



**MISSION
INNOVATION**

accelerating the clean energy revolution

POA MATERIALI AVANZATI PER L'ENERGIA

**PROGETTO IEMAP - Piattaforma Italiana Accelerata per i Materiali per
l'Energia**

Istruzioni per l'accesso, l'autenticazione e l'utilizzo dell'infrastruttura CRESCO

Filippo Palombi, Simone Giusepponi, Claudio Ronchetti, Massimo Celino



ISTRUZIONI PER L'ACCESSO, L'AUTENTICAZIONE E L'UTILIZZO DELL'INFRASTRUTTURA CRESCO

Filippo Palombi (ENEA), Simone Giusepponi (ENEA), Claudio Ronchetti (ENEA), Massimo Celino (ENEA)

Maggio 2022

Report MISSION INNOVATION

Ministero della Transizione Ecologica - ENEA

Mission Innovation 2021-2024 - I annualità

Progetto: Piattaforma accelerata per i Materiali per l'Energia

Work package: IEMAP: Italian Energy Materials Acceleration Platform

Linea di attività 1.1: Ambiente di sviluppo HPC: accessi e autenticazione

Responsabile del Progetto: Massimo Celino, ENEA

Responsabile della LA: Filippo Palombi, ENEA

Indice

Sommario	4
Sinossi LA1.1.....	4
1 Introduzione.....	5
2 L’Infrastruttura CRESCO ed il data storage del progetto IEMAP	5
2.1 L’infrastruttura CRESCO	5
2.2 Gestione degli account	7
2.3 Procedura per la richiesta di credenziali di accesso.....	8
2.4 Accesso all’infrastruttura tramite SSH	9
2.5 Accesso all’infrastruttura tramite FARO	11
2.6 Accesso all’infrastruttura tramite MobaXterm	12
3 Creazione di un’area di storage per il progetto IEMAP.....	12
3.1 Trasferimento dati da/verso l’area di storage	15
4 Installazione del package di workflow AiiDA	16
4.1 Installazione del server PostgreSQL.....	19
4.2 Installazione del message broker RabbitMQ	20
4.3 Configurazione dell’infrastruttura CRESCO in AiiDA	21
5 Conclusioni	23
6 Riferimenti bibliografici	23

Sommario

Sinossi LA1.1

Scopo di questa linea di attività è la realizzazione di una piattaforma ICT che assista la selezione di nuovi materiali per l'energia, tramite sistemi computazionali e di analisi dati ad elevate prestazioni (HPC/HPDA). La piattaforma risponderà in modo integrato alle molteplici esigenze del progetto, permettendo in particolare: (i) la condivisione di grandi moli di dati sperimentali prodotti nei laboratori di ricerca sui nuovi materiali afferenti alle istituzioni che partecipano al progetto; (ii) la condivisione di grandi moli di dati sperimentali prodotti da laboratori esterni, che siano di interesse per le linee di attività sperimentali del progetto; (iii) l'analisi dei dati sperimentali basata su tecniche di intelligenza artificiale per la selezione di materiali target da studiare tramite simulazioni numeriche su macchine parallele; (iv) lo storage dei dati prodotti dalle simulazioni numeriche; (v) l'analisi dei dati prodotti dalle simulazioni numeriche e la condivisione dei risultati derivanti dalle analisi. La piattaforma utilizzerà l'infrastruttura di calcolo ad alte prestazioni CRESCO, attualmente operativa presso il C. R. ENEA di Portici. L'infrastruttura dispone di un supercalcolatore Tier-1 denominato CRESCO6, un sistema HPC costituito da 434 nodi di calcolo (20.832 core per potenza massima di 1,4 PFlop/s), che si è classificato 420° nella lista TOP500 di novembre 2018. CRESCO6 si basa sulla piattaforma Lenovo ThinkSystem SD530. L'infrastruttura CRESCO include altri sistemi HPC secondari che adottano differenti soluzioni architetturali. Tutti i sistemi HPC CRESCO sono integrati in un'infrastruttura di calcolo distribuita geograficamente denominata ENEAGRID, dislocata su 6 centri di ricerca ENEA nel territorio nazionale. Accesso e autenticazione. Preliminarmente alle attività di ricerca, ogni partecipante al progetto dovrà provvedere all'accesso alla piattaforma ICT. Ognuno riceverà un account su ENEAGRID. I dati dei partecipanti verranno organizzati in due filesystem: un filesystem AFS per l'accesso ai servizi, un filesystem IBM GPFS per le attività computazionali di alte prestazioni.

AFS fornirà un ambiente di lavoro comune per l'accesso ai servizi e risorse informatiche, indipendentemente dalla posizione fisica degli utenti. Al contempo, GPFS verrà usato principalmente per applicazioni parallele e per sfruttare al meglio la potenzialità dei cluster HPC CRESCO. AFS è un filesystem geograficamente distribuito che offre un'architettura client-server federata per il filesharing; ciò significa che è possibile usufruire delle risorse presenti in diversi domini e gestite da più server. I principali vantaggi derivanti dall'utilizzo di AFS sono, oltre all'indipendenza dalla posizione fisica e geografica dell'utente, la sicurezza e la capacità di migrazione trasparente. La sicurezza dei file condivisi in AFS è salvaguardata dal meccanismo di autenticazione e autorizzazione Kerberos 5, ed il meccanismo di access control list (ACL) è molto più potente e flessibile rispetto ai permessi standard POSIX. AFS viene usato in quanto costituisce uno spazio scalabile, distribuito e sicuro contenente i dati dei singoli utenti, e consente di condividere applicazioni e librerie nonché aree per il lavoro collaborativo con i dati prodotti nei vari progetti. Il sistema AFS risponde in modo diretto e semplice alle esigenze di data preservation riguardanti il supporto fisico dei dati (backup, disaster recovery ecc.), e mette a disposizione dei laboratori gli strumenti necessari per rispondere facilmente anche a tutte le esigenze relative all'elaborazione dei dati e alla produzione dei metadati.

1 Introduzione

L'attività infrastrutturale del progetto IEMAP è finalizzata alla predisposizione e alla manutenzione di tutti i servizi hardware/software necessari al corretto e completo sviluppo delle linee di attività proposte per l'automazione della ricerca di nuovi materiali attraverso tecniche di intelligenza artificiale. Come tale, essa costituisce un fondamento essenziale e propedeutico alla piena realizzazione delle finalità del progetto.

Per lo sviluppo dell'infrastruttura computazionale di IEMAP, si è scelto di utilizzare come base di partenza l'infrastruttura CRESCO/ENEAGRID esistente in ENEA. Tale infrastruttura è stata preliminarmente resa accessibile a tutti gli stakeholders del progetto nella forma in cui essa preesisteva al progetto stesso. Quindi si è proceduto a progettare e realizzare i servizi aggiuntivi richiesti. In particolare:

1. Sono stati creati account per l'accesso all'infrastruttura CRESCO/ENEAGRID di tutti gli stakeholder del progetto;
2. È stata svolta un'attività di formazione degli stakeholder riguardante le caratteristiche principali hardware e software dell'infrastruttura CRESCO/ENEAGRID, le modalità di accesso, l'utilizzo dei PC Cluster CRESCO, le modalità di trasferimento dati da e verso l'infrastruttura;
3. È stata predisposta un'area di data storage strutturata e dedicata esclusivamente al progetto IEMAP per la condivisione/conservazione dei dati prodotti nell'ambito del progetto;
4. È stato installato e configurato un sistema di workflow basato sul software AiiDA per l'automazione delle simulazioni numeriche per la ricerca di nuovi materiali

Nel presente documento descriviamo dettagliatamente l'attività svolta. Nella sez. 2 presentiamo brevemente l'infrastruttura CRESCO/ENEAGRID operante in ENEA [1]. Nella sez. 3 descriviamo l'area di storage predisposta per IEMAP. Nella sez. 4 descriviamo gli aspetti tecnici dell'installazione di AiiDA. Infine, nella sez. 5 riportiamo le nostre conclusioni.

2 L'Infrastruttura CRESCO ed il data storage del progetto IEMAP

2.1 L'infrastruttura CRESCO

CRESCO6 è un cluster HPC (High Performance Computing) basato su Linux Centos 7.3. Esso si inserisce nella cornice di una infrastruttura di calcolo detta ENEAGRID, progettata e realizzata per la gestione e l'utilizzo di CRESCO6 e dei suoi predecessori attualmente operativi, CRESCO4 e CRESCO5. Mostriamo in Figura 1 una immagine della macchina, installata nella sala calcolo del C. R. ENEA di Portici.

CRESCO6 è composto da 434 nodi per un totale di 20.832 core. L'hardware si basa sulla piattaforma Lenovo ThinkSystem SD530, un server a due socket ultra denso ed economico in un fattore di forma rack da 0,5U inserito in un contenitore 2U a quattro modalità. Ogni nodo è dotato di:

- 2 CPU Intel Xeon Platinum 8160, ciascuna con 24 core con una frequenza di clock di 2,1GHz;
- Una RAM di 192GB, corrispondente a 4GB/core;
- Un'interfaccia di rete Intel Omni-Path serie 100 a porta singola PCIe 3.0x16 HFA a bassa latenza. I nodi sono interconnessi da una rete Intel Omni-Path con 21 switch Intel Edge serie 100 da 48 porte

ciascuno, larghezza di banda pari a 100 Gbps, latenza pari a 100 ns. Le connessioni tra i nodi hanno una topologia ad albero grasso affusolato a 2 livelli 2:1 (Fig.2). Il consumo di energia elettrica durante carichi di lavoro di elaborazione elevati ammonta a 190kW sostenuti.



Figura 1. Il cluster HPC CRESCO

I sistemi di storage di CRESCO6 si basano su soluzioni DDN con rete basata su InfiniBand. Le specifiche tecniche sono le seguenti:

- 1 x DDN S2A9900, dimensione 600TB raw, IB/QDR, larghezza di banda I/O: 6GB/s;
- 2 x DDN SFA7700, dimensione 1PB raw, IB/FDR, larghezza di banda I/O: 11GB/s;
- 1 x DDN SFA7990, dimensione 1PB raw, IB/EDR, larghezza di banda I/O: 20GB/s.

Il file system ad alte prestazioni si basa su GPFS Spectrum Scale con 6 nodi NSD in grado di fornire I/O parallelo. I dettagli hardware dei server NSD e dello switch fabric che si connette ai sistemi di storage sono:

- 6 server NSD per GPFS basati su Lenovo ThinkSystem SR650 2U, 96GB di RAM, 2 CPU IntelXeon Gold 5518 a 2,3GHz 12 core, 1 QLogic IBA7322 QDR InfiniBand HCA (rev02), 1 Mellanox MCX454A-FCAT DUAL FDR InfiniBand HCA, 1 Intel Omni-Path HFI Silicon serie 100;
- 1 switch 1U Mellanox MSX6036F - 36 porte FDR InfiniBand per il collegamento dei server NSD ai sistemi di storage DDN;
- 1 switch 1U QLogic 12300 – 36 porte QDR InfiniBand per il collegamento dei server NSD ai cluster CRESCO 4/5.

Lo stack software principale dei sistemi multi-fabric CRESCO include: sistemi operativi, fabric InfiniBand e Omni-Path, RDMA, SRP, pacchetti Multipath e infine IBM Spectrum Scale GPFS.

La suite software di base disponibile su CRESCO6 e utilizzata nelle attività di ricerca e sviluppo è costituita da:

- Sistema operativo (SO): Linux Centos 7.3 – Kernel Linux: v.3.10.0-514.26.2.el7.x86_64
- File system distribuito: Andrew File System – openAFS v.1.6.22
- Filesystem ad alte prestazioni: IBM Spectrum Scale GPFS – v.4.2.3
- Sistema di gestione delle risorse: IBM Spectrum LFS – v.9.1
- Compilatori: GNU GCC – v.7.3.0
- Libreria MPI: OpenMPI – v.3.1.2 Libreria

L'interconnessione tra i nodi di CRESCO6 è basata su Intel Omni-Path 100Gbps. La sfida principale, per implementare il cluster CRESCO6 HPC nell'infrastruttura del data center CRESCO, è stata progettare e implementare un layout su cui il vecchio fabric InfiniBand QDR (40 Gbps) dei cluster CRESCO 4/5, inclusi gli storage ad alte prestazioni dotati di DDN con InfiniBand FDR (56 Gbps), il nuovo fabric Omni-Path (100 Gbps) del cluster CRESCO6 e il filesystem ad alte prestazioni IBM Spectrum Scale (GPFS), lavorano tutti insieme in un'unica infrastruttura [2]. I principali componenti hardware e il layout della soluzione multi-fabric sono illustrati in Figura 2.

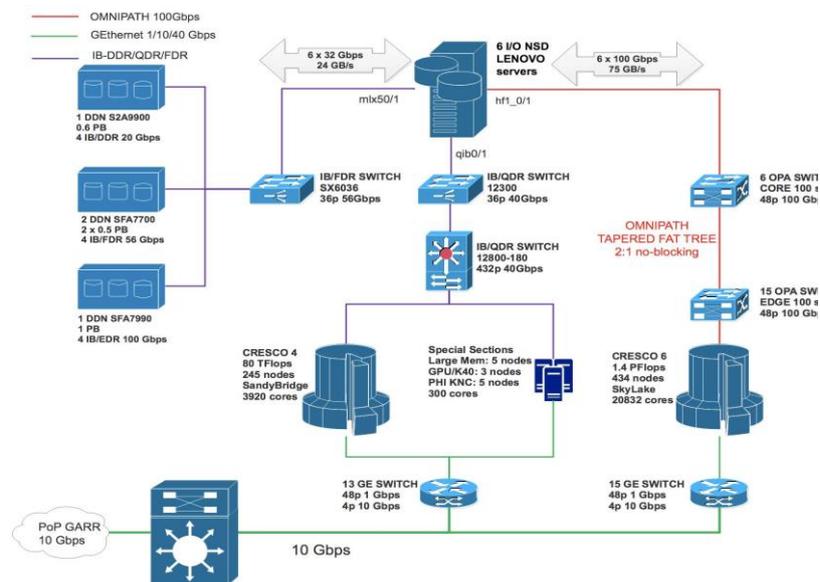


Figura 2. Layout dell'infrastruttura CRESCO

2.2 Gestione degli account

L'infrastruttura ENEAGRID è dotata di un'interfaccia web amichevole, denominata WARC, per la creazione e la gestione degli utenti. L'interfaccia è visibile solo nella intranet ENEA all'indirizzo

<https://hpc-por-s2.portici.enea.it/cgi-bin/webgentool2/login.cgi>

ovvero può essere raggiunta via link collegandosi al sito web di CRESCO,

<https://www.eneagrid.enea.it/CRESCOportal>

La schermata principale dell'interfaccia è mostrata in Figura 3. Il pannello centrale di WARC è dedicato alla gestione delle utenze ENEAGRID. L'interfaccia, sviluppata in-house dall'unità tecnica HPC, offre la possibilità di creare, eliminare ovvero modificare tutte le informazioni associate ad una data utenza.

Quale esempio, mostriamo in Figura 4 la maschera web relativa alla creazione di nuove utenze. Per ogni utenza creata, l'interfaccia richiede la specifica del server AFS più la relativa partizione disco, su cui verrà creato il volume (o i volumi) AFS necessari ad ospitare la *home directory* dell'utente, l'unità organizzativa di appartenenza dell'utente (nel caso di utenti interni ad ENEA), la quota disco associata all'utenza, la data di inattivazione automatica dell'account (per gli utenti per i quali è previsto un utilizzo solo temporaneo dell'infrastruttura), alcuni dati anagrafici, eventuali parametri per la limitazione della potenza di calcolo messa a disposizione dell'utente, una stringa di testo per tenere traccia sinteticamente dell'attività di calcolo dell'utente (progetto, collaborazione, etc.) e, nel caso di utenti esterni ad ENEA, i dati di un contatto interno all'ente che svolga le funzioni di supervisor per le attività programmate. La sottomissione del processo di creazione dell'utenza determina la creazione automatizzata dello spazio utente sui dischi AFS, di quello sui dischi GPFS, e implementa tutti i processi di inizializzazione (.login, .profile, etc.) necessari perché l'utenza possa utilizzare correttamente e interamente la piattaforma software di ENEAGRID.

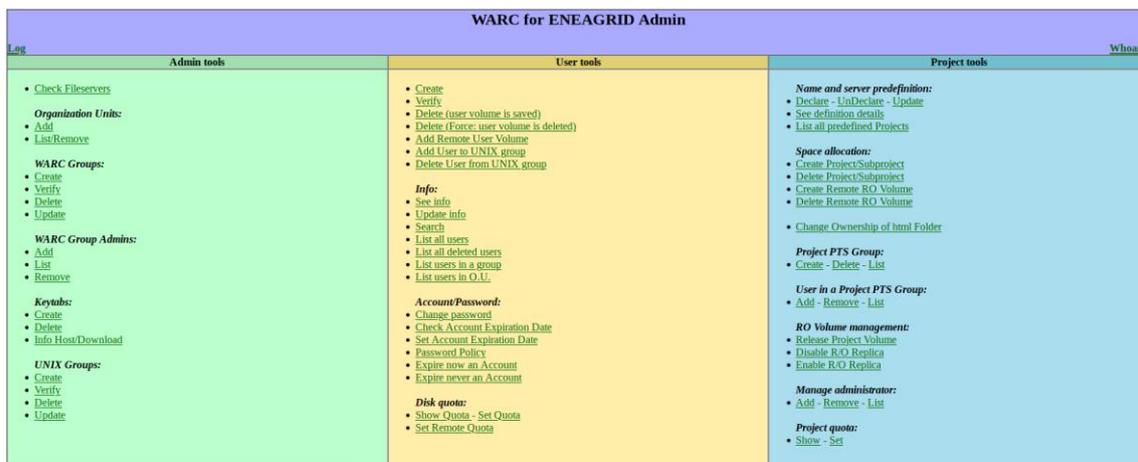


Figura 3. Schermata principale di WARC

2.3 Procedura per la richiesta di credenziali di accesso

Tutti gli utenti dell'infrastruttura ricevono credenziali di accesso previa richiesta da inoltrare per via telematica al sito

<https://gridaccount.enea.it>

Welcome to WARC tool cruser

Group Name ('por' unless explicitly requested elsewhere):	- v
User Name:	palombi
Server Name:	- v
Partition:	- v
Organization Unit:	- v
Password:
Retype Password:	
Quota: (KB)	5000000
Account Expiration Date: (gg-mm-aaaa)	never
Name:	
Surname:	
Phone: (undefined: 'empty')	
Email: (undefined: 'empty' or 'unknow')	
Shell: ('bash' unless otherwise explicitly requested)	bash v
MonthlyMaxWallClockTime (core*hour):	empty
Info: (undefined: 'empty')	empty
*** ENEA Supervisor - to be ONLY & ALWAYS defined for non ENEA employees	
Name : (undefined: 'empty')	empty
Surname: (undefined: 'empty')	empty
Phone: (undefined: 'empty')	empty
Email:(name.surname@enea.it or 'empty')	empty

Figura 4. Creazione di un'utenza tramite WARC

La procedura prevede il completamento di una form online, illustrata in Figura 5, l'invio di una relazione esplicativa dell'attività computazionale proposta e la sottoscrizione di un elenco di condizioni volte a garantire

- un utilizzo dell'infrastruttura conforme con le finalità dichiarate e nei limiti previsti dai sistemi di scheduling installati per la gestione simultanea e concorrentiale di molti utenti
- un uso responsabile dello spazio dati fornito
- l'impegno tassativo a citare/ringraziare lo staff che gestisce l'infrastruttura in ogni pubblicazione, di qualsivoglia natura, contenente risultati ottenuti attraverso l'uso dell'infrastruttura stessa.

2.4 Accesso all'infrastruttura tramite SSH

La modalità standard per accedere all'infrastruttura prevede l'uso di un client SSH. Gli utenti che si avvalgono dei sistemi operativi Linux o macOS possono effettuare la connessione direttamente da linea di comando tramite console, mentre gli utenti che usano ordinariamente MS-Windows devono necessariamente scaricare e installare uno dei tanti client disponibili su Internet. Tra questi, i più noti sono

- KiTTY
- Solar PuTTY, SuperPuTTY, PuTTY Tray, ExtraPuTTY
- MobaXterm
- WinSCP
- SmarTTY

- Zoc



Figura 5. Form di richiesta credenziali ENEAGRID

Per entrambe le tipologie di utenti, l'infrastruttura prevede due soli server di accesso, visibili globalmente, uno fisicamente installato nel C.R. ENEA di Portici, l'altro in quello di Frascati:

- cresco-in.portici.enea.it
- cresco-in.frascati.enea.it

Su entrambi i nodi di accesso è presente un server SSHD operante sulla porta 22 (standard per SSH), programmato in modo l'X-tunnelling per l'uso di applicazioni grafiche in remoto. Come esempio, riportiamo il comando di connessione al nodo di portici, su Linux:

```
ssh -X username@cresco-in.portici.enea.it
```

I server rispondono al tentativo di connessione attraverso autenticazione basata su Kerberos5. Previo inserimento della password associata all'account 'username', i server di accesso propongono un elenco di front-end per la connessione ai vari sistemi di calcolo (tra cui CRESCO6) presenti nell'infrastruttura, come illustrato in Fig. 5.

```

(palombi) cresco-in.portici.enea.it — Konsole
File Edit View Bookmarks Settings Help
filippo@filippo-Inspiran-5582:~$ ssh -X palombi@cresco-in.portici.enea.it
palombi@cresco-in.portici.enea.it's password:
Last login: Mon Jun  6 15:54:09 2022 from 192.168.215.138
***** Front-end Menu
Host cresco-in2.portici.enea.it on glo  9 giu 2022, 09:32:19, CEST
Local tty /dev/pts/4 connected from remote host 192.168.219.198

Welcome to user palombi on ENEAGRID CRESCO Clusters

For general information see:
http://www.eneagrid.enea.it
http://www.cresco.enea.it

This host is a ssh gateway to CRESCO front-ends for users connecting from NON ENEA networks.
This host MUST NOT be used neither as Front-end nor for Data Transfer by sftp/scp.
When connecting from ENEA networks, please access directly the Front-end by ssh.
For scp/sftp from NON ENEA networks please use cresco-scp.portici.enea.it.

You are connected to cresco-in2.portici.enea.it.
Please select a Front-end in the following list
*** Site-----Host-----
1) Portici: cresco4x001          (also for interactive jobs)
2) Portici: cresco4x002          (also for interactive jobs)
3) Portici: cresco4x001          (also for interactive jobs)
4) Portici: cresco4x002          (also for interactive jobs)
5) Portici: cresco4x001          (node with 768 GB RAM, also for interactive jobs)
6) Casaccia: cresco4x001
7) Frascati: cresco4f001
8) Brindisi: crescob-nv11       (only for AI projects)
*** Select the frontend or (@)exit and logout >

```

Figura 6. Connessione al server cresco-in.portici.enea.it via SSH

2.5 Accesso all'infrastruttura tramite FARO

Un'altra possibilità per connettersi a CRESCO è offerta da FARO, un applicazione web sviluppata in ENEA, che funziona sul browser e si basa su ThinLinc (<https://www.cendio.com/>). L'applicazione FARO può essere lanciata collegandosi al sito web

<https://www.cresco.enea.it>

e seguendo il link

[cresco-in-gui.portici.enea.it](https://www.cresco.enea.it/cresco-in-gui.portici.enea.it)

presente al centro della pagina, come indicato in Figura 7. Una volta effettuata la connessione previa inserimento di username e password, l'utente è in grado di operare attraverso una console attiva all'interno del browser.



Figura 7. Connessione a CRESCO via FARO

2.6 Accesso all'infrastruttura tramite MobaXterm

Per ragioni di completezza, discutiamo brevemente l'accesso all'infrastruttura attraverso il client MobaXterm per Windows. MobaXterm è un client molto versatile per fare remote computing. In una singola applicazione esso fornisce numerose funzioni pensate per programmatori, webmasters, amministratori IT, etc. MobaXterm implementa tutti i principali protocolli di connessione (SSH, RDP, VNC, FTP, MOSH, ...) e comandi Unix (bash, ls, sed, grep, awk, rsync, ...). Il client è inoltre provvisto di un server X *embedded*, il quale consente operazioni grafiche senza bisogno di trasferire file da remoto. L'uso di applicazioni grafiche remote risulta particolarmente veloce. Ciò rende questo client uno strumento eccellente per gli utenti CRESCO che usano Windows. La creazione di una nuova connessione richiede l'inserimento dei dati di connessione (RemoteHost, Username, Port) all'interno di un pop-up lanciato dal pulsante Session e illustrato in Figura 8. La connessione apre una console identica a quella di un SO Linux.

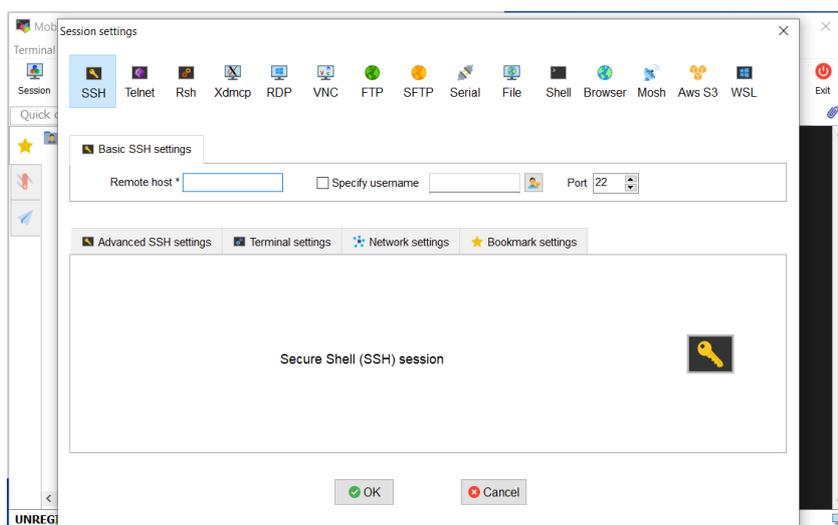


Figura 8. Creazione di un profilo di connessione SSH su MobaXterm

3 Creazione di un'area di storage per il progetto IEMAP

Per raccogliere i dati prodotti nell'ambito del progetto IEMAP, siano essi risultati sperimentali o frutto di simulazioni numeriche, si è deciso di approntare un'area di storage visibile ed utilizzabile da tutti gli stakeholder del progetto. Le specifiche sulla base delle quali è stato progettato tale spazio dati sono le seguenti

- lo storage viene gestito dallo staff ICT dell'ENEA;
- i membri di ciascun gruppo di ricerca afferente al progetto hanno la possibilità di raggiungere lo storage da remoto, cioè dal proprio PC in ufficio o a casa, senza limitazioni;
- a ciascun gruppo di ricerca afferente al progetto è assegnata una sotto-area dello storage;

- ciascuna sotto-area è accessibile in modalità WRITE da tutti i membri del gruppo di ricerca assegnatario della sotto-area;
- ciascuna sotto-area è accessibile in modalità READ da tutti i membri degli altri gruppi di ricerca afferenti al progetto IEMAP in modo da favorire la condivisione dei dati;
- l'intera area di storage risulta inaccessibile a tutti gli utenti ENEAGRID non afferenti al progetto IEMAP.

In considerazione di tali specifiche, si è scelto di ospitare lo storage sul disco GPFs installato al C.R. Enea di Portici. Lo storage appare agli utenti abilitati al suo utilizzo come una directory standard di un SO Linux.

Più precisamente, esso è raggiungibile dal nodo di frontend di CRESCO6, denominato

cresco6x001.portici.enea.it

al path

[/gporq3/store_3/project/misinn/](#)

La sua struttura è illustrata in Figura 9. Come da specifiche, ciascun gruppo di ricerca ha accesso in scrittura alla propria sottodirectory, mentre può accedere solo in lettura alle sottodirectory degli altri gruppi di ricerca.

Per poter ottenere questo risultato si è considerato e sfruttato il fatto che i filesystem GPFs implementano un sistema di accessi basato su UNIX e si è proceduto come descritto nel seguito.

Per prima cosa, si è considerato preliminarmente che nell'infrastruttura ENEAGRID, tutte le utenze appartengono di default al gruppo UNIX *enea* al momento della loro creazione. Per poter permettere agli stakeholder del progetto di usare lo spazio storage in oggetto in modo esclusivo, è stato definito un gruppo UNIX denominato *misinn*, cui sono stati assegnati, quale gruppo UNIX aggiuntivo, tutti e soli gli stakeholder del progetto. E' stata quindi creata la directory madre dello spazio storage *misinn* e le è stato assegnato il gruppo omonimo *misinn* quale gruppo proprietario. Quindi è stato applicato il permesso **setgid** alla directory madre.

```
File Edit View Bookmarks Settings Help
<palombi@cresco6x001 /gporq3/store_3/project/misinn> pwd
/gporq3/store_3/project/misinn
<palombi@cresco6x001 /gporq3/store_3/project/misinn> ls -lah
total 21K
drwxrws-- 21 root misinn 4.0K Jun 9 12:07 .
drwxr-sr-x 11 root root 4.0K Jan 26 16:44 ..
drwxrwsr-x 2 root micnricom 4.0K Dec 9 11:38 cnr-icom
drwxrwsr-x 2 root micnricmate 4.0K Dec 20 16:25 cnr-icmate
drwxrwsr-x 2 root micnrimem 4.0K Dec 9 11:38 cnr-imem
drwxrwsr-x 2 root micnrimm 4.0K Dec 9 11:39 cnr-imm
drwxrwsr-x 2 root micnripcf 4.0K Dec 9 11:38 cnr-ipc
drwxrwsr-x 2 root micnrismb 4.0K Dec 9 11:38 cnr-ismn-bologna
drwxrwsr-x 2 root micnrismp 4.0K Dec 9 11:38 cnr-ismn-palermo
drwxrwsr-x 2 root micnrismr 4.0K Dec 9 11:10 cnr-ism-roma
drwxrwsr-x 2 root micnrismt 4.0K Dec 9 11:10 cnr-ism-trieste
drwxrwsr-x 2 root micnrisof 4.0K Dec 9 11:38 cnr-isof
drwxrwsr-x 2 root micnristec 4.0K Dec 9 11:37 cnr-istec
drwxrwsr-x 2 root micnritae 4.0K Dec 9 11:39 cnr-itae
drwxrwsr-x 2 root micnrnanotec 4.0K Dec 9 11:39 cnr-nanotec
drwxrwsr-x 2 root micnrspina 4.0K Dec 9 11:39 cnr-spin-aquila
drwxrwsr-x 2 root micnrspinn 4.0K Dec 9 11:39 cnr-spin-napoli
drwxrwsr-x 2 root micnrspinr 4.0K Dec 9 11:39 cnr-spin-roma
drwxrwsr-x 2 root mienea 4.0K Jun 9 13:39 enea
drwxrwsr-x 2 root miitt 4.0K Dec 9 10:37 iit
drwxrwsr-x 2 root mirse 4.0K Dec 9 10:37 rse
<palombi@cresco6x001 /gporq3/store_3/project/misinn> █
```

Figura 9. Struttura dello spazio storage di IEMAP sul filesystem GPFs del C.R. ENEA di Portici

In tal modo qualunque stakeholder del progetto è in grado di accedere alla directory madre dello storage, mentre tutti gli altri utenti afferenti al gruppo *enea* non sono in grado di accedervi. Al fine di differenziare gli stakeholder secondo l'appartenenza a diversi gruppi di ricerca, sono stati successivamente creati gruppi UNIX distinti per ciascun gruppo di ricerca, usando una nomenclatura del tipo '*mi + acronimo-specifico-del-gruppo-di-ricerca*', dove l'acronimo *mi* sta per Mission Innovation e serve a identificare immediatamente i gruppi UNIX come afferenti al progetto IEMAP nell'ambito di un insieme molto vasto di gruppi UNIX esistente all'interno dell'infrastruttura ENEAGRID, mentre l'acronimo specifico del gruppo di ricerca identifica il gruppo di ricerca stesso. Per es., il gruppo UNIX *micnrismp* è il gruppo di appartenenza di tutti gli stakeholders afferenti all'unità ISMN di Palermo del CNR che partecipano al progetto IEMAP. E' stata quindi creata una sotto-directory nella directory madre dello storage, per ciascun gruppo di ricerca e le è stato assegnato quale gruppo UNIX proprietario il gruppo UNIX associato al gruppo di ricerca corrispondente. Le sotto-directory create sono assegnate ai gruppi di ricerca come nella tabella seguente.

Gruppo di Ricerca	Nome Gruppo UNIX sul filesystem GPFs	ID Gruppo UNIX sul filesystem GPFs	Directory nello storage /gporq3/store_3/project /misinn
CNR Institute of Condensed Matter	micnricmate	10241	cnr-icmate
CNR Institute of Structure of Matter - Rome	micnrismr	10246	cnr-ism-roma
CNR Institute of Structure of Matter Trieste	micnrismt	10247	cnr-ism-trieste
CNR Institute of Science and Technology for Ceramics	micnristec	10248	cnr-istec
CNR Istituto Materiali per l'Elettronica e il Magnetismo	micnrimem	10249	cnr-imem

Gruppo di Ricerca	Nome Gruppo UNIX sul filesystem GPFs	ID Gruppo UNIX sul filesystem GPFs	Directory nello storage /gporq3/store_3/project /misinn
CNR Institute for the Chemistry of Organometallic Compounds	micnriccom	10250	cnr-iccom
CNR Istituto per lo Studio dei Materiali Nanostrutturati Bologna	micnrismnb	10251	cnr-ismn-bologna
CNR Istituto per lo Studio dei Materiali Nanostrutturati Palermo	micnrismnp	10252	cnr-ismn-palermo
CNR Istituto per i processi Chimico Fisici	micnripcf	10253	cnr-ipcf
CNR Institute for Organic Synthesis and Photoreactivity	micnrisof	10254	cnr-isof
CNR Istituto di Tecnologie Avanzate per l'Energia	micnritae	10255	cnr-itae
CNR Istituto per la Microelettronica e Microsistemi	micnrimm	10256	cnr-imm
CNR Institut of Nanotechnology	micnrrnanotec	10257	cnr-nanotec
CNR SuPerconducting and other INnovative materials and devices Institute – Roma	micnrspinr	10258	cnr-spin-roma
CNR SuPerconducting and other INnovative materials and devices Institute – L'Aquila	micnrspina	10259	cnr-spin-aquila
CNR SuPerconducting and other INnovative materials and devices Institute – Napoli	micnrspinn	10260	cnr-spin-napoli
RSE – Ricerca sul Sistema Energetico	mirse	10243	rse
Istituto Italiano di Tecnologia	miiit	10244	iit
ENEA – Agenzia Nazionale per le nuove tecnologie, l'energia e lo sviluppo economico sostenibile	mienea	10245	enea

Infine, a ciascuna sotto-directory è stato applicato il permesso **setgid** in modo che il contenuto della sotto-directory rimanga interamente in gestione agli stakeholder di riferimento.

3.1 Trasferimento dati da/verso l'area di storage

Tutti i trasferimenti dall'area di storage verso macchine esterne o da macchine esterne verso l'area di storage avvengono, seguendo regole predefinite e vigenti nell'infrastruttura ENEAGRID, attraverso una macchina su cui è attivo un server scp. Questa macchina è formalmente denominata

cresco-scp.portici.enea.it.

Tale identificativo rappresenta, in realtà, un alias a due server fisici, entrambi installati presso il C.R. Enea di Portici, che effettuano trasferimenti scp in modo da avere un carico dinamicamente bilanciato in entrata/uscita. I due server servono l'intera comunità ENEAGRID, cioè coprono i trasferimenti del progetto IEMAP e di altri progetti in corso.

Per utenti che si avvalgono dei SO Linux e macOS, i trasferimenti verso lo storage dati vengono effettuati attraverso il comando

```
> scp -r DirectoryLocale username@cresco-scp.portici.enea.it:/gporq3/store_3/project/misinn/SottoDirectoryGruppo
```

mentre i trasferimenti dallo storage dati vengono effettuati attraverso il comando

```
> scp -r username@cresco-scp.portici.enea.it:/gporq3/store_3/project/misinn/SottoDirectoryGruppo DirectoryLocale
```

dove l'opzione *-r* (*recursive*) deve essere utilizzata solo nel caso di trasferimento di intere directory e relative sottodirectory.

Per gli utenti che si avvalgono del SO MS-Windows10, è possibile semplificare il trasferimento di files da/verso l'area di storage in modo da evitare l'uso esplicito di un client ssh. Si è proposta a tal fine una soluzione che utilizza l'utility SSHFS-WIN, la quale prevede le seguenti operazione che, ciascun utente deve effettuare in locale sul proprio PC:

1. Al sito <http://www.secfs.net/winfsp/rel/> scaricare e installare i pacchetti

WinFsp e SSHFS-Win (x64)

2. Su Windows10 aprire il prompt dei comandi e digitare il comando

```
net use Z: \\sshfs.r\username@cresco-scp.portici.enea.it\gporq3\store_3\project\misinn\
```

È importante notare che

- nel comando precedente, tra Z: e \\ c'è uno spazio;
- lo username da usare è quello dell'account ENEAGRID;
- nella specifica del path globale, si deve usare il backslash '\' come simbolo separatore di directory;
- l'unità sulla quale viene montato lo spazio storage NON deve risultare già in uso.

4 Installazione del package di workflow AiiDA

I flussi di lavoro (o workflow) sono una delle componenti più importanti per effettuare calcoli complessi con sistemi HPC. Essi consentono all'utente di eseguire catene di calcoli ben definite su qualsiasi numero di strutture di input, sia generate che acquisite da una fonte esterna. Uno dei pacchetti software open-source maggiormente utilizzati per gestire flussi di lavoro è AiiDA [3], liberamente scaricabile dal sito

<https://www.aiida.net/>

Invece di offrire un numero limitato di schemi di automazione, predisposti per alcune funzioni specifiche (equazione di stato, fononi, ecc...) in AiiDA è presente un motore di workflow completo, dove l'utente può in linea di principio scrivere ogni possibile interazione con tutti le componenti di AiiDA, dal motore di invio alle connessioni ai database dei materiali. In AiiDA un flusso di lavoro è uno script python eseguito da un demone, contenente diverse funzioni definite dall'utente chiamate passi. In ogni passo sono disponibili tutte le funzioni AiiDA e i calcoli, avviati e recuperati, così come altri sotto-flussi di lavoro.

Si è deciso di inserire l'installazione e il testing di AiiDA nella parte infrastrutturale di IEMAP in modo da avere un primo esplicito esempio di esperimento di automazione di simulazioni numeriche per la ricerca di nuovi materiali in CRESCO. In questa sezione discutiamo gli aspetti tecnici del processo di installazione.

Il pacchetto AiiDA è stato installato in ambiente CONDA sulla partizione AFS dell'ENEA nella directory

```
/afs/enea.it/project/ai4mat/software/aiida
```

L'uso di AiiDA prevede la creazione preliminare di un profilo utente attraverso **verdi**, un eseguibile facente parte del package AiiDA e operante da linea di comando, che permette l'interazione con le componenti del workflow. Tale profilo è associato ad un utente, responsabile della gestione del profilo stesso. Si è deciso di definire un unico profilo, denominato **ai4mat** (un acronimo che sta per *artificial intelligence for materials*) utilizzato da tutti i membri gruppo ENEA di IEMAP e un nuovo utente ENEAGRID, denominato **aiidausr** (con indirizzo di posta elettronica aiida.user@enea.it), responsabile di detto profilo. Per poter funzionare correttamente, il profilo aiida deve inoltre poter accedere ad un database PostgreSQL e al message broker RabbitMQ. Si è deciso di installare un server del database e del message broker su una macchina virtuale creata appositamente e denominata

```
aiida.portici.enea.it.
```

Su tale macchina è stato installato il SO Linux *CentOS 7.8.2003*. Mentre il server PostgreSQL è fisicamente installato su *aiida.portici.enea.it*, il repository del database, cioè lo spazio dati su cui il medesimo server insiste, si trova nella directory GPFs dell'utente *aiidausr*, cioè nella directory

```
/afs/enea.it/por/user/aiidausr/PFS/por/.aiida/repository/ai4mat
```

dove il path */afs/enea.it/por/user/aiidausr/PFS/por/* è un link che punta alla directory fisica

```
/gporq3/store_0/usr/aiidausr/
```

del disco GPFs di CRESCO6.

La creazione del profilo *ai4mat* è stata ottenuta attraverso il comando

```
> verdi setup --profile ai4mat
```

La procedura di setup prevede che l'utente definisca una serie di specifiche associate al profilo. Riportiamo in Figura 10 l'IO del processo di creazione del profilo.

```

(/afs/enea.it/project/ai4mat/software/aiida) <aiidausr@cresco6x001 ~> verdi setup --profile ai4mat
Info: enter "?" for help
Info: enter "!" to ignore the default and set no value
Email Address (for sharing data): aiida.user@enea.it
First name: Aiida
Last name: User
Institution: ENEA
Database engine (postgresql_psycpg2) [postgresql_psycpg2]:
Database backend (django, sqlalchemy) [django]:
Database host: aiida.portici.enea.it
Database port [5432]:
Database name: ai4mat
Database username: aiidausr
Database password:
Broker protocol (amqp, amqps) [amqp]:
Broker username [guest]:
Broker password [guest]:
Broker host [127.0.0.1]: aiida.portici.enea.it
Broker port [5672]:
Broker virtual host name []
Repository directory [/afs/enea.it/por/user/aiidausr/.aiida/repository/ai4mat]: /afs/enea.it/por/user/aiidausr/PFS/por/.aiida/repository/ai4mat
Success: created new profile 'ai4mat'.
Info: migrating the database.
Operations to perform:
  Apply all migrations: auth, contenttypes, db
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0001_initial... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying db.0001_initial... OK
  Applying db.0002_db_state_change... OK
  Applying db.0003_add_link_type... OK
  Applying db.0004_add_daemon_and_uid_indices... OK
  Applying db.0005_add_cmtime_indices... OK
  Applying db.0006_delete_dbpath... OK
  Applying db.0007_update_linktypes... OK
  Applying db.0008_code_hidden_to_extra... OK
  Applying db.0009_base_data_plugin_type_string... OK
  Applying db.0010_process_type... OK
  Applying db.0011_delete_kombu_tables... OK
  Applying db.0012_drop_dblock... OK
  Applying db.0013_django_1_8... OK
  Applying db.0014_add_node_uid_unique_constraint... OK
  Applying db.0015_invalidating_node_hash... OK
  Applying db.0016_code_sub_class_of_data... OK
  Applying db.0019_migrate_builtin_calculations... OK
  Applying db.0020_provenance_redesign... OK
  Applying db.0021_dbgroup_name_to_label_type_to_type_string... OK
  Applying db.0022_dbgroup_type_string_change_content... OK
  Applying db.0023_calc_job_option_attribute_keys... OK
  Applying db.0024_dblog_update... OK
  Applying db.0025_move_data_within_node_module... OK
  Applying db.0026_trajectory_symbols_to_attribute... OK
  Applying db.0027_delete_trajectory_symbols_array... OK
  Applying db.0028_remove_node_prefix... OK
  Applying db.0029_rename_parameter_data_to_dict... OK
  Applying db.0030_dbnode_type_to_dbnode_type... OK
  Applying db.0031_remove_dbcomputer_enabled... OK
  Applying db.0032_remove_legacy_workflows... OK
  Applying db.0033_replace_text_field_with_json_field... OK
  Applying db.0034_drop_node_columns_nodeversion_public... OK
  Applying db.0035_simplify_user_model... OK
  Applying db.0036_drop_computer_transport_params... OK
  Applying db.0037_attributes_extras_settings_json... OK
  Applying db.0038_data_migration_legacy_job_calculations... OK
  Applying db.0039_reset_hash... OK
  Applying db.0040_data_migration_legacy_process_attributes... OK
  Applying db.0041_seal_unsealed_processes... OK
  Applying db.0042_prepare_schema_reset... OK
  Applying db.0043_default_link_label... OK
  Applying db.0044_dbgroup_type_string... OK
  Applying db.0045_dbgroup_extras... OK
Success: database migration completed.

```

Figura 10. Creazione del profilo ai4mat in AiiDA

Lo stato del profilo può essere verificato attraverso il comando

> *verdi status*

```

(/afs/enea.it/project/ai4mat/software/aiida) <aiidausr@cresco6x001 ~> verdi status
✓ config dir: /gporq3/store_0/usr/aiidausr/.aiida
✓ profile: On profile ai4mat
✓ repository: /gporq3/store_0/usr/aiidausr/.aiida/repository/ai4mat
✓ postgres: Connected as aiidausr@aiida.portici.enea.it:5432
✓ rabbitmq: Connected as amqp://aiidausr:plbdmn.21@aiida.portici.enea.it:5672?heartbeat=600
daemon: Daemon is running as PID 1537309 since 2022-06-11 15:45:41
(/afs/enea.it/project/ai4mat/software/aiida) <aiidausr@cresco6x001 ~>

```

Figura 11. Output del comando 'verdi status'

il cui output è riportato in Figura 11.

4.1 Installazione del server PostgreSQL

AiiDA necessita di un database di back-end per memorizzare i nodi, gli attributi dei nodi e altre informazioni, in modo che l'utente possa eseguire query molto veloci sui risultati. Prima di installare AiiDA, è necessario scegliere (ed eventualmente configurare) un backend adatto. Gli sviluppatori di AiiDA suggeriscono di utilizzare PostgreSQL, in quanto considerato il database più versatile e dotato di caratteristiche configurabili.

È semplice installare PostgreSQL dai repository CentOS 7. Si inizia con il seguente comando:

```
> sudo yum install postgresql-server postgresql-contrib
```

Al termine dell'installazione, è possibile inizializzare il database utilizzando il comando

```
> sudo postgresql-setup initdb
```

Infine, è possibile configurare PostgreSQL in modo che si avvii automaticamente a ogni riavvio del sistema utilizzando il comando

```
> sudo systemctl enable postgresql
```

Una volta installato postgresQL sulla macchina aiida.portici.enea.it, abbiamo proceduto a creare il database ai4mat. La creazione del database prevede la seguente procedura.

1. Essendo connessi alla macchina, passare all'utenza postgres tramite il comando

```
> su - postgres
```

2. Collegarsi quindi al database attraverso il comando

```
> psql
```

3. Una volta collegati, digitare il comando di creazione del database

```
postgres=# CREATE DATABASE ai4mat;
```

4. Infine, per controllare che il database sia realmente controllato, digitare il comando

```
postgres=# \l;
```

L'esecuzione di questo comando su aiida.portici.enea.it produce l'output mostrato in Figura 12.

```
postgres=# \l
```

List of databases					
Name	Owner	Encoding	Collate	Ctype	Access privileges
ai4mat	postgres	UTF8	en_US.UTF-8	en_US.UTF-8	=Tc/postgres + postgres=CTc/postgres+ aiaausr=CTc/postgres
postgres	postgres	UTF8	en_US.UTF-8	en_US.UTF-8	
template0	postgres	UTF8	en_US.UTF-8	en_US.UTF-8	=c/postgres + postgres=CTc/postgres
template1	postgres	UTF8	en_US.UTF-8	en_US.UTF-8	=c/postgres + postgres=CTc/postgres

```
(4 rows)
postgres=# █
```

Figura 12. Elenco database PostgreSQL su aiida.portici.enea.it

4.2 Installazione del message broker RabbitMQ

RabbitMQ è un middleware responsabile del dispatch di messaggi tra le diversi componenti di AiiDA. E' possibile effettuare il download del package dalla pagina web

<https://www.rabbitmq.com/download.html>

Si è effettuata una installazione di RabbitMQ sulla macchina virtuale aiida.portici.enea.it. In particolare, al suddetto sito si è scelto di utilizzare la compilazione denominata 'Generic binary build'. Seguendo il link, si è effettuato il download del pacchetto `rabbitmq-server-generic-unix-3.8.16.tar.xz`. Si è quindi proceduto all'installazione del pacchetto seguendo le istruzioni fornite sul sito medesimo. Nella fase preliminare dell'installazione, si è osservato tuttavia che RabbitMQ richiede l'installazione del pacchetto `erlang` come dipendenza. L'installazione di `erlang` sul SO Linux Centos7 non è triviale, poiché non si trova nel repository della distribuzione. Di seguito riportiamo una descrizione dettagliata dell'installazione di `erlang`.

Ci sono due passi che devono essere considerati di primaria importanza per procedere: mapping al repository *Extra Packages for Enterprise Linux* (EPEL) e aggiornamento degli strumenti già installati sulla macchina virtuale. Per prima cosa abbiamo mappato l'EPEL. Questo può essere fatto con il semplice comando

```
> sudo yum install epel-release
```

Quindi ci siamo assicurati che i pacchetti installati con il nostro SO fossero aggiornati. Il comando seguente aggiorna tutti i pacchetti sul sistema e tenta di rimuovere i pacchetti obsoleti:

```
> sudo yum -y update && sudo yum -y upgrade
```

Quindi abbiamo installato tutti gli strumenti di sviluppo necessari per installare `erlang`, attraverso il comando

```
> sudo yum install gcc gcc-c++ glibc-devel make ncurses-devel openssl-devel autoconf  
java-1.8.0-openjdk-devel git
```

Si deve quindi installare `wxBase`, un pacchetto che contiene le librerie necessarie a `erlang`. A tal fine, abbiamo usato il comando

```
> sudo yum install wxBase.x86_64
```

Un altro pacchetto necessario al corretto funzionamento di *erlang* è *wget*. Abbiamo installato quest'ultimo pacchetto attraverso il comando

```
> sudo yum -y install wget
```

Una volta completata l'installazione dei pacchetti preliminari, abbiamo proceduto con l'installazione di *erlang*. Al momento della stesura di questo deliverable, la versione stabile più recente di *erlang* è la 18.0. Il pacchetto può essere scaricato dal sito web

erlang-solutions.com

Attraverso i prossimi due comandi, abbiamo recuperato un pacchetto che aggiunge un nuovo repository al nostro elenco di repository e lo installa.

```
> wget http://packages.erlang-solutions.com/erlang-solutions-1.0-1.noarch.rpm
```

```
> sudo rpm -Uvh erlang-solutions-1.0-1.noarch.rpm
```

Ora che il nostro sistema sa dove ottenere l'ultima versione stabile di *erlang*, possiamo effettivamente installarla. Il pacchetto *esl-erlang* è il pacchetto di installazione completo di *erlang/OTP*, che abbiamo installato tramite il comando

```
> sudo yum install esl-erlang
```

A questo punto *erlang* è installato e abbiamo potuto effettuare l'installazione di RabbitMQ in maniera agevole seguendo le istruzioni di installazione presenti sul sito di riferimento.

Per permettere ad AiiDA di utilizzare RabbitMQ, abbiamo dovuto aprire sulla macchina virtuale *aiida.portici.enea.it* la porta di rete 5672 e abbiamo dovuto lanciare il server di RabbitMQ. L'eseguibile che avvia il server sulla macchina virtuale è il file

```
/usr/local/rabbitmq_server-3.8.16/sbin/rabbitmq-server
```

Il comando per avviare il server in background è

```
> sudo rabbitmq-server -detached
```

Si noterà che nell'output del comando *verdi status* riportato in Figura XYZ è indicata chiaramente la connessione al server di RabbitMQ sulla macchina virtuale *aiida.portici.enea.it* alla porta 5672.

4.3 Configurazione dell'infrastruttura CRESCO in AiiDA

AiiDA è un'infrastruttura Python open-source che permette l'esecuzione in modo automatico di calcoli o codici complessi da parte dei ricercatori.

Tramite interfacce, AiiDA può supportare qualsiasi codice computazionale (plugin) e gestire risorse di calcolo remote ed eterogenee che funzionano con i principali scheduler (SLURM, PBS Pro, torque, SGE or LSF).

Affinché AiiDA possa sottomettere i calcoli o i codici sull'infrastruttura HPC di ENEA, è necessario interfacciare AiiDA con il cluster HPC tramite la procedura di configurazione.

La procedura può essere richiamata tramite il comando

> *verdi computer setup*

e richiede di impostare le informazioni riportate in Tabella 1.

Tabella 1. Parametri di configurazione di CRESCO in AiiDA

Nome parametro	Valore assegnato	Descrizione
Computer label	cresco6s	L'etichetta (di facile utilizzo) della nuova istanza del computer che sta per essere creata nel DB (l'etichetta viene utilizzata ad esempio quando devi scegliere un computer per avviare un calcolo su di esso). Le etichette devono essere univoche
Fully-qualified hostname	cresco6x001.portici.enea.it	Il nome host completo del computer a cui ci si vuole connettere
Description	Configurazione di produzione su cresco6x001.portici.enea.it che usa l'area di scratch come work directory	Una descrizione della risorsa computazionale. Questo è utile se si hanno molte risorse e si vuole aggiungere del testo per distinguerle
Enabled	True	Se False, il computer viene disabilitato e i calcoli ad esso associati non verranno inviati. Questo permette di disabilitare temporaneamente un computer se sta dando problemi o è fermo per manutenzione, senza la necessità di cancellarlo dal DB
Transport plugin	local	Il tipo di plugin di trasporto da utilizzare
Scheduler plugin	lsf	Il nome del plugin che deve essere utilizzato per gestire il job scheduler sulla risorsa computazionale
shebang line	#!/bin/bash	Questa è la prima riga riportata all'inizio dello script di invio. L'impostazione predefinita è '#!/bin/bash'
Work directory on the computer	/gporq3/store_0/usr/aiaidaur/	Il percorso assoluto della directory sul computer remoto in cui AiiDA eseguirà i calcoli (spesso è lo scratch del computer)
Mpirun command	mpirun -bootstrap rsh	Il comando mpirun necessario sul cluster per eseguire programmi paralleli MPI
Default number of CPUs per machine	48	Il numero di processi MPI per macchina che devono essere eseguiti se non diversamente specificato

5 Conclusioni

In questo documento abbiamo riassunto l'attività infrastrutturale svolta nel primo anno di finanziamento del progetto IEMAP. Il lavoro eseguito si è articolato lungo due linee principali: da un lato permettere l'accesso all'infrastruttura a tutti gli stakeholder del progetto, dall'altro realizzare una serie di servizi volti a garantire ad essi la possibilità di perseguire le finalità scientifiche del progetto stesso. In particolare, partendo dall'infrastruttura CRESCO esistente in ENEA, abbiamo creato account per gli utenti appartenenti ai vari gruppi di ricerca coinvolti e li abbiamo istruiti sulle caratteristiche, le modalità d'accesso e le potenzialità dell'infrastruttura. Quindi abbiamo costruito un'area di storage riservata ad IEMAP per consentire la condivisione dei dati raccolti dai ricercatori, siano essi di natura computazionale ovvero sperimentale ed abbiamo fornito assistenza durante la fase di testing da parte dei ricercatori. Parallelamente, abbiamo installato e configurato un ambiente per la gestione automatizzata dei flussi di lavoro basata sul software AiiDA. Ciò ha consentito l'avvio dell'attività investigativa riguardante la progettazione di materiali basata sull'uso di algoritmi di intelligenza artificiale. Nel presente documento sono riportate tutte le informazioni necessarie ad utenti ed amministratori per poter utilizzare CRESCO e gestire i servizi realizzati nel contesto di IEMAP ed eventualmente permetterne una riproduzione efficiente e rapida.

6 Riferimenti bibliografici

- [1] G. Ponti *et al.*, "The role of medium size facilities in the HPC ecosystem: the case of the new CRESCO4 cluster integrated in the ENEAGRID infrastructure," *2014 International Conference on High Performance Computing & Simulation (HPCS)*, 2014, pp. 1030-1033, doi: 10.1109/HPCSim.2014.6903807.
- [2] F. Iannone *et al.*, "CRESCO ENEA HPC clusters: a working example of a multifabric GPFS Spectrum Scale layout," *2019 International Conference on High Performance Computing & Simulation (HPCS)*, 2019, pp. 1051-1052, doi: 10.1109/HPCS48598.2019.9188135.
- [3] Huber, S.P., Zoupanos, S., Uhrin, M. *et al.* AiiDA 1.0, a scalable computational infrastructure for automated reproducible workflows and data provenance. *Sci Data* **7**, 300 (2020).
<https://doi.org/10.1038/s41597-020-00638-4>

7 Abbreviazioni ed acronimi

DB= DataBase

FAIR (dati) = dati che rispettano i principi di rintracciabilità (findability), accessibilità (accessibility), interoperabilità (interoperability) e riusabilità (reusability)

API= Application Programming Interface

FRD= Fair Data Point

JSON= JavaScript Object Notation

CLI= Command Line Interface

GPFS= General Parallel File System

POSIX= Portable Operating System Interface for Unix

SQL= Structured Query Language

PFlops= Peta Floating operations per second

SO= Sistema Operativo