spettro elettromagnetico. In generale, ogni missione ha il proprio pacchetto di riduzione dati e le proprie calibrazioni che devono essere aggiornate periodicamente per poter effettuare una corretta analisi dei dati scientifici.
L'utilizzo del software scientifico direttamente dal cluster di calcolo ha il vantaggio, per l'utente, di non dover installare ogni pacchetto sul proprio pc, avendo la certezza di lavorare con versioni del software sempre aggiornate. Inoltre, ciascun utente può richiedere l'installazione di un nuovo pacchetto scientifico, laddove non fosse già presente.

Date le caratteristiche delle macchine, il software per l'analisi scientifica di alta energia e' attualmente installato su login01, login02, login03. Di seguito viene riportato l'elenco dei pacchetti per la riduzione e analisi dei dati scientifici attualmente installati sul cluster di calcolo:

Nodo di login	Sistema operativo	Software Scientifico/Versione
login01, login02	CentOS7.9	**HEASOFT v. 6.29** **CIAO v. 4.13** **w/ CALDB v. 4.9.4** **SAS v.19.1** **ds9 v 8.1** **OSA v. 11.1** **TOPCAT**

Table 4. Elenco Software Scientifici

Di seguito viene riportata una lista dei compilatori, delle librerie e dei software attualmente presenti sul cluster di calcolo:

Compilatori e librerie MPI	GCC
	Scalasca
	Intel
	openMPI

Table 5. Elenco Compilatori

	Scalapack

Librerie di calcolo scientifico	Lapack
	MKL
	GSL

Table 6. Librerie di Calcolo Scientifico

Librerie per accesso ai dati astrofisici (I/O)	HEALPIX
	Cfitsio
	HDF5

Table 7. Librerie per accesso a dati Astrofisici

Software di visualizzazione dati scientifici	Paraview
	Paraview MPI
	Ds9
	pyDs9
	Gnuplot
	Grace, Xmgr
	smongo

Table 8. Software di Visualizzazione dati Scientifici

	IDL
	Geant4

Table 9. Software di Calcolo e Simulazione

3.1 Ambiente di sviluppo del cluster di calcolo

3.1.1 Settaggio dell'ambiente

L'ambiente di sviluppo del cluster è organizzato in maniera modulare. Questo permette di personalizzare in maniera rapida il proprio ambiente di lavoro specificatamente per i software che si devono utilizzare/sviluppare.
Caricando un modulo non si fa altro che inizializzare con una singola istruzione tutta una serie di variabili d'ambiente che permettono di utilizzare lo specifico linguaggio/compilatore/software del modulo.
Sul cluster sono presenti diversi moduli che possono essere gestiti nella propria finestra usando i seguenti comandi:

- > **module av** permette di visualizzare tutti i moduli presenti sul cluster. I moduli hanno all'interno dei loro nomi la versione del compilatore/linguaggio/software che viene caricato;
- > **module av parola_chiave** esempio "module av mpi" mostra tutti i moduli che hanno la parola chiave nel nome;
- > **module display nome_modulo** permette di visualizzare lo script del modulo. Questo comando può essere utile nel caso di alcuni moduli contenitore che caricano diversi compilatori/software insieme che potrebbero velocizzare la preparazione dell'ambiente (es. ADRIANO3 creato per i codici di cosmologia che carica compilatore intel, mpi, openmp etc.);
- > **module load nome_modulo** questo è il comando per caricare il modulo che si è scelto. Prestare attenzione che alcuni moduli sono interdipendenti, come ad esempio alcune librerie python. In questo caso, si suggerisce la creazione di uno script in cui si caricano tutti i

moduli necessari in modo da comprimere tutto in una singola istruzione "> source script.sh" ed evitare dimenticanze (se i moduli necessari sono sempre gli stessi tale script può essere caricato direttamente nella shell bash in automatico, ma è consigliabile solo per utenti esperti);

- > **module unload nome_modulo** è il contrario di module load, semplicemente smonta il modulo, è utile se uno dei moduli caricati non è più necessario o è incompatibile con altri che servono in quel momento;
- > **module purge** permette di smontare tutti i moduli montati in precedenza. Per quanto radicale, il module purge è una soluzione auspicabile quando si debba iniziare una compilazione di un nuovo codice che richieda un diverso ambiente di sviluppo rispetto a quanto usato prima, o si verifichino delle incompatibilità in fase di compilazione o run. Smontando tutti i moduli caricati, il module purge permette di ripartire ex-novo evitando il classico errore da modulo che ci si era dimenticati di aver montato che si dimostra incompatibile con quello necessario per il nuovo codice.

- > **module list** permette di visualizzare tutti i moduli caricati.

3.2 Software per analisi dei dati scientifici in banda X

Il sistema di calcolo dell'OAS fornisce capacità di calcolo per l'analisi dei dati astrofisici di alta energia. Attualmente, come riportato in Tabella 4, sono installati i pacchetti software per analizzare i dati di tutti i principali satelliti in orbita. Questo include anche i relativi file di calibrazione. Non tutte le missioni "passate", tuttavia, sono incluse nel database dei file di calibrazione. In caso di necessita' di qualche particolare pacchetto mancante, ovviamente si puo' richiedere che questo venga installato.

La semplice inizializzazione di **HEASoft** permette di installare il s/w scientifico necessario per analizzare i dati di molti telescopi. Di seguito viene riportato un esempio di come inizializzare l'ambiente per la riduzione dati dei satelliti XMM-Newton (**SAS**) e Chandra (**Ciao**).

Login al cluster e ai nodi di calcolo:

```
> ssh -X username@login02.oas.inaf.it
>  srun  --x11  --nodes=1  --ntasks-per-node=2  --time=05:00:00  -p
large --pty bash
```

Se si opera in bash shell (o simili) i comandi da dare, sono, in sequenza quelli riportati qui sotto (si consiglia di includerli in un file di inizializzazione):

```
> module load heasoft-6.29
> module load gcc-9.4
> heainit

> module load sas-19
> export SAS_CCFPATH=/prod_oasbo/CALDB/data/xmm
> export SAS_PATH=/prod_oasbo/sas/xmmsas_20210317_1624/
> source $SAS_PATH/sas-setup.sh
> sas
```

Se si opera in C-shell (o simili) i comandi da dare, sono, in sequenza quelli riportati qui sotto (si consiglia di includerli in un file di inizializzazione)

>module load gcc-9.4
>setenv HEADAS /prod_oasbo/heasoft/heasoft-6.29/x86_64-pc-linux-gnu-libc2.17
>source $HEADAS/headas-init.csh

>setenv SAS_DIR /prod_oasbo/sas/xmmsas_20210317_1624/
>setenv SAS_CCFPATH /prod_oasbo/CALDB/data/xmm
>setenv SAS_PATH $SAS_DIR
>setenv LIBRARY_PATH $SAS_DIR/lib
>setenv LIBRARY_PATH $SAS_DIR/libextra:$LIBRARY_PATH
>source $SAS_DIR/sas-setup.csh

Inizializzazione ambiente Chandra (CIAO)

Se si opera in bash shell (o simili) i comandi da dare, sono, in sequenza quelli riportati qui sotto (si consiglia di includerli in un file di inizializzazione)

```
> module load ciao-4.13
> alias ciao="source /prod_oasbo/ciao/413/ciao-4.13/bin/ciao.sh"
> ciao
```

Se si opera in C-shell (o simili) i comandi da dare, sono, in sequenza quelli riportati qui sotto (si consiglia di includerli in un file di inizializzazione)

```
>module load ciao-4.13
>alias ciao "source /prod_oasbo/ciao/413/ciao-4.13/bin/ciao.csh"
>ciao
```

Tutti i comandi possono anche essere raccolti in un unico file di inizializzazione che viene chiamato una volta sola, e i vari pacchetti richiamati attraverso gli alias.

3.2.1 Gestione, attivazione e linking dell'ambiente python

Il software scientifico include un modulo python 3.x con installati i più comuni moduli per lo sviluppo di codice scientifico in ambiente python:

- numpy
- matplotlib
- scipy
- astropy

Su richiesta si potrebbe installare anche un modulo python 2.7, ma è da notare che questa versione di python non è più supportata (EOL) dal 1 gennaio 2020.
Per caricare l'ambiente python, si applicano le stesse istruzioni del precedente paragrafo.
Se si installa un software scientifico che fornisce anche un wrapper python e la possibilità di usare le sue librerie all'interno di uno script python (e.g. Xspec e pyXspec all'interno di HEASOFT), è necessario caricare il modulo python per lo sviluppo scientifico in modo da collegare l'installazione

allo stesso ambiente in cui sono installati gli altri moduli python, invece della versione di python fornita dal sistema.
Per utilizzare la libreria python fornita dal software scientifico, è necessario caricare il modulo python per lo sviluppo scientifico.

4 Utilizzo del Cluster di Calcolo

In questo paragrafo viene spiegato come utilizzare il cluster

4.1 Nodi di login

Come specificato in precedenza, i nodi di login devo essere utilizzati per job (eseguibili) seriali o di breve durata o che non carichino eccessivamente i core dei login in modo da non rallentare gli altri utenti che accedono alla macchina proprio tramite questi nodi.
Eventualmente fare uso del comando per il numero di threads
export OMP_NUM_THREADS=num core che si vuole usare
per non occupare tutti i core del nodo di login.
Per verificare l'utilizzo in termini di carico per la macchina si possono usare le istruzioni
> top

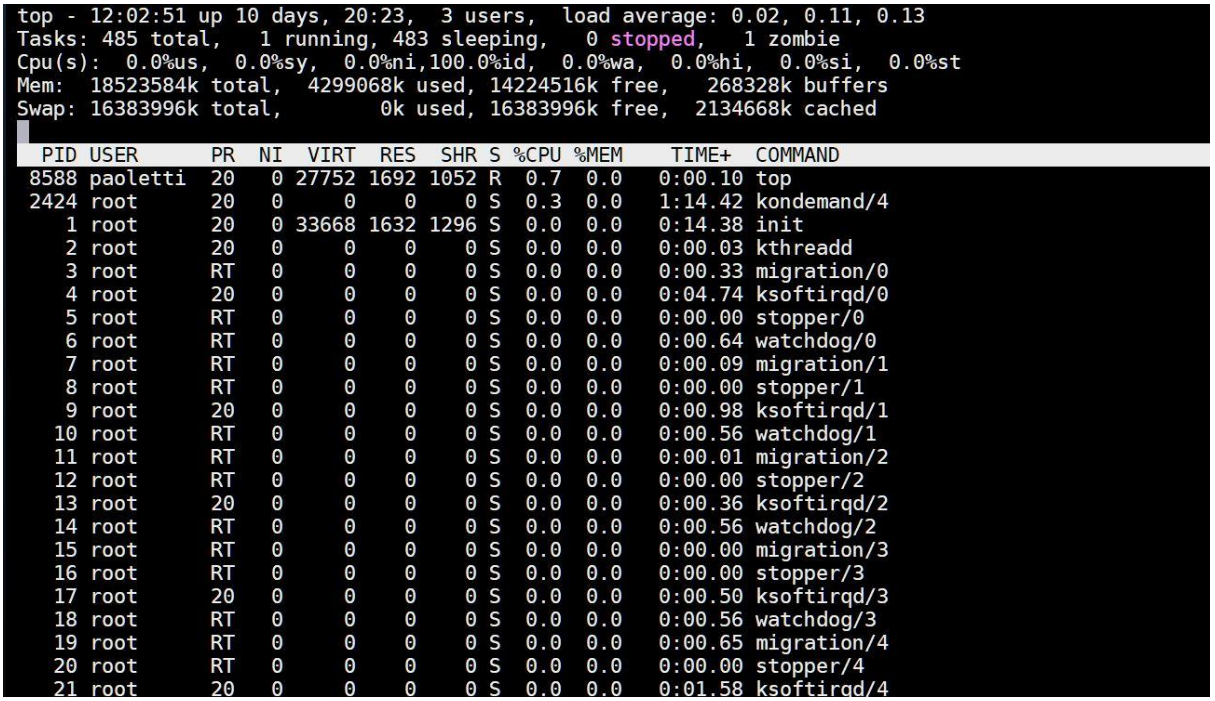

Figure 2. Carico nei Nodi di Calcolo

in cui vengono visualizzati tutti i job running sul login dove siete in ordine decrescente di occupazione e da cui poi si esce con
>q
oppure
> ps -au nome_utente per vedere solo i vostri job.

Nello sfortunato caso di impallamento di un job che cessi di rispondere ad ogni possibile stimolo (classici > ctrl c risultano inefficaci) si può procedere con la soluzione
> kill numero_PID
se anche ciò non bastasse e il vostro processo risultasse ancora attivo (dategli qualche secondo per tentare lo shutdown) come soluzione estrema
>kill -9 numero_PID
in questo caso il kill del job è brutale ergo perderete tutti eventuali risultati e metaoutput che il job stava producendo. Usatelo con cautela.
Nel caso si voglia procedere con un job sul nodo di login ma non si voglia che questo venga interrotto quando chiudete la shell è possibile utilizzare l'istruzione

4.2 Sottomissione job in nodi di calcolo

Come detto i nodi di login sono perlopiù per la gestione di script, jobs e software seriali e che non richiedano alte performance a livello di calcolo o l'utilizzo di calcolo parallelo. Per il calcolo parallelo o ad alta performance è necessario sottomettere i job sui nodi di calcolo che sono molto più performanti rispetto ai nodi di login.

Come scritto i nodi di calcolo montano uno scheduler slurm. Per il calcolo parallelo ricordatevi di caricare i moduli necessari affinché la compilazione del codice che volete utilizzare avvenga effettivamente in maniera tale da runnare i job in parallelo. Ad esempio per quanto riguarda intel si parla dei moduli mpi per i parallelizzare multiprocessi e openmp invece per parallelizzare i threads, mpi4py per il multiprocesso di python etc.. A seconda del codice/software utilizzato dovreste trovare le specifiche richieste.

Di sotto si riporta un esempio di script di sottomissione. Sono evidenziati in colore diverso e commentate le righe che devono essere personalizzate sul vostro job.
Per una lista completa dei comandi da aggiungere allo script, si rimanda alla documentazione ufficiale[1].

Uno script di esempio in slurm con 4 processo on 2 threads ciascuno (8 core di occupazione complessiva) è

```
#!/bin/bash
##Indicazione del Walltime
#SBATCH --time=24:00:00
#Numero di nodi da utilizzare come sopra si consiglia salvo prevenire prima gli altri utenti di non richiedere più di un nodo essendo le risorse limitate. Va indicato anche il sockets per node che nel nostro caso è 1
#SBATCH --nodes=1
#SBATCH --sockets-per-node=1
##Numero di processi totali che volete per mpi
#SBATCH -n 4
##Questa istruzione vi riporta praticamente il numero di threads, ovvero il numero di core che assegnate al processo praticamente dipende da quanto è parallelizzato il codice internamente ad esempio in openmp. Con questo scheduler dovete specificare anche che a ogni thread date un core.
#SBATCH --cores-per-socket=2
#SBATCH --threads-per-core=1
#questo è il nome della partizione dove runnate che nel nostro caso è large
#SBATCH -p large
##Nome job e file di output ed error
#SBATCH --job-name=CC
#SBATCH --err=CC.err
#SBATCH --out=CC.out

## [OPZIONALE] per attivare l'invio di e-mail alla fine del run, aggiungere allo script
#SBATCH --mail-type=ALL
#SBATCH --mail-user=yourname(at)inaf.it


##PARTE IN CUI CARICATE I MODULI LE VARIABILI DI AMBIENTE CHE VI SERVONO

module load XXX
source XYZ.profile
```

[1] https://slurm.schedmd.com/documentation.html

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:XXXX/lib

##SPECIFICATE NUOVAMENTE IL NUMERO DI THREADS PER OPENMP (CHE HA DA ESSERE IL
cpus-per-task)
export OMP_NUM_THREADS=2

   module list

   echo $PATH
   echo $LD_LIBRARY_PATH

date

cd  CARTELLA IN CUI SIETE

STRINGA ESEGUIBILE
MPIRUN/MPI4PY/ETC. ./XXXX XYZ.INI

date
```

Una volta creato il vostro script che vi consigliamo di puntare come nome_file.slurm per riconoscerlo da eventuali script di sottomissione con altri scheduler (nel caso soprattutto si rimpallino i codici tra diverse macchine esterne a OAS) il comando di sottomissione è

> sbatch script.slurm

Per controllare lo status dei job sui nodi

>squeue

da cui vi apparirà una schermata in cui avrete visualizzati tutti i job

```
Id                    Owner      Submitted   ST PRI Class        Running On
--------------------- ---------- ----------- -- --- ------------ -----------
node02.699.0          paoletti    6/11 16:19 R  50  large        node01
login01.1950.0        paoletti    6/11 16:19 R  50  large        node02
node01.696.0          paoletti    6/14 16:19 R  50  large        node02
login03.eth.942.0     paoletti    6/15 10:49 R  50  large        node02
login01.1974.0        ballardini  6/18 11:16 R  50  large        node01
login01.1975.0        ballardini  6/18 11:16 R  50  large        node02
login01.1976.0        fioretti    6/18 11:54 R  50  large        node01
login01.1977.0        fioretti    6/18 11:56 R  50  large        node01

8 job step(s) in queue, 0 waiting, 0 pending, 8 running, 0 held, 0 preempted
```

Figure 3. Visualizzazione JOB co squeue

Id è il nome del job sulla macchina, ST è lo status (R=running, P=pending, H=held, ST=starting), viene comunicato anche il nodo su cui sta runnando il job.

Per cancellare un job

>scancel Id

I file di output e di error sono prodotti nei .out e .err specificati nello script di sottomissione (in quello riportato qui contengono il numero del job).

Al momento le code sono autogestite il che ovviamente richiede una autoregolazione da parte degli utenti. Lo strumento di monitoraggio squeue è sempre la cosa migliore per vedere se ci sono utenti in coda e regolarsi sulla proprio occupazione.

Se possibile all'interno del vostro codice se avete dei run molto lunghi predisporre sempre per dei punti di ripristino e/o checkpoint che vi garantiscano di non perdere il lavoro in caso di interruzione del cluster per eventi straordinari.

4.3 Ambiente di sviluppo codici di predizioni cosmologia e struttura su grande scala

Si illustra ora brevemente i software principali per le predizioni degli osservabili cosmologici e alcune linee guida per il loro utilizzo in cluster.

Boltzmann codes

Per le simulazioni osservabili CMB e struttura su grande scala i codici Einstein Boltzmann principali:

4.3.1 CAMB

https://camb.info/

Camb è un codice nativo in fortran 90, è stato recentemente modificato con un'interfaccia python ma il core fisico delle equazioni è maggiormente gestibile dalla parte fortran dei sorgenti che garantiscono anche maggiore stabilità numerica rispetto a python (con meno rischio di errori silenti quando si vanno a modificare i sorgenti) si consiglia quindi di usare i sorgenti in fortran per eventuali implementazioni. Prima di compilare bisogna settare l'ambiente caricando i moduli o INTEL o GCC.

4.3.2 CLASS

https://lesgourg.github.io/class_public/class.html

Class è un codice a base C anche qui prima di compilare bisogna caricare il compilatore o INTEL o GCC. Nell'effettuare modifiche prestare particolare attenzione. Essendo il codice strutturato con l'utilizzo copioso di puntatori assicurarsi di effettuare correttamente le disallocazioni di memoria in fase di modifica e controllare eventuali memory leakage che posso portare a un sovraccarico del sistema e quindi a serie problematiche di run per il Markov Chain MonteCarlo.
Ricordarsi che INTEL in generale performa molto meglio di GCC in termini di velocità in openmp.

4.3.3 MCMC

Per l'esplorazione dello spazio dei parametri cosmologici e analisi dati CMB e LSS.
In questo caso si utilizzano i Markov Chain MonteCarlo (o alternativamente i codici Fisher che però sono o privati o di collaborazione per cui rivolgersi agli eventuali interessati) associati alle likelihood sviluppate dalle varie collaborazioni.
I principali codici sono

4.3.4 COSMOMC

https://cosmologist.info/cosmomc/

a base fortran e agganciato al Boltzmann CAMB. Tendenzialmente molto stabile ma richiede un certo livello di sviluppo per implementare varianti. Interfacciato con i principali esperimenti CMB e LSS.

4.3.5 MONTEPYTHON

https://baudren.github.io/montepython.html

wrapper MCMC in python interfacciato a CLASS. Versioni tassativamente dalla 3.3 in poi per garantire stabilità. Molto più complesso in termini di modifiche e implementazioni essendo legato a class. Tendenzialmente più rapidamente aggiornato rispetto a cosmomc per nuove likelihood.

4.3.6 COBAYA

https://cobaya.readthedocs.io/en/latest/installation.html
Wrapper ibrido interfacciabile con CLASS e CAMB. Concept completamente diverso da cosmomc e montepython richiede input tramite yaml ed è completamente python based.
Tutti e tre i codici precedenti richiedono il settaggio sempre INTEL e GCC e in aggiunta PYTHON dovendo utilizzare gli Einstein Boltzmann.
In aggiunta essendo codici che oltre alla parallelizzazione multithreading con per esempio openmp utilizzano il multiprocessing con mpi e quindi richiedono il caricamento del modulo MPI o MPI4PY per i due codici a base python.
GLI MCMC DEVONO ESSERE SOTTOMESSI SOLO IN NODO DI CALCOLO. Il quantitativo di memoria e di cpu richiesti infatti impallerebbero i nodi di login impedendo agli altri utenti di utilizzarli. Tenere conto del livello di saturazione dello scaling mpi e openmp di tali codici che non è infinito. Salvo modifiche fisiche particolari in generale uno schema 4 proc x 2 thread è già buono per il run quadratico medio.
Come buona norma sempre ridurre gli output al minimo in quanto le procedure I/O sono le più dispendiose a livello di nodo di calcolo.

4.3.7 LIKELIHOOD:

in generale i moduli di likelihood sono forniti direttamente con interfaccia cosmomc e montepython dalle collaborazioni dei vari esperimenti come semplici plug in.
Per quanto riguarda la likelihood Planck la versione pubblica finale con anche tutte le catene utilizzabili per i confronti si trova:
/blasco/planck/
Da cui si deve solamente fare uno script clik.profile contenente l'istruzione

```
export PLANCKLIKE=cliklike
export
CLIKPATH=/prod_oasbo/planck/Clik2018_Public/plc_3.0_release/code/plc_3.
0/plc-3.0/
source
/prod_oasbo/planck/Clik2018_Public/plc_3.0_release/code/plc_3.0/plc-3.0
/bin/clik_profile.sh
```

da caricarsi sia prima della compilazione che nello script di sottomissione del job per settare le variabili di ambiente della likelihood.
Essendo il pacchetto estremamente voluminoso in termini di memoria e numero di files si consiglia di usare quello comune e non riscaricarselo in locale.

Quando verrà deliverato al pubblico analoga procedura verrà effettuata per la likelihood Euclid attualmente in fase di sviluppo interno alla collaborazione.

5 Utilizzo dell'ambiente anaconda

Sul cluster e' installata la versione 3.2021.05 di anaconda. Per utilizzarla e' necessario caricare il modulo appropriato e inizializzare l'ambiente con i seguenti comandi:

```
[root@login02 ~]# module load anaconda3-2021.05
[root@login02 ~]# anaconda_init
```

i seguenti comandi possono essere utilizzati per visualizzare i container installati e caricarli.

```
[derosa@login02]~>conda info --envs
# conda environments:
#
base             *  /prod_oasbo/anaconda/Anaconda3-2021.05
astroconda             /prod_oasbo/anaconda/Anaconda3-2021.05/envs/astroconda
iraf27                 /prod_oasbo/anaconda/Anaconda3-2021.05/envs/iraf27

[derosa@login02]~>conda activate base
(base) [derosa@login02]~>conda activate iraf27
(iraf27) [derosa@login02]~>conda deactivate
(base) [derosa@login02]~>conda deactivate
```

Nel caso sia necessario e' possibile installare, su richiesta, altri ambienti anaconda.