

EGRID Project

EGRID TUTORIAL

HANDOUTS PER I PARTECIPANTI ALLA RELEASE 1.0 DI EGRID

Document identifier:

Date: October 22, 2004

Partner(s):

Lead Partner: **The Abdus Salam ICTP, Trieste**

Document status: **Version 0.9.9**

Author(s): The EGRID team

File: **egrid-tutorial**

Abstract: Queste note sono uno strumento di aiuto per le persone che dovranno imparare ad usare il middleware di griglia fornito dal progetto EDG ed usato/modificato all'interno del progetto EGRID.

È rivolto a persone che abbiano una conoscenza base del sistema operativo Linux/Unix e conoscano comandi elementari di shell e l'uso di un editor di testo.

CONTENTS

1. INTRODUZIONE	4
1.1. INTRODUZIONE AI CONCETTI DI GRIGLIA	4
1.1.1. SPAZIO DISCO E CALCOLO	4
1.1.2. SISTEMA INFORMATIVO INTERNO ALLA GRIGLIA	6
1.1.3. MECCANISMO DI FUNZIONAMENTO	6
1.1.4. INTERAZIONE CON L'UTENTE	6
1.1.5. AUTENTICAZIONE E SICUREZZA	6
1.2. INIZIARE A LAVORARE IN GRIGLIA	7
1.2.1. OPERAZIONI PRELIMINARI	7
1.2.2. INSTALLAZIONE DEL CERTIFICATO NELLA UI	7
1.2.3. RICHIESTA DI ABILITAZIONE AL LAVORO IN GRIGLIA EGRID	7
1.2.4. ATTIVAZIONE E GESTIONE DI UN PROXY DEL CERTIFICATO	7
2. LAVORARE CON I DATI IN EGRID	9
2.1. ORGANIZZAZIONE DEI DATI IN EGRID	9
2.2. LAVORARE CON I COMANDI FORNITI DAL REPLICA MANAGER DI EDG	11
2.2.1. CONOSCERE LA LISTA DI SE DELLA PROPRIA VO E IL LORO ATTRIBUTI	11
2.2.2. CARICARE UN FILE DI DATI LOCALE IN UNO SE E REGISTRARLO NEL CATALOGO	13
2.2.3. CREAZIONE DI UNA REPLICA IN UN ALTRO SE.	14
2.2.4. COPIARE UN FILE PRESENTE IN UNO SE, IN UNO DIVERSO ASSEGNANDOLI UN NUOVO NOME LOGICO	15
2.2.5. ELENCO DI TUTTE LE COPIE FISICHE DI UN MEDESIMO FILE LOGICO	15
2.2.6. COPIA DI UN FILE DALLA GRIGLIA NELLA UI LOCALE	16
2.2.7. RIMOZIONE DI UN FILE DA UNO SE	16
2.2.8. ELENCO DEL CONTENUTO DI UNO SE	16
2.3. COMANDI DI ALTO LIVELLO EGRID	17
2.3.1. CARICARE UN FILE DATI LOCALE NEL NODO DI CACHE	17
2.3.2. REPLICARE UN FILE DALLA CACHE AL NODO PRINCIPALE	17
2.3.3. REPLICARE UN FILE DAL NODO PRINCIPALE ALLA CACHE	17
2.3.4. COPIARE UN FILE DAL NODO PRINCIPALE A QUELLO DI CACHE, CON UN NUOVO NOME LOGICO	18
2.3.5. COPIARE UN FILE DAL NODO DI CACHE A QUELLO PRINCIPALE, CON UN NUOVO NOME LOGICO	18
2.3.6. SCARICARE UN FILE IN LOCALE	18
2.3.7. CAPIRE DOVE SI TROVA UN FILE	18
2.3.8. VISUALIZZARE FILE IN CACHE	19
2.3.9. CANCELLARE FILE DALLA CACHE	19
2.3.10. ELENCARE I FILE IN GRIGLIA	19
2.3.11. ELIMINARE FILE IN GRIGLIA	19
2.4. COME LAVORARE CON LA DIRECTORY PROGETTI SULLO SE CENTRALE	20
2.4.1. EGRID GROUP MANAGEMENT	20
2.5. CREAZIONE DELLA DIRECTORY RELATIVA AL GRUPPO	23
2.6. PERMESSI SU FILES E DIRECTORY	25

3. ESEMPI DI JOB SUBMISSION IN EGRID	27
3.1. DESCRIZIONE DEL MECCANISMO DI SOTTOMISSIONE DEI JOB	27
3.2. SCENARI DI UTILIZZO DELLA JOB SUBMISSION IN EGRID	28
3.3. FASI DI SOTTOMISSIONE DI UN JOB IN GRIGLIA	28
3.4. SCRITTURA DEL FILE JDL	28
3.4.1. LANCIARE UN COMANDO PRESENTE SU UN WORKER NODE	28
3.4.2. ESECUZIONE DI UNO SCRIPT/PROGRAMMA PREPARATO IN LOCALE CHE CREA UN FILE DI OUTPUT 2	
3.4.3. ESECUZIONE DI UN JOB CHE ACCETTA DEI PARAMETRI IN STDINPUT E CREA UN NUMERO NON DEFINITO	
3.4.4. UTILIZZO DI SOFTWARE DI EGRID	30
3.4.5. GESTIONE DEI LFN IN GRIGLIA	31
3.5. SOTTOMISSIONE DI UN JOB IN GRIGLIA	33
3.6. STATI DI UN JOB IN GRIGLIA	34
3.7. RITIRO DEL RISULTATO DI UN JOB	35
3.8. CANCELLAZIONE DI UN JOB	35
3.9. ESEMPIO DI DEBUG	36
3.9.1. ESEMPIO DI DEBUG	37
3.10. COMANDI EGRID PER LA SOTTOMISSIONE DI JOBS	39
3.10.1. WRAPPER DI SOTTOMMISSIONE	39
3.10.2. OPERAZIONE EGRID PER LA TRASFORMAZIONE DEI DATI GREZZI DEI CD/DVD, IN FORMATO ASCII 3	
 A EGRID LIVE UI	 41
A1. REQUISITI PER IL FUNZIONAMENTO	41
A1.1. DATA E ORA	41
A1.2. NOME DNS	41
A1.3. FIREWALL	41
A1.4. HARDWARE	42
A2. USARE <i>EGRID LIVE UI</i>	42
A2.1. AVVIARE IL LIVE CD	42
A2.2. IL SISTEMA GRAFICO	42
A2.3. I COMANDI DI GRIGLIA	42
A3. INSTALLAZIONE	44
A3.1. PRIMA DI COMINCIARE	45
A3.2. PROCEDURA DI INSTALLAZIONE	45
A3.3. POST-INSTALLAZIONE	45
A4. DOWNLOAD	46
A5. LINK & ULTERIORI INFORMAZIONI	46
A6. CONTATTI	46

1. INTRODUZIONE

Questo documento è una guida alla prima versione dell'infrastruttura computazionale per la gestione dei dati del progetto EGRID, che si basa sul middleware di griglia sviluppato all'interno del progetto EDG (European Data Grid). Una serie di presentazioni complementari forniscono una panoramica completa dei concetti di Grid, del middleware EDG e delle soluzioni specifiche adottate all'interno del progetto EGRID.

In questa prima versione per le operazioni tipiche degli utenti dell'infrastruttura EGRID vengono forniti comandi di più alto livello rispetto a quelli generici del middleware EDG. Questo per facilitare l'utente e consentirgli un più agile svolgimento dei compiti. Vengono comunque discussi sia gli strumenti EGRID che quelli EDG perché sebbene il grosso del lavoro viene svolto dai comandi EGRID, rimane necessario l'utilizzo di alcuni comandi di EDG. Inoltre i comandi EGRID si appoggiano a quelli di EDG per cui la loro conoscenza risulta utile a capire i limiti e le capacità dei primi.

Si assume che chi legge abbia familiarità con l'ambiente Linux/Unix (comandi shell di base) e che abbia ottenuto un certificato di autenticazione.

1.1. INTRODUZIONE AI CONCETTI DI GRIGLIA

La tecnologia di griglia consente di avere macchine che erogano servizi di calcolo e di spazio disco, distribuite a livello geografico; e di accedere a tali servizi senza dover scegliere esplicitamente la macchina che li eroga, cioè utilizzandoli automaticamente e dinamicamente. La flessibilità introdotta dall'automatismo permette di attingere a più risorse di calcolo e di disco all'aumentare del fabbisogno.

La tecnologia di griglia quindi riguarda l'accesso alle risorse, rendendolo uniforme; e un meccanismo automatico per evitare di andare alla ricerca manualmente.

In particolare nel caso di EGRID si è scelto di adottare l'implementazione tecnica di EDG. In quest'ottica una griglia è costituita da un insieme di macchine detti nodi, ciascuno con funzionalità diverse: fig. 1 riporta i vari componenti e i legami tra di loro. Ci sono più nodi che forniscono capacità d'elaborazione e spazio disco (CE, WN e SE); un altro sistema di nodi è incaricato di monitorare i primi e mantenere aggiornato un catalogo che ne riassume lo stato del funzionamento (BDII,GIIS,GRIS); due nodi specifici hanno il compito di ricercare le macchine più adatte allo svolgimento di un programma sottoposto in griglia e alla localizzazione dei dati necessari (RB e RLS); un ultimo nodo (UI) ha il compito di ricevere richieste da parte dell'utente e sottoporle ai due nodi precedenti per l'esecuzione in griglia. L'accesso ai nodi, infine, viene controllato da un meccanismo di sicurezza che identifica l'utente e le operazioni che può svolgere. Una definizione di tutti questi termini sarà presentata nei sottoparagrafi seguenti.

1.1.1. SPAZIO DISCO E CALCOLO

SE Storage Element: nodo di griglia su cui vengono archiviati dati da elaborare, o che devono essere accessibili a più fruitori. L'SE maschera l'eterogeneità dei sistemi di archiviazione rendendo uniforme l'accesso alla risorsa spazio disco. Per esempio, un ente vuole mettere a disposizione della griglia un NAS da 10TB; un utente privato vuole mettere a disposizione 30GB di un disco IDE della propria macchina. In ambedue i casi il software di griglia installato in tali macchine presenterà lo spazio mascherando la tecnologia hardware specifica che sta dietro. Nel nostro esempio avremo uno SE da 30GB ed un altro da 10TB.

CE Computing Element e WN Worker Nodes: elementi di griglia incaricati rispettivamente di accettare programmi inviati e di eseguirli. In particolare il CE rende uniforme l'accesso ai sistemi di calcolo che per EDG sono il batch processing per l'esecuzione seriale e parallela dei programmi. In altre parole il CE mantiene delle code d'attesa per i programmi da eseguire (quali ad esempio sistemi PBS o LSF); quando arriva il loro turno i programmi possono partire in parallelo su più WN se sono stati scritti con apposite librerie (ad es. MPI), oppure vengono smistati a singoli WN per l'esecuzione esclusiva in quella macchina. Per esempio un ente può mettere a disposizione 100 macchine P-IV a 3GHz; il CE maschererà un sistema di code PBS per le 100 macchine dell'ente; i programmi paralleli che l'utente invierà sfrutteranno i 100 WN, mentre i programmi seriali verranno eseguiti su un singolo WN.

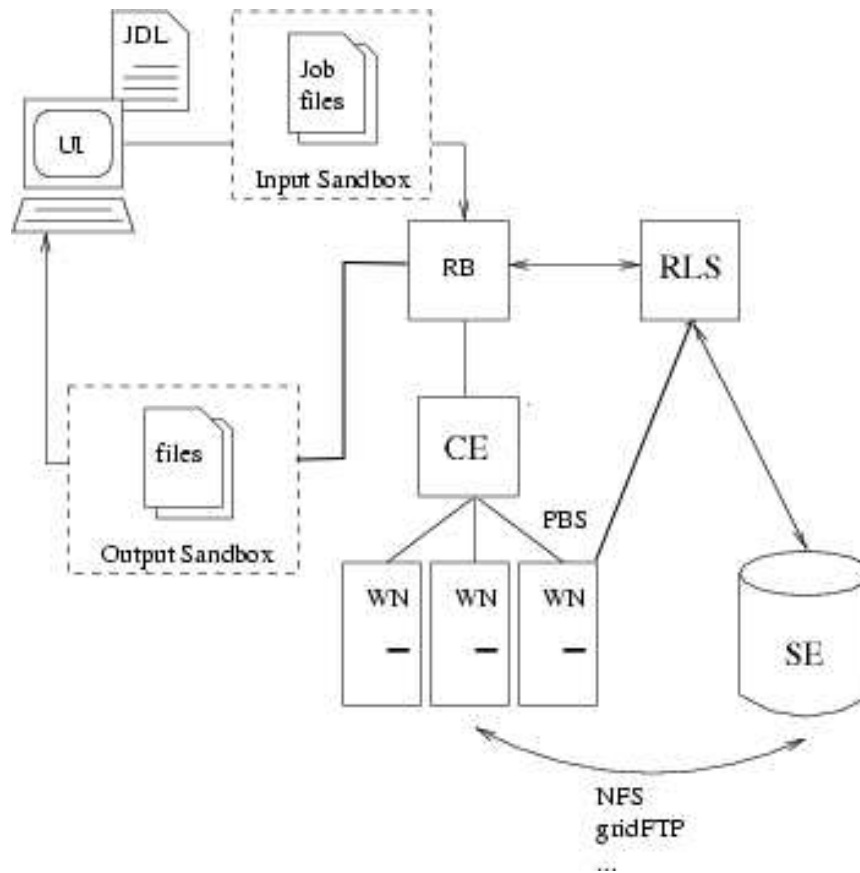


Figura 1: Schema del funzionamento della griglia EDG

1.1.2. SISTEMA INFORMATIVO INTERNO ALLA GRIGLIA

I meccanismi automatici di ricerca/accesso alle risorse di spazio disco e calcolo di EDG si basano su cataloghi che costituiscono il Sistema Informativo della griglia, nel quale vengono raccolte le caratteristiche tecniche dei CE e degli SE (attributi di spazio dello SE, sistema operativo del CE, presenza di gestori di code o librerie parallele, grado d'intasamento del CE, ecc.).

GRIS, GIIS, BDII: ciascun SE pubblica le proprie caratteristiche su un proprio catalogo che prende il nome di Grid Resource Information Service GRIS; invece il CE pubblica le proprie informazioni su un altro catalogo che prende il nome di Grid Index Information Service GIIS; infine le informazioni di tutti i cataloghi finiscono in uno principale detto "Berkeley Database Information Index" (BDII).

1.1.3. MECCANISMO DI FUNZIONAMENTO

Per EDG il funzionamento della griglia prevede di specificare il programma da eseguire ed uno o più file su cui il programma deve lavorare. Automaticamente viene scelto un CE sul quale eseguire il programma, ed uno SE dal quale prelevare i file. Possono venire specificati altri requisiti sul CE quali il numero di CPU libere: vengono determinati i CE che soddisfano tutti i requisiti, e tra questi viene scelto quello più vicino allo SE contenente i file. Queste due azioni, cioè la trasparenza del luogo fisico di residenza di un file ed il mascheramento della macchina che eseguirà il programma, sono ottenuti dai due servizi seguenti che sfruttano il Sistema Informativo.

RLS Replica Location Server: elemento di griglia che fornisce la capacità di specificare un nome file senza preoccuparsi dello SE su cui risiede. Offre inoltre la possibilità di avere più copie del file fisico ciascuna su diversi SE: un meccanismo di nomi logici permette d'identificare un file indipendentemente dalla locazione fisica delle copie.

RB Resource Broker: elemento di griglia con la capacità di scegliere un CE in base a requisiti specificati e di arrivare il più possibile vicino ad uno SE che contenga i file su cui lavorare.

1.1.4. INTERAZIONE CON L'UTENTE

UI User Interface: nodo che permette all'utente di operare all'interno della griglia: l'utente deve quindi collegarsi alla UI per avere accesso completo alle risorse. La UI raccoglie tutti i comandi per caricare i propri file dati in uno SE, per sottoporre l'esecuzione di un programma in griglia cioè la sottomissione di un lavoro, ecc. È la UI che dialoga con l'RLS e con il RB.

Dal punto di vista operativo, l'utente scrive un file di testo nel linguaggio di griglia chiamato "Job Description Language" (JDL) dove vengono specificati i programmi da eseguire, i file richiesti, i requisiti di calcolo, ecc. Tramite i comandi della UI viene trasmesso al RB questo file in JDL, e il RB s'incarica di eseguire le richieste tra cui anche di chiamare l'RLS per capire dove risiedono fisicamente i file. L'utente ha poi ulteriori comandi per monitorare lo stato del lavoro e recuperarne i risultati.

1.1.5. AUTENTICAZIONE E SICUREZZA

Il riconoscimento dell'utente di griglia avviene tramite l'utilizzo di certificati: speciali file contenenti anche i dati dell'utente; ciascun utente ha un certificato proprio. I dati dell'utente all'interno del certificato prendono il nome di Certificate Subject, e rappresentano ciò di cui il certificato attesta la veridicità. All'inizio di una sessione di lavoro in griglia, viene creato un certificato delegato chiamato *Proxy* che ha una valenza limitata nel tempo.

Quando viene eseguito un qualsiasi comando della UI viene verificata prima l'esistenza e validità del Proxy, in tal caso portando a termine il comando; altrimenti l'operazione viene interrotta. I comandi di griglia richiedono in genere l'esecuzione di azioni nei vari nodi che sono sparsi geograficamente: in ciascuno di questi nodi viene spedita assieme alla richiesta di servizio pure un'altro certificato proxy, ulteriormente delegato. Ciascun nodo della griglia prima di fornire il servizio richiesto verifica il Proxy che riceve: se il Certificate Subject del Proxy è presente in una lista locale di autorizzazione, allora l'operazione viene eseguita altrimenti la richiesta viene respinta.

Un insieme di Certificate Subject che hanno accesso alle stesse risorse CE, WN ed SE, prende il nome di "Virtual Organization" (VO): è un concetto che riflette l'organizzazione in gruppi di lavoro, e inoltre facilita all'amministratore delle macchine di griglia la loro gestione.

1.2. INIZIARE A LAVORARE IN GRIGLIA

1.2.1. OPERAZIONI PRELIMINARI

In questa sezione verranno spiegati i passaggi per diventare un utente di griglia, vale a dire come richiedere e configurare un account presso la UI che è il punto di accesso che raccoglie tutti gli strumenti per poterci lavorare, e come richiedere/installare un certificato che fornisce invece l'autorizzazione a lavorarci.

Richiesta e configurazione di un account presso il nodo UI La richiesta deve essere fatta all'Amministratore di Sistema del proprio dipartimento, secondo modalità che variano da istituto a istituto. L'Amministratore di Sistema, all'atto della richiesta dell'account fornirà pure istruzioni sulla configurazione e sull'utilizzo.

Richiesta di un certificato di griglia Presso il proprio istituto deve essere attiva una RA Registration Authority che ha il compito di ricevere le richieste di certificato da parte degli utenti, e di inoltrarle alla CA Certification Authority che è il centro preposto al rilascio dello stesso. Per EGRID la CA è l'INFN-CA (<http://security.fi.infn.it/CA/>).

Il co-ordinatore locale fornirà informazioni su come contattare la propria RA e sulla procedura da seguire. In genere al termine di adempimenti burocratici l'utente riceve un codice personale e un e-mail da INFN-CA che spiega come utilizzare il codice. Le istruzioni consentono di prelevare una copia del certificato tramite l'utilizzo del browser: seguendo le istruzioni il certificato si troverà all'interno del browser.

1.2.2. INSTALLAZIONE DEL CERTIFICATO NELLA UI

Nella pratica l'avere un certificato si traduce nell'avere due file: `usercert.pem` e `userkey.pem`. All'interno del browser si trova un solo file che li raccoglie entrambi: l'operazione per scaricare tale file dall'interno del browser è simile per tutti i tipi di browser, e consiste nel navigare i menu dei comandi fino ad arrivare alla sezione dedicata ai certificati e salvare il file con il nome *infn*. I comandi per estrarre i due file si trovano nella UI; trasferirvi quindi il file e procedere all'estrazione come riportato di seguito.

Estrazione del file `usercert.pem`: `openssl pkcs12 -in infn.p12 -nokeys -out usercert.pem`

Estrazione del file `userkey.pem`: `openssl pkcs12 -in infn.p12 -nocerts -out userkey.pem`

I due file vanno messi in una directory chiamata `.globus` nella propria home sulla UI. Crearla e spostarvi ambedue i file, impostando i permessi di `usercert.pem` a `0644`, e quelli di `userkey.pem` a `0600`.

Per le operazioni di spostamento, creazione directory, collegamento alla UI, impostazione dei permessi, ecc., fare riferimento al proprio Amministratore di Sistema e al sito <http://www.egrid.it/doc/utenti/CA-INFN>.

Va da sé che questa operazione d'installazione del certificato si esegue solo la prima volta e che successivamente si parte dalla sezione 1.2.4..

1.2.3. RICHIESTA DI ABILITAZIONE AL LAVORO IN GRIGLIA EGRID

Una volta completati i passi precedenti l'utente deve essere abilitato nella virtual organization EGRID: l'operazione deve essere fatta a livello centrale dagli amministratori della VO. L'utente deve quindi informare gli amministratori inviando un e-mail all'indirizzo staff@egrid.it contenente nome e cognome del richiedente, eventualmente la preferenza per la username di griglia, e deve essere firmata digitalmente con il proprio certificato.

1.2.4. ATTIVAZIONE E GESTIONE DI UN PROXY DEL CERTIFICATO

I comandi di griglia non lavorano direttamente con il certificato, bensì con una sua delega chiamata *proxy* che deve essere attivata prima di svolgere una qualunque azione in griglia: è l'operazione da svolgere sempre prima di iniziare a lavorare.

Attivazione di un proxy Quando si attiva un proxy in griglia bisogna specificare la durata della sua validità. È importante che sia impostata adeguatamente perchè se scade a metà di un'elaborazione, tutto il lavoro svolto fino a quel momento viene perso.

All'attivazione viene richiesta la passphrase del certificato: una frase segreta specificata al momento della richiesta del certificato, che controlla l'attivazione dei proxy.

Il seguente comando attiva un proxy con la durata di 24 ore:

```
grid-proxy-init -hours 24
```

Risposta:

```
Your identity: /C=IT/L=Trieste - Italy/O=EGRID/OU=ICTP/CN=Ezio Corso
Enter GRID pass phrase for this identity:
Creating proxy ..... Done
Your proxy is valid until: Wed Aug 18 09:56:16 2004
```

Se non viene specificato il parametro -hours, viene impiegato il valore prestabilito di 12 ore.

Eliminazione di un proxy Al termine del lavoro in griglia è consigliabile distruggere il proxy:

```
grid-proxy-destroy
```

I comandi precedenti sono utilizzati sempre per lavorare in griglia e sono perciò i più comuni; ve ne sono altri due che risultano utili: uno per il cambio della passphrase del certificato, e l'altro per ottenere informazioni sul proxy attivato.

Comando per lo stato del proxy:

```
grid-proxy-info
```

Risposta:

```
subject  : /C=IT/L=Trieste - Italy/O=EGRID/OU=ICTP/CN=Ezio Corso/CN=proxy
issuer   : /C=IT/L=Trieste - Italy/O=EGRID/OU=ICTP/CN=Ezio Corso
type     : full
strength : 512 bits
path     : /tmp/x509up_u500
timeleft : 23:59:56
```

Comando per il cambio della passphrase:

```
grid-change-pass-phrase
```

Risposta:

```
read RSA key
Enter PEM pass phrase:
writing RSA key
Enter PEM pass phrase:
Verifying password - Enter PEM pass phrase:
```

2. LAVORARE CON I DATI IN EGRID

In questo capitolo viene spiegato il modo di lavorare con i dati del progetto EGRID all'interno dell'infrastruttura di griglia installata. Nella sezione seguente viene presentata l'organizzazione dei dati assieme a varie convenzioni. Successivamente vengono presentati esempi ed esercizi sui comandi di basso livello EDG, seguiti a ruota dagli strumenti sviluppati autonomamente dal progetto EGRID e che permettono una più facile interazione da parte dell'utente.

2.1. ORGANIZZAZIONE DEI DATI IN EGRID

In questa sezione presentiamo come i dati vengono immagazzinati nella griglia computazionale EGRID, assieme a convenzioni e modi di lavorare.

Dal punto di vista degli SE di EGRID vale a dire della distribuzione fisica dei file, *l'architettura scelta è a stella*: nodo centrale è l'INFN di Padova che fornisce uno SE principale da 2,6TB in cui archiviare in modo affidabile i dati (backup su nastro e alta affidabilità del servizio di storage). I nodi di storage periferici del sistema corrispondono a ciascun gruppo di ricerca e hanno lo scopo di minimizzare lo spostamento d'ingenti quantità di dati all'interno della griglia. Questo per ovviare a possibili limitazioni di banda verso/da Padova. L'utente avrà infatti a disposizione le procedure e strumenti che permettono di ridurre gli spostamenti e quindi mascherare la limitatezza della infrastruttura hardware di rete.

La modalità operativa qui proposta sfrutta gli strumenti forniti da *replica manager*, che si appoggia su una serie di servizi sviluppati all'interno di EDG - **Replica Location Service, Replica Metadata Catalog, Replica Optimization Service**. Questa tecnologia, pur facilitando lo spostamento dei dati fra storage diversi non offre al momento alcuna possibilità di garantire la riservatezza e sicurezza necessarie per i dati del progetto EGRID (si veda a questo proposito la relazione tecnica del progetto). Per ovviare a questi limiti si è elaborata una soluzione provvisoria che cerca di limitare i problemi di sicurezza organizzando i dati nella maniera qui di seguito descritta.

L'organizzazione in directory dei file nello SE è articolata secondo due direttive: consentire agli amministratori dei nodi periferici di organizzare i propri SE come meglio credono in accordo con i propri utenti e secondo proprie esigenze di sicurezza; garantire nello SE principale a Padova un'efficiente organizzazione dei dati e una ragionevole sicurezza di griglia. In quest'ottica nello SE principale di Padova l'organizzazione in directory assieme ai rispettivi permessi sono prefissati, mentre per i nodi periferici viene lasciata piena libertà. Per semplicità lavorativa nei confronti degli utenti viene consigliato in generale di mantenere il proprio SE in sincronia con quello di Padova: a tali fini vengono messi a disposizione dei sistemisti opportuni strumenti.

Nello SE principale di EGRID a Padova, è presente la seguente struttura di directory:

```
--
|-- fonti/
|--   |-- contratto1/
|--   |--   |-- originali/
|--   |--   |--   |-- cd1.tar.gz
|--   |--   |--   |-- cd2.tar.gz
|--   |--   |--   |-- [...]
|--   |--   |-- trasformati
|--   |--   |--   |-- [...]
|--   |-- contratto2/
|--   |-- [...]
|-- [...]
|-- [...]
|-- progetti/
|--   |-- prog1/
|--   |-- prog2/
```

[...]

```
|--utenti/
|   |--ecorso/
|   |   |-- private/
|   |--zoicas/
```

A partire dalla radice vi sono tre directory principali: *fonti per raccogliere i dati acquistati*, progetti per gestire gruppi di ricerca, e *utenti con lo spazio personale (home) in griglia di ciascun ricercatore*.

La directory *fonti* contiene una sottodirectory per ciascun contratto; ciascuna di queste ne contiene altre due: originali e trasformati. La prima raccoglie i contenuti di ogni CD/DVD su cui sono distribuiti i dati, archiviati in un file *.tar.gz* per ciascun supporto. Il contenuto della seconda è lo stesso di originali ma trattato per risultare in una forma standard e utilizzabile dai ricercatori: i singoli file in questa directory hanno nomi assegnati secondo la *Logical File Name Convention* [?] . Ciascuna directory di contratto risulta accessibile in lettura solo agli utenti che ne hanno diritto in quanto, per esempio, facenti parte dell'istituto che ha acquistato i dati; l'amministratore del contratto invece vi accede anche in scrittura per poterci caricare i dati.

La directory *progetti* consente la creazione e la condivisione di sottodirectory tra ricercatori di un medesimo progetto: in questa sottodirectory viene data massima libertà agli utenti per organizzarsi come meglio credono. Il creatore della sottodirectory imposta quali sono i gruppi che vi possono accedere e con quali diritti. Le operazioni di creazione e gestione dei gruppi da parte dei singoli utenti avviene tramite una serie di strumenti sviluppati all'interno del progetto EGRID. Nell'ultimo paragrafo di questo capitolo viene illustrato nel dettaglio un esempio di creazione e gestione di una directory sullo SE centrale tramite tali strumenti (comandi ad interfaccia grafica e a linea di comando).

La directory *utenti* contiene una sottodirectory per ciascun ricercatore che è leggibile da parte di tutti gli utenti EGRID. Al suo interno si trova *private* che invece non è accessibile a nessun altro.

I file caricati nello SE principale vengono, perciò, organizzati all'interno delle strutture mostrate e i *diritti di accesso seguono da dove sono stati collocati*.

Ciascun file fisico, però, è anche individuato da un nome logico che lo identifica in griglia indipendentemente dallo SE su cui risiede, come accennato nel capitolo introduttivo. Il software EDG consente di specificare nomi logici completamente diversi da quelli fisici: se il numero di SE e di file caricati in griglia è elevato la massima flessibilità offerta può essere origine di confusione. Per questo motivo è stata definita una convenzione sul modo di costruire il nome logico da assegnare a ciascun file.

La convenzione sfrutta il fatto che nei nomi logici è consentito l'utilizzo del carattere '/'. Il nome logico da assegnare a un file fisico è costruito semplicemente facendolo coincidere con l'intero percorso nella struttura di directory fisica accennata poc'anzi. Quindi, un nome logico che inizi per *lfn:/utenti/* indica un file collocato nella cartella */utenti/* dello SE principale, e per via della struttura di directory creata i nomi logici per i dati nello SE principale potranno solo iniziare per *lfn:/fonti/*, *lfn:/progetti/* e *lfn:/utenti/*.

È il meccanismo dei nomi logici che consente di evitare di caricare/scaricare ripetutamente lo stesso file da e verso il centro della stella da parte di più utenti dello stesso nodo periferico. In EGRID una qualsiasi operazione di spostamento dati viene suddivisa in due fasi: una prima in cui s'interagisce con SE locale, e una successiva con quello principale. Ad esempio nel caso del caricamento dei CD inizialmente si effettua il trasferimento nello SE del proprio istituto, per poi creare una replica nello SE centrale di Padova in un momento in cui c'è poco traffico in rete come di notte o durante i fine settimana. Analogamente per lo scaricamento di file dati sulla propria macchina UI s'interagirà prima con lo storage centrale e poi con lo storage locale. Periodicamente si controllerà il grado di riempimento degli SE locali e li si svuoterà se necessario, basandosi sul fatto che la copia principale a Padova è sempre disponibile e che comunque è a Padova che avrà luogo il grosso del lavoro sui dati. Queste operazioni sono possibili utilizzando sia i comandi del middleware RLS di EDG che saranno presentati nella prossima sezione, che i comandi EGRID presentati dopo e che eseguono le operazioni in modo più semplice per l'utente.

La capacità di riferirsi a più file fisici col medesimo nome logico comporta una maggior attenzione sulle operazioni che si vogliono eseguire sui file. Infatti si devono adottare due diversi modi d'agire quando interessano delle elaborazioni sui super nodi locali, ma con file di dati inizialmente presenti a Padova.

- Se l'uso del file sarà personale e comporterà la modifica del suo contenuto, conviene che alla copia locale venga assegnato un altro nome logico: gli altri ricercatori non correranno il rischio di lavorare su un file non integro. Se la struttura di directory locale è diversa da quella di Padova si consiglia per convenzione di

assegnare un nome logico che inizi col nome della città scritto in minuscolo, seguito dal percorso completo locale.

- Se invece è prevedibile che più ricercatori operino sui medesimi file, allora conviene eseguire una replica locale che ne conserverà il nome logico ed eviterà ad altri utenti locali il tempo di scaricamento dallo storage principale.

Capire se un file dati da scaricare sullo SE del proprio istituto sarà per uso e modifiche personali oppure per l'utilizzo integro col resto del gruppo di ricerca, non è solo importante per l'assegnazione o meno di un nuovo nome logico. Per poter garantire agli altri ricercatori l'integrità dei dati scaricati, il file dovrà risiedere in una directory in cui gli utenti possono solo leggere ma non scrivere. Il ricercatore deve perciò capire in quale directory del proprio SE dovrà salvare tale file. I permessi sulla struttura di directory locale devono essere impostati con cura dall'amministratore del nodo per garantire la presenza di zone dove è possibile leggere, scrivere o entrambe le operazioni. Il ricercatore in ultima analisi è responsabile per la protezione dei dati scaricati, dovendo far attenzione di non copiare un file riservato in una zona pubblica.

L'organizzazione e l'impostazione dei permessi nelle directory non è un problema banale perché in genere operazioni che nello SE principale di Padova richiedono solo diritti di lettura, si trasformano in operazioni anche di scrittura negli SE locali, con tutti i problemi di sicurezza che ciò comporta. Si rimanda agli appendici dedicati agli amministratori per strategie ed esempi del caso ??.

L'ultima convenzione EGRID sulla gestione dei dati riguarda il caricamento dei CD/DVD dei contratti acquistati dai gruppi di ricerca. Essendo un'operazione sensibile viene prevista la figura dell'*amministratore del contratto*: dell'insieme di utenti appartenenti al gruppo di ricerca, è l'unico con diritto di scrittura nella directory del contratto sullo SE principale, mentre gli altri possono solo leggervi. Il nome della directory che accoglierà i dati andrà comunicato agli amministratori centrali EGRID che provvederanno alla costruzione della struttura ad albero spiegata all'inizio, oltre ad impostare i gruppi utenti con opportuni dritti di accesso. Per esempio se "ctMadrid2435" è il nome scelto per il contratto, a Padova verranno create le directory /fonti/ctMadrid2435, /fonti/ctMadrid2435/originali e /fonti/ctMadrid2435/trasformati; il contenuto andrà copiato sotto *originali*.

I CD/DVD non vengono copiati nella *sottodirectory* originali del contratto così come stanno ma vengono prima preparati opportunamente: questa operazione preliminare serve a garantire un grado d'uniformità per le trasformazioni successive in formati più utili ai ricercatori. Per convenzione il contenuto viene copiato integralmente all'interno di una directory con nome arbitrariamente scelto, di una macchina non in griglia; successivamente questa directory viene compressa in un file tar.gz. È questo file compresso che viene caricato a Padova.

2.2. LAVORARE CON I COMANDI FORNITI DAL REPLICA MANAGER DI EDG

L'insieme dei comandi EDG della UI che gestiscono i file in griglia prende il nome di Replica Manager. Si tratta di strumenti per la gestione dei file fisici residenti negli SE, oltre all'importante funzionalità di assegnazione di un nome logico che è vitale per i meccanismi automatici di calcolo in griglia.

Tramite il Replica Manager si possono infatti:

- scoprire quali sono i SE della propria VO,
- sceglierne uno e caricarvi un file dati,
- assegnare al file un nome logico che lo identifichi nell'intera griglia per le elaborazioni successive,
- di replicarlo in altri SE posti vicino a CE con elevata potenza di elaborazione.
- eliminare repliche ed elencare il contenuto di uno SE.

Per ciascuna delle azioni qui elencate presentiamo nel seguito esercizi opportuni. Si rimanda ai tutorial di INFN-GRID [?] per una serie dettagliata di esempi.

2.2.1. CONOSCERE LA LISTA DI SE DELLA PROPRIA VO E IL LORO ATTRIBUTI

Con questa azione si vuole elencare la lista di tutti gli SE della VO EGRID.

Vengono elencate tutte le informazioni riguardanti la VO, non solo quelle relative agli SE che sono lo spazio disco di griglia a disposizione degli utenti, ma anche per i CE che forniscono la potenza di calcolo alla VO. In particolare

per ciascun SE viene elencato il nome della macchina, oltre alla directory radice a partire dalla quale verranno salvati i file. Questi due parametri verranno impiegati in seguito.

Comando:

```
[UI:]$>edg-rm --vo=egrid printInfo
\end{verbatim}
```

Risposta:

```
\begin{verbatim}
```

```
edg-rm --vo=egrid printInfo
```

```
Have 2SEs
```

```
Have 6CEs
```

```
Information generated at Oct 7, 2004 1:11:53 PM
```

```
=====
```

```
VO used           : egrid
default SE        : egrid-10.egrid.it
default CE        : egrid-2.egrid.it
Info Service      : MDS
```

```
=====
```

Data Management Service information

```
LRC endpoint      : http://egrid-17.egrid.it:8080/egrid/edg-local-replica-catalog/services/edg-local-rep
RMC Endpoint      : http://egrid-17.egrid.it:8080/egrid/edg-replica-metadata-catalog/services/edg-replic
ROS endpoint      :
```

Computing Element information

```
List of CE ID's   : egrid-2.egrid.it:2119/jobmanager-lcgpbs-long,
                   egrid-2.egrid.it:2119/jobmanager-lcgpbs-short,
                   egrid-2.egrid.it:2119/jobmanager-lcgpbs-infinite,
                   egrid-4.egrid.it:2119/jobmanager-lcgpbs-short,
                   egrid-4.egrid.it:2119/jobmanager-lcgpbs-long,
                   egrid-4.egrid.it:2119/jobmanager-lcgpbs-infinite
```

CE egrid-2.egrid.it:2119/jobmanager-lcgpbs-long

```
Name              : long
closeSEs          : egrid-10.egrid.it
VOs               : egrid
```

CE egrid-2.egrid.it:2119/jobmanager-lcgpbs-short

```
Name              : short
closeSEs          : egrid-10.egrid.it
VOs               : egrid
```

CE egrid-2.egrid.it:2119/jobmanager-lcgpbs-infinite

```
Name              : infinite
closeSEs          : egrid-10.egrid.it
VOs               : egrid
```

CE egrid-4.egrid.it:2119/jobmanager-lcgpbs-short

Name : short
closeSEs : egrid-4.egrid.it
VOs : egrid

CE egrid-4.egrid.it:2119/jobmanager-lcgpbs-long

Name : long
closeSEs : egrid-4.egrid.it
VOs : egrid

CE egrid-4.egrid.it:2119/jobmanager-lcgpbs-infinite

Name : infinite
closeSEs : egrid-4.egrid.it
VOs : egrid

Storage Element information

List of SE ID's : egrid-10.egrid.it, egrid-4.egrid.it

SE at EGRID-ICTP

name : EGRID-ICTP
host : **egrid-10.egrid.it**
type : disk
accesspoint : **/flatfiles/SE00**
VOs : egrid
VO dir for egrid : **/egrid**
protocols : gsiftp, rfio

SE at EGRID-ICTP

name : EGRID-ICTP
host : **egrid-4.egrid.it**
type : disk
accesspoint : **/flatfiles/SE00**
VOs : egrid
VO dir for egrid : **/egrid**
protocols : gsiftp, rfio

2.2.2. CARICARE UN FILE DI DATI LOCALE IN UNO SE E REGISTRARLO NEL CATALOGO

Con questa azione si assume che l'utente *ecorso* abbia il file *dati.txt* presente inizialmente nella propria home della UI, cioè il file non sia in griglia. In particolare nella UI il file si trovi in */home/egrid/ecorso/*.

Si vuole quindi caricarlo in uno SE e assegnargli un nome logico in modo tale che il file possa essere utilizzato per elaborazioni successive nei CE+WN, entri cioè a far parte del contesto di griglia. In particolare si è scelto come SE *egrid-10.egrid.it*; in base alle informazioni ottenute dall'esercizio precedente la radice del percorso all'interno dello SE sarà */flatfiles/SE00/egrid/*. Per quanto riguarda la struttura di directory illustrata precedentemente, si vuole salvare il file in *utenti/ecorso/* con nome fisico *dati.txt*.

Coma da convenzione il nome logico da assegnare sarà */utenti/ecorso/dati.txt*.

COMPLETARE!!!!*****

Con questa azione si vuole prendere un file locale dati.txt che si trova nella home directory degli utenti sulla UI (/home/egrid/utente/), e caricarlo sullo SE egrid-10.egrid.it salvandolo nella propria home directory sulla SE principale (/flatfiles/SE00/egrid/utenti/utente/) con lo stesso nome fisico dati.txt. Inoltre gli si vuole assegnare come nome logico parte del percorso assieme allo stesso nome /egrid/utenti/utente/dati.txt

Comando:

```
[UI:]$>edg-rm      --vo=egrid      copyAndRegisterFile \                Nota (0)
file:/home/egrid/ecorso/dati.txt \                Nota (1)
-d   sfm://egrid-10.egrid.it/flatfiles/SE00/egrid/utenti/ecorso/dati.txt \  Nota (2)
-l   lfn:/utenti/ecorso/dati.txt                Nota (3)
```

Risposta:

guid:4b40cd1a-f044-11d8-9ac4-dda9887c9c46 Nota(4)

Spiegazioni:

Nota 0 : L'azione 'copyAndRegisterFile' consente di copiare un file su uno SE specifico, e di registrare un nome logico per renderlo disponibile ai meccanismi automatici di calcolo.

Nota 1 : specificare il nome di file locale col percorso completo. La prima parte è costituita dal prefisso file: che indica un file che risiede in locale; segue il percorso /home/egrid/ecorso/ ed il nome del file dati.txt.

Nota 2 : specificare la destinazione fisica. La prima parte è costituita dal prefisso sfm:// iniziali di Storage File Name; segue il nome dello SE egrid-10.egrid.it. A ruota si ha il percorso completo di dove scrivere il file: la radice /flatfiles/SE00/ che è specifica dello SE, il percorso di directory desiderato /egrid/utenti/ecorso/, infine il nome da assegnare al file dati.txt. (per stabilire SE e radice, seguire l'esempio precedente)

Nota 3 : specificare il nome logico. È il nome con cui identificare in griglia sia il file in quello SE che tutte le repliche in altri SE ed è centrale per il funzionamento dei meccanismi automatici di calcolo ed elaborazione in griglia. Il prefisso è lfn: iniziali di Logical File Name; segue il nome che si desidera dare. Per convenzione EGRID riportata nella sezione precedente, il nome logico contiene tutto il percorso e finisce col nome del file stesso.

Nota 4 viene restituito un identificativo unico del file.

2.2.3. CREAZIONE DI UNA REPLICA IN UN ALTRO SE.

Con questa azione si vuole creare una replica del file appena caricato e registrato, in un secondo SE. Per individuare il file da replicare si utilizza il lfn con cui è stato registrato, e si specifica un SE di destinazione come nell'esempio precedente.

Comando:

```
[UI:]$>edg-rm      --vo=egrid      replicateFile                Nota (0)
lfn:/utenti/ecorso/dati.txt \                Nota (1)
-d   sfm://egrid-4.egrid.it/flatfiles/SE00/egrid/utenti/ecorso/dati.txt  Nota (2)
```

Risposta:

sfm://egrid-4.egrid.it/flatfiles/SE00/egrid/utenti/ecorso/dati.txt Nota(3)

Spiegazioni:

Nota 0 : Replica Manager viene invocato con l'azione di replicateFile che accetterà un file individuato da un nome logico e lo replicherà in un altro SE.

Nota 1 : specificare il nome logico del file da replicare.

Nota 2 : specificare il nome del file fisico.

Nota 3 : la risposta ricevuta coincide con il sfm specificato.

2.2.4. COPIARE UN FILE PRESENTE IN UNO SE, IN UNO DIVERSO ASSEGNANDOLI UN NUOVO NOME LOGICO

Con questa azione si vuole copiare file tra SE ma a differenza del punto precedente in cui viene mantenuto il nome logico, in questa circostanza si desidera assegnarne uno nuovo. E' un esempio di azione discussa nel paragrafo precedente: i dati vogliono essere usati dall'utente singolo e ogni altro utente del nodo non deve usare questa copia che puo' risultare modificata e non devono essere confusi con i dati del repository centrale.

Comando

```
[UI:]$>edg-rm --vo=egrid copyAndRegisterFile \
gsiftp://egrid-10.egrid.it/flatfiles/SE00/egrid/utenti/ecorso/dati.txt \
-d sfn://egrid-4.egrid.it/flatfiles/SE00/egrid/utenti/ecorso/dati2.txt \
-l lfn:/utenti/ecorso/dati2.txt
```

Nota (0)
Nota (1)
Nota (2)
Nota (3)

Risposta:

guid:1f11ed3d-e111-11d8-3dd4-dda9484e9d56 Nota(4)

Spiegazioni:

Nota 0 : il comando da specificare è lo stesso del punto 2.3.2, ma verranno utilizzate opzioni diverse.

Nota 1 : a differenza del caso precedente, il file di partenza va specificato con il prefisso gsiftp:// invece di file:, seguito a ruota dallo SE egrid-10.egrid.it, la radice di quello SE /flatfiles/SE00, ed il percorso completo /egrid/utenti/ecorso/dati.txt.

Nota 2 : viene specificato lo SE di destinazione come già spiegato, e cioè utilizzando il prefisso sfn://, seguito dallo SE egrid-4.egrid.it, dalla radice /flatfiles/SE00, dalla directory che interessa /egrid/utenti/ecorso/ e dal nuovo nome fisico del file dati2.txt.

Nota 3 : nuovo nome logico del file; come da convenzione EGRID lo si fa coincidere con l'intero percorso fisico a cui viene prefissato il nome della città. Nell'esempio qui discusso non e' così perchè lo SE egrid-4 è in sincronia con lo storage principale. Se si fosse lavorato su di un SE con albero di directories diverso il nome logico ricalcherebbe quella struttura.

Nota 4 : viene restituito un codice identificativo unico del file nella griglia.

2.2.5. ELENCO DI TUTTE LE COPIE FISICHE DI UN MEDESIMO FILE LOGICO

Con questa azione è possibile elencare tutti i file fisici che corrispondono ad un nome logico dato.

Comando:

```
[UI:]$>edg-rm --vo=egrid listReplicas \
lfn:/utenti/ecorso/dati.txt
```

Nota(0)
Nota(1)

Risposta:

sfn://kyoto.roma1.infn.it/flatfiles/SE00/egrid/utenti/ecorso/dati.txt Nota(2)
sfn://egrid-10.egrid.it/flatfiles/SE00/egrid/utenti/ecorso/dati.txt

Spiegazioni:

Nota 0 : invocare il comando di Replica Manager con l'azione di listReplicas.

Nota 1 : specificare il nome di file logico.

Nota 2 : viene restituito un elenco di file fisici nel consueto formato sfn://[...] già spiegato in note precedenti.

2.2.6. COPIA DI UN FILE DALLA GRIGLIA NELLA UI LOCALE

Con questa azione si vuole copiare un file di cui si conosce il nome logico dalla griglia al computer locale dove è installata la UI.

Comandi:

```
[UI:]$>edg-rm --vo=egrid copyFile \  
lfn:/utenti/ecorso/dati.txt \  
file:/home/egrid/ecorso/datiScaricato.txt
```

Risposta: [Vuoto]

Da notare che il file locale di arrivo include il percorso completo.

2.2.7. RIMOZIONE DI UN FILE DA UNO SE

Per la rimozione di una copia si fa riferimento direttamente al file fisico, basandosi su informazioni ottenute sulle repliche disponibili con il comando `edg-rm listReplicas` illustrato in precedenza.

Comando:

```
[UI:]$>edg-rm --vo=egrid deleteFile \  
sfn://egrid-4.egrid.it/flatfiles/SE00/egrid/utenti/ecorso/dati.txt
```

Risposta: [vuoto]

2.2.8. ELENCO DEL CONTENUTO DI UNO SE

Può essere utile elencare il contenuto di un determinato SE, per esempio quello principale di Padova per rammentare i nomi dei file caricati.

Comando:

```
[UI:]$>edg-rm --vo=egrid list \  
sfn://egrid-10.egrid.it/flatfiles/SE00/egrid
```

Risposta:

```
fonti  
progetti  
utenti
```

Per la convenzione EGRID spiegata in precedenza, la risposta è costituita dalle tre directory `fonti`, `progetti` e `utenti`. Il comando in sé, però, non fornisce indicazioni sul fatto che si tratta di directory o file, la loro dimensione, ecc.; per questo è utile utilizzare un comando alternativo: `edg-gridftp-ls`

Comando:

```
[UI:]$>edg-gridftp-ls --verbose \  
gsiftp://egrid-10.egrid.it/flatfiles/SE00/egrid
```

Risposta:

```
drwxr-x---    3 egridadm egridusr    4096 Sep 21 23:55 fonti  
drwxr-x--T    2 egridadm egridusr    4096 Sep 23 15:38 progetti  
drwxr-x---   27 egridadm egridusr    4096 Sep  2 12:30 utenti
```

Vengono restituite le informazioni classiche di un comando `ls -l` di Unix.

2.3. COMANDI DI ALTO LIVELLO EGRID

Come si è potuto osservare nella sezione precedente dedicata ai comandi EDG, il loro utilizzo è pesante per via della conoscenza dettagliata sia dell'organizzazione in directory che dei rispettivi permessi, che l'utente deve avere dei vari SE su cui andrà ad operare. Per semplificare il lavoro dell'utente vengono forniti i comandi EGRID: si basano esclusivamente sul nome logico dei file e separano gli SE in due, quello locale chiamato di *cache*, e quello di Padova che rimane il principale *main*. Per raggiungere questi obiettivi di semplificazione i comandi EGRID si basano centralmente sulle convenzioni sui nomi logici, assieme alla struttura locale di directory che deve ricalcare quella di Padova opportunamente impostata con diritti compatibili.

Nota importante per Sys Admin: I comandi EGRID operano basandosi **esclusivamente** sul nome logico del file che deve seguire le convenzioni citate nella sezione 0: il nome logico viene interpretato e, qualora una delle directory non esiste in locale, viene creata al volo. Questo ha conseguenze importanti perché dipende da come si è scelto di organizzare la propria struttura di directory locale: se questa è in sincronia con quella del nodo centrale di Padova, e la gestione degli utenti in gruppi è finemente impostata, allora le operazioni procedono in modo fluido; se invece il nodo periferico è gestito in modo autonomo, i comandi presentati in questa sezione potrebbero avere un comportamento imprevedibile e molto spesso non andare a buon fine.

Per questo è fortemente raccomandato di mantenere sui nodi periferici la stessa struttura di directory presente sul nodo principale ed evitare una gestione personalizzate dello SE locale. In questa maniera gli utenti possono beneficiare di tutti i comandi EGRID senza problemi.

2.3.1. CARICARE UN FILE DATI LOCALE NEL NODO DI CACHE

Comando:

```
[UI:]$>egrid-cache-upload ./dati.txt lfn:/utenti/ecorso/dati.txt
```

Risposta:

```
guid:b5d25a3f-1222-11d9-b902-ac5860cb0bcc
```

Nel comando si specifica il nome locale del file da caricare in cache `./dati.txt` seguito dal nome logico da assegnare `lfn:/utenti/ecorso/dati.txt`. Viene restituito un codice identificativo unico del file in griglia.

2.3.2. REPLICARE UN FILE DALLA CACHE AL NODO PRINCIPALE

Comando:

```
[UI:]$>egrid-replicate-to-main lfn:/utenti/ecorso/dati.txt
```

Risposta:

```
replicating lfn:/utenti/ecorso/dati.txt  
sfn://egrid-10.egrid.it/flatfiles/SE00/egrid/utenti/ecorso/dati.txt
```

Il comando richiede di specificare il nome logico del file che si trova in cache e che si desidera replicare a Padova. Viene restituito un avviso di replica in corso e alla fine viene mostrato il nome fisico assegnato.

2.3.3. REPLICARE UN FILE DAL NODO PRINCIPALE ALLA CACHE

Comando:

```
[UI:]$>egrid-replicate-to-cache lfn:/utenti/ecorso/dati5.txt
```

Risposta:

```
replicating lfn:/utenti/ecorso/dati5.txt  
sfn://egrid-4.egrid.it/flatfiles/SE00/egrid/utenti/ecorso/dati5.txt
```

Il comando richiede di specificare il nome logico del file nello SE principale che si desidera portare in cache; viene restituito un messaggio di replica in corso alla fine della quale si visualizza il nome fisico del file.

2.3.4. COPIARE UN FILE DAL NODO PRINCIPALE A QUELLO DI CACHE, CON UN NUOVO NOME LOGICO

Comando:

```
[UI:]$>egrid-copy-to-cache lfn:/utenti/ecorso/dati2.txt lfn:/utenti/ecorso/copiad2.txt
```

Risposta:

```
guid:88028c5c-15e6-11d9-a163-8aba5c2c0c74
```

Il comando prevede che venga specificato il nome logico del file sorgente, seguito dal nuovo nome logico. Il nuovo file viene creato nel nodo di cache. Viene restituito un codice unico che identifica la copia in griglia.

2.3.5. COPIARE UN FILE DAL NODO DI CACHE A QUELLO PRINCIPALE, CON UN NUOVO NOME LOGICO

Comando:

```
[UI:]$>egrid-copy-to-main lfn:/utenti/ecorso/private/elaborati-locali.txt lfn:/utenti/ecorso/dati10.txt
\end{\verbatim}
```

Risposta:

```
\begin{\verbatim}
```

```
guid:34089d7f-09e7-11aa-b354-8ddd9d2d99
```

Il comando prevede di specificare il nome logico del file di partenza ed il nuovo nome logico da assegnare al file nel nodo principale. Restituisce un identificativo unico del file in griglia.

2.3.6. SCARICARE UN FILE IN LOCALE

Il comando automaticamente verifica la cache e se vi trova il file allora lo scarica altrimenti lo preleva da Padova.

Comando:

```
[UI:]$>egrid-download lfn:/utenti/ecorso/dati.txt
```

Risposta: *[vuoto]*

Il comando non restituisce alcuna risposta e in locale si troverà il file con nome simile a quello logico ma con i caratteri '/' ed ':' sostituiti da '~'. Nell'esempio in corso facendo un listato della directory corrente si troverà il file lfn~utenti~ecorso~dati2.txt.

2.3.7. CAPIRE DOVE SI TROVA UN FILE

Il comando riporta se il file specificato si trova sul nodo principale oppure in cache.

Comando:

```
[UI:]$>egrid-where lfn:/utenti/ecorso/dati.txt
```

Risposta:

```
MAIN   : lfn:/utenti/ecorso/dati5.txt: sfn://egrid-10.egrid.it/flatfiles/SE00/egrid/utenti/ecorso/dati5.txt
CACHE  : lfn:/utenti/ecorso/dati5.txt: sfn://egrid-4.egrid.it/flatfiles/SE00/egrid/utenti/ecorso/dati5.txt
```

Il comando richiede di specificare il nome logico del file e restituisce per ciascuna replica il luogo di residenza (CACHE o MAIN), il nome logico, il nome fisico.

2.3.8. VISUALIZZARE FILE IN CACHE

Il comando elenca tutti i file presenti in cache.

Comando:

```
[UI:]$>egrid-cache-list-files lfn:/utenti/ecorso/dati.txt
```

Risposta:

```
lfn:/utenti/ecorso/dati.txt  
lfn:/utenti/ecorso/dati2.txt
```

Il comando richiede di specificare un nome logico da ricercare nella cache: è possibile inserire caratteri jolly quali '*' e '?' per ricercare più file.

2.3.9. CANCELLARE FILE DALLA CACHE

Il comando consente di rimuovere file dalla cache.

Comando:

```
[UI:]$>egrid-cache-remove lfn:/utenti/ecorso/dati.txt
```

Risposta: *[vuoto]*

Il comando richiede di specificare il nome logico del file che si trova in cache, e se l'operazione va a buon fine non viene restituito alcun messaggio.

2.3.10. ELENCARE I FILE IN GRIGLIA

Comando:

```
[UI:]$>egrid-list-files 'lfn:*' Nota (1)
```

Risposta:

```
lfn:/utenti/zoicas/NYSE-2002-02.tar.gz  
lfn:/utenti/zoicas/NYSE-2002-02-medium.txt  
lfn:/utenti/zoicas/NYSE-2002-02-short.txt
```

Note:

- (1) È sufficiente specificare il nome logico con il prefisso lfn: Vengono accettati i caratteri jolly quali '*' e '?'. Inoltre è importante racchiudere tra apostrofi il nome logico, per evitare possibili problemi con il sistema operativo della UI

2.3.11. ELIMINARE FILE IN GRIGLIA

Con questo comando si possono eliminare tutte le copie di un file, sia in cache che nel nodo principale.

Comando:

```
[UI:]$>egrid-remove-file lfn:/utenti/ecorso/dati.txt Nota (1)
```

Risposta:

Removed all LFNS

Note:

Nota (1) Per cancellare i file non è possibile utilizzare i caratteri jolly quali '*' e '?', ma bisogna specificare il nome logico completo. A differenza del comando per elencare i file in griglia, non serve racchiudere con apostrofi il nome logico.

2.4. COME LAVORARE CON LA DIRECTORY PROGETTI SULLO SE CENTRALE

Come anticipato nel paragrafo uno presentiamo qui un esempio specifico di come lavorare con la directory progetti sullo SE principale di Egrid. Tratteremo un esempio dove un utente generico decide di creare una directory a partire dal path `/flatfi les/SE00/egrid/progetti` e renderla poi disponibile ad un gruppo di collaboratori. Useremo qui gli strumenti egrid seguenti:

1. `egrid-groupm` : tool di management dei gruppi (grafico)
2. `egrid-groupdir`: tool per la creazione delle directory (linea di comando)
3. `egrid-chmod`: tool per cambiare i permessi

2.4.1. EGRID GROUP MANAGEMENT

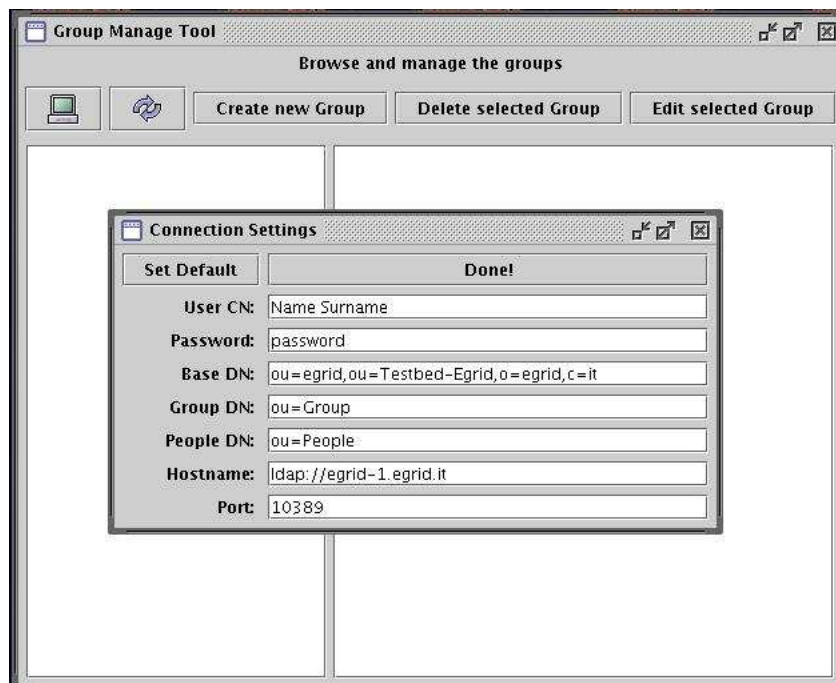
Prima di tutto è necessario creare e popolare il gruppo di lavoro prima di creare la directory in progetti; per questo scopo si utilizzerà il tool di management dei gruppi egrid invocato dal comando:

```
[UI:]$> egrid-groupm
```

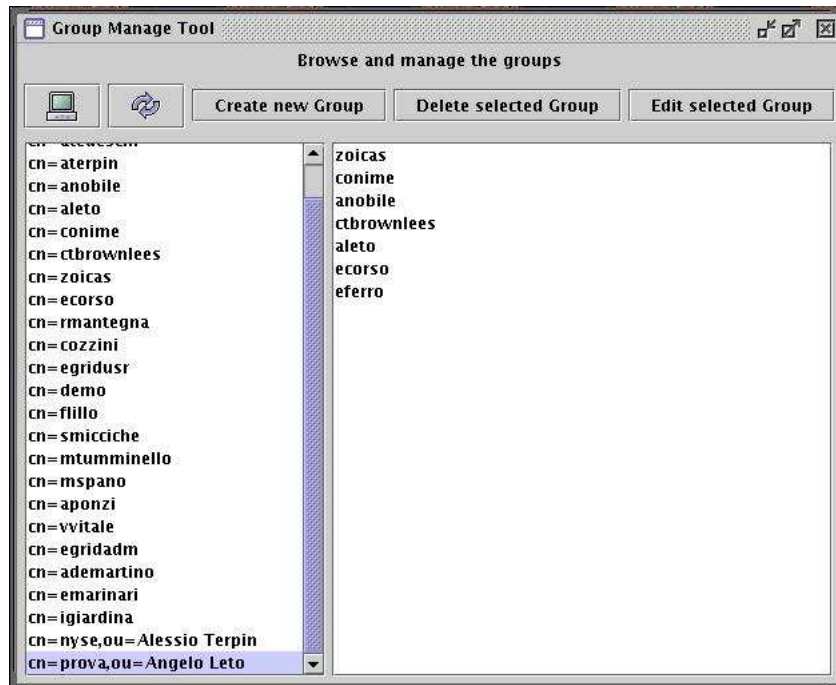
Si tratta di un tool grafico che visualizzerà una finestra come quella visualizzata in figura 1. Nel seguito presentiamo le funzionalità dello strumento.

Browse the groups

1. Settare i parametri di connessione verso il server cliccando sulle prima icona in alto a sinistra del pannello principale



2. sostituire il vostro username e password a quelle di default (che non danno alcun accesso) e cliccare sul pulsante DONE! per rendere effettivi i settaggi
3. cliccare il secondo pulsante in alto a sinistra per iniziare la connessione con il server; tale azione popolerà il lato sinistro del pannello con i gruppi associati alla VO egrid. cliccando su uno dei nomi apparsi pannello di sinistra, si vedrà il pannello di destra mostrare quali membri fanno parte del gruppo selezionato



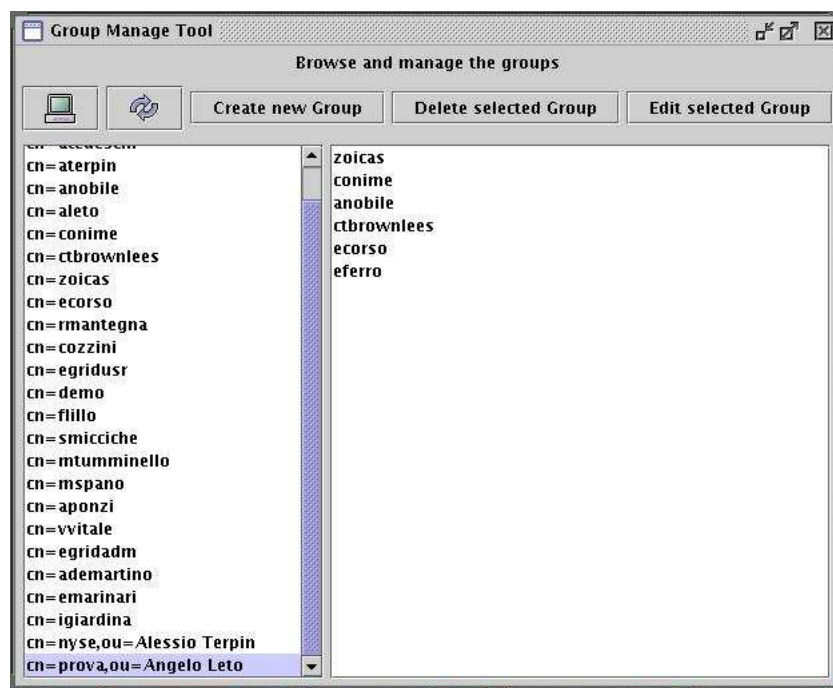
ADD GROUP

Per aggiungere un nuovo gruppo di lavoro bisogna cliccare sul pulsante etichettato Create a new Group, si vedrà apparire un nuovo menù dal quale selezionare i membri che faranno parte del gruppo. Almeno un membro è necessario, poichè la directory relativa al gruppo può essere creata sul filesystem solo da un membro del gruppo, oppure se specificato solo dall'utente indicato nel campo owner del menu di creazione.

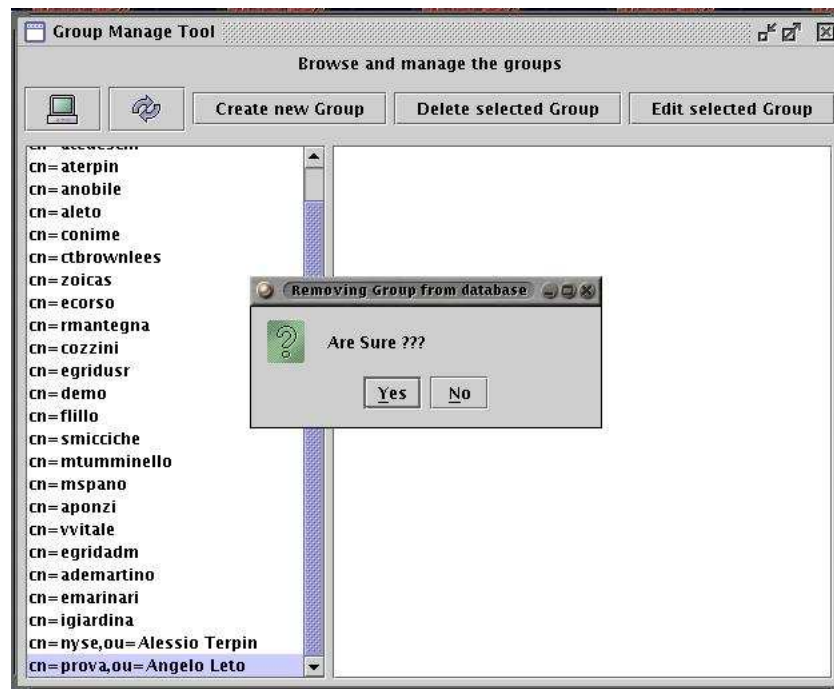
- Gli elementi del menù a sinistra indicano gli utenti che possono essere aggiunti al gruppo, sul lato destro invece quelli che verranno aggiunti al gruppo
- gli utenti selezionati possono essere spostati da un pannello all'altro fino alla conferma di fine operazione da parte del creatore



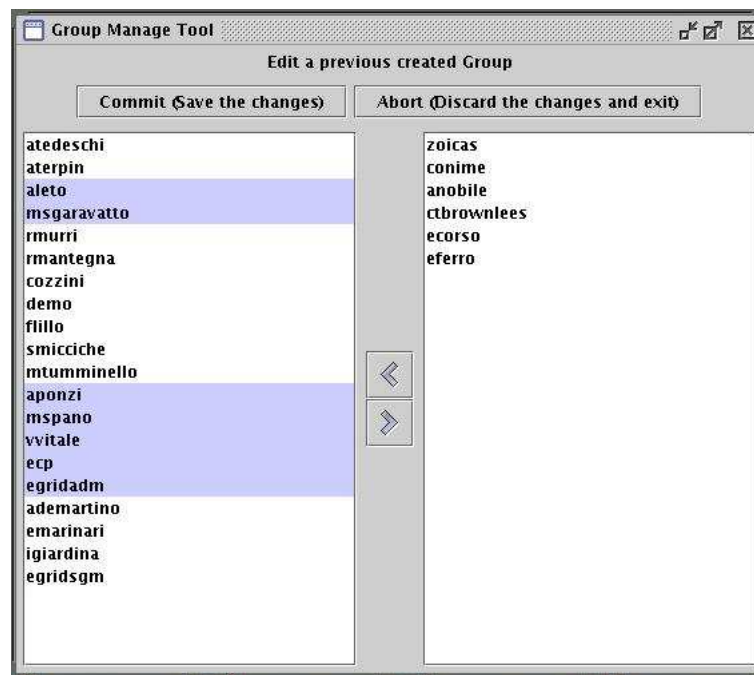
- Il campo group name non può essere vuoto, deve essere un nome avente meno di 32 caratteri, senza caratteri speciali
- Il campo owner specifica con quale owner deve essere creata la directory del gruppo sul filesystem; se questa informazione non è specificata l'ownership della directory viene assegnata al creatore della directory (se questi appartiene al gruppo)
- Finite le operazioni di popolamento del gruppo (che non può essere vuoto) cliccare sul pulsante save and exit per rendere effettivi i cambiamenti. Al ritorno sul menù principale per visualizzare il nuovo gruppo aggiunto è necessario effettuare un refresh dei dati tramite il secondo pulsante il alto a sinistra del menù principale. La figura seguente mostra la creazione del gruppo prova



DELETE Group È possibile cancellare un gruppo solo se si è il proprietario (lo si è creato), selezionandolo dalla parte destra del menù principale e cliccando sul pulsante Delete the selected Group.



EDIT Group È possibile aggiungere o sottrarre dinamicamente utenti da un gruppo (solo se si è il creatore) attraverso la voce Edit the selected Group del menù principale. Viene mostrato un menù simile a quello di creazione Gruppo dal quale è possibile gestire gli utenti sul gruppo creato.



Concluse le operazioni si può scegliere se applicarle al gruppo tramite il pulsante etichettato Commit (Save the changes) o se terminare senza alterare il gruppo tramite il pulsante Abort. Si noti che solo i gruppi chiamati come un utente possono non mostrare membri, dato che hanno automaticamente come membro l'utente corrispondente.

2.5. CREAZIONE DELLA DIRECTORY RELATIVA AL GRUPPO

Dopo aver creato il gruppo è necessario creare la directory in progetti tale operazione va fatta tramite il tool a riga di comando egrid-groupdir

```
[UI]$> egrid-groupdir
too few parameters
```

```
USAGE: ./egrid-groupdir [--se SE_hostname] [--create dirname | --ls]
      -h, --help           : this help
      --se                 : storage element hostname
                          if omitted the default value will be MainSE hostname
      -c, --create dirname : create the directory dirname
      --ls                 : list content of "progetti" dir.
```

L'opzione -ls mostra il contenuto della directory progetti ad es:

```
[UI]$> egrid-groupdir --ls
total 0
```

la directory progetti sullo storage principale e' vuota.

Per creare una directory bisogna utilizzare l'opzione -c o -create seguita dal nome del gruppo precedentemente creato tramite il tool egrid-groupm es:

```
[UI]$> egrid-groupdir -c prov
220 egrid-10.egrid.it GridFTP Server 1.5 GSSAPI type Globus/GSI wu-2.6.2 (gcc32dbg,1032298778-28) ready
230 User aleto logged in.
200-egrid-mkgroupdir.pl --dirname=prov
200-LDAP search returned 0 entries
200-check if the group really exist!
200 (end of 'egrid-mkgroupdir.pl -dirname=prov')
```

Il comando mostra un messaggio che indica che il gruppo immesso prov non è stato trovato tra quelli esistenti

(riga 4: 200-LDAP search returned 0 entries)

la directory non è stata creata. Questo può accadere per due ragioni:

1. Il gruppo non e' mai stato creato, l'errore si presentera' ogni qualvolta si tenti di creare tale directory.
2. Il gruppo e' stato creato, ma le informazioni non si sono propagate dal server sul quale e' stato creato il gruppo, a quello utilizzato per la verifica dell'esistenza dello stesso; in tal caso il messaggio cesserà di apparire. Per questa ragione e' consigliabile attendere un tempo maggiore o uguale a 120 secondi a partire dalla creazione del gruppo prima di creare la directory.

Tentiamo ora di creare la directory immettendo il nome corretto del gruppo prova

```
[UI]$> egrid-groupdir -c prova
220 egrid-10.egrid.it GridFTP Server 1.5 GSSAPI type Globus/GSI wu-2.6.2 (gcc32dbg, 1032298778-28) ready
230 User aleto logged in.
200-egrid-mkgroupdir.pl --dirname=prova
200-You are not a group's member: prova
200 (end of 'egrid-mkgroupdir.pl --dirname=prova')
```

Questa volta il messaggio indica che la directory non è stata creata perchè non si è membri del gruppo prova; riproviamo dopo aver aggiunto noi stessi al gruppo prova:

```
[UI]$> egrid-groupdir -c prova
220 egrid-10.egrid.it GridFTP Server 1.5 GSSAPI type Globus/GSI wu-2.6.2 (gcc32dbg, 1032298778-28) ready
230 User aleto logged in.
200-egrid-mkgroupdir.pl --dirname=prova
200-Create Directory
200 (end of 'egrid-mkgroupdir.pl --dirname=prova')
```

come è possibile vedere la directory viene correttamente creata:

```
[UI]$> egrid-groupdir --ls
total 4
drwxrwx---  2 owner    prova          4096 Oct  4 10:40 prova
```

Il proprietario della directory è l'utente owner che è anche l'utente che ha eseguito il comando. Se invece in fase di creazione del gruppo si specifica il proprietario, la directory creata avrà un owner indipendentemente da chi avrà creato la directory es:

```
[UI]$> egrid-groupdir --ls
total 4
drwxrwx---  2 proprietario prova      4096 Oct  4 10:48 prova
```

```
[UI]$> whoami
owner
```

Il proprietario del gruppo è diverso dal nome utente che crea la directory. Tutti gli esempi sopra elencati, rimangono validi se si specifica tramite l'opzione `-se` il nome dell'hostname sul quale si eseguono le operazioni. In assenza di tale parametro lo script farà riferimento allo Storage element specificato dal file di configurazione: `/opt/egrid/etc/egrid.vo.conf` dalla variabile: `MAIN_SE_FQH` (main storage element fully qualified hostname) Le directory create in questo modo possono essere rimosse solo dall'amministratore di sistema, inoltre poiché il sistema di autorizzazione si basa sul meccanismo di autorizzazione di UNIX senza posix ACL ad ogni utente non è concesso di essere membro di più di 32 gruppi contemporaneamente; per questa ragione è necessario gestire con parsimonia tale funzionalità.

2.6. PERMESSI SU FILES E DIRECTORY

La procedura gestione dei permessi sulla directory creata è del tutto uguale a quella operata sul resto del filesystem per il quale è possibile utilizzare il comando `egrid-chmod` che utilizza la direttiva `SITE` di `gridFTP` come illustrato di seguito. Il comando `egrid-chmod` permette di cambiare/impostare i permessi (intesi nel senso di unix) a files e directory:

```
[UI]$> egrid-chmod
too few parameters
USAGE:
/opt/egrid/bin/egrid-chmod [-m mode -H hostname -f PATH] | [-m mode --lfn logical finename]
    -h, --help                : this help
    -H, --hostname            : storage element hostname
    -m, --mode                : unix permission mode
    -f, --path                : path of target file
    --lfn                     : target logical file name
    --sfn                     : storage file name
```

Il comando accetta come input tre diversi modi per specificare i files/dir. destinatari/e:

Hostname PATH : fully qualified hostname dello storage e path completo del file sul filesystem.

storage file name : nome nella forma `sfn://fqh/path`

logical file name : nome nella forma `lfn:nomelogicofile`

In questo caso il programma tenta di modificare i permessi di tutte le istanze fisiche del nome logico specificato, quindi potrebbe fallire per i files sui quali non si è autorizzati ad effettuare `chmod`.

Esempi:

- Logical file name

```
[UI]$> egrid-where lfn:/test/xx
SE   : lfn:/test/xx: sfn://egrid-10.egrid.it/tmp/xx
```

```
[UI]$> edg-gridftp-ls --verbose gsiftp://egrid-10.egrid.it/tmp/xx
-rwx-----  1 aleto    aleto          0 Oct  5 10:52 /tmp/xx
```

```
[UI]$> egrid-chmod -m 755 --lfn lfn:/test/xx
sfn://egrid-10.egrid.it/tmp/xx
220 egrid-10.egrid.it GridFTP Server 1.5 GSSAPI type Globus/GSI wu-2.6.2 (gcc32dbg, 1032298778-28)
230-No directory! Logging in with home=/
230 User aleto logged in.
200 CHMOD command successful.
```

```
[UI]$> edg-gridftp-ls --verbose gsiftp://egrid-10.egrid.it/tmp/xx
-rwxr-xr-x    1 aleto    aleto          0 Oct  5 10:57 /tmp/xx
```

- Storage file name:

```
[UI]$> egrid-chmod -m 444 --sfn sfn://egrid-10.egrid.it/tmp/xx
220 egrid-10.egrid.it GridFTP Server 1.5 GSSAPI type Globus/GSI wu-2.6.2 (gcc32dbg, 1032298778-28)
230-No directory! Logging in with home=/
230 User aleto logged in.
200 CHMOD command successful.
```

```
[UI]$> edg-gridftp-ls --verbose gsiftp://egrid-10.egrid.it/tmp/xx
-r--r--r--    1 aleto    aleto          0 Oct  5 10:57 /tmp/xx
```

3. ESEMPI DI JOB SUBMISSION IN EGRID

L'insieme dei comandi della UI per l'esecuzione dei programmi in griglia prende il nome di *job submission*. In questa sezione presentiamo il meccanismo di job submission all'interno di Egrid ed alcuni esempi di come si usano i comandi deputati allo scopo.

Va notato che questa che il middleware necessario viene interamente fornito dal software EDG: in questa sezione si presentano quindi alcuni esercizi generali di EDG (si veda il tutorial opportuno) accompagnati da alcuni esercizi azioni specifici del progetto Egrid. La sezione seguente presenta nel dettaglio il meccanismo di sottomissione dei jobs, mentre le sezioni seguenti sono invece dedicate ad azioni/esercizi.

3.1. DESCRIZIONE DEL MECCANISMO DI SOTTOMISSIONE DEI JOB

Nel contesto di griglia la sottomissione di un job richiede l'interazione tra vari sistemi, che sono:

- UI Interfaccia Utente ove l'utente interagisce
- RB Resource Broker che si occupa di trovare la risorsa migliore dove inviare il job e di gestire il controllo dello stesso. L'RB è composto da vari sottosistemi
 - NS **Network Server** provvede ad accettare le richieste di sottomissione dei job
 - WM **Workload Manager** coordina le varie parti
 - MM **Match Maker** provvede a contattare le risorse *esterne* BDII e RLS per scegliere il CE a cui sottomettere il Job
 - JA **Job Adapter** si occupa di costruire il wrapper che verrà sottomesso al CE
 - JC **Job Controller** si occupa di sottomettere il job al CE tramite **condorG**
 - LM **Log Monitor** provvede a controllare lo stato dei log di condorG
 - LB **Logging e Booking service** che mantiene le informazioni relative allo stato dei Job, concentra e raccoglie le informazioni dai vari sistemi.
 - **Sandbox** il meccanismo che permette lo staging dei file, dalla UI
- RLS Replica Location Service che mantiene le informazioni relative ai file e nomi logici
- BDII Berkley Databe Information Service che mantiene le informazioni relative alle risorse disponibili in griglia
- CE Computing Element gateway verso i WN normalmente un sistema di code PBS o LSF ove avviene
- WN Worker Node ove avviene l'effettiva elaborazione del job
- SE Storage Element ove i dati sono salvati.

La sottomissione di un job avviene utilizzando i comandi messi a disposizione di LCG-2/EDG. I principali sono: La sottomissione di un job richiede la creazione di un file in formato JDL (Job Description Language) che descrive il job, elencando le risorse necessarie per la sua esecuzione, le informazioni saranno poi processate dal RB per trovare la risorsa migliore. Ogni job che è stato sottomesso in griglia è indentificato da una stringa univoca chiamata

edg-job-submit job.jdl	Sottomissione di un job
edg-job-status JOBID	Controllo dello stato di un job
edg-job-get-output JOBID	Ritiro dell'output
edg-job-cancel JOBID	Cancellazione di un Job

Tabella 1: I principali comandi per la sottomissione di job

JOBID che serve per interagire con lo stesso.

3.2. SCENARI DI UTILIZZO DELLA JOB SUBMISSION IN EGRID

Il meccanismo di sottomissione job viene usato all'interno di Egrid per molteplici azioni tra le quali citiamo:

1. esecuzioni di programmi di simulazioni di lunga durata.
2. installazione di Software sviluppato in locale dagli utenti che devono essere messe a disposizione in griglia per un suo successivo utilizzo come nel punto precedente.
3. azioni di trasformazioni sui dati immagazzinati in griglia secondo quanto discusso nel capitolo precedente.

3.3. FASI DI SOTTOMISSIONE DI UN JOB IN GRIGLIA

Ogni sottomissione di un job, indipendentemente dal tipo è composta dalle seguenti fasi:

1. Scrittura del file JDL
2. Sottomissione del Job `edg-job-submit`
3. Monitoring dello stato `edg-job-status`
4. Ritiro del risultato `edg-job-get-output`

3.4. SCRITTURA DEL FILE JDL

Un file JDL è un file di testo contenente la descrizione del Job si vuole sottomettere in griglia, è composto da varie coppie Attributo valore. Per una lista completa degli attributi vedere [JDL Attributes -DataGrid-01-TEN-0142-0-2] Introduciamo i vari attributi tramite i seguenti esempi di complessità crescente.

1. Lanciare un comando presente sul Worker Node
2. Caricamento di un eseguibile Locale per utilizzo in Griglia
3. Utilizzo di Standard Input in Griglia
4. Richiesta di utilizzo di Software di esperimento
5. Gestione di LFN in griglia

3.4.1. LANCIARE UN COMANDO PRESENTE SU UN WORKER NODE

Lanceremo il comando `/bin/echo` presente sul worker node per fare stampare hello. `Hello.jdl`

```
[
Executable ="/bin/echo";
Arguments = "hello";
StdOutput = "stdout";
StdError = "stderr";
OutputSandbox = {"stdout", "stderr"};
]
```

Analizziamo ora gli Attributi presenti:

- **Executable:** è il nome del comando remoto che si desidera utilizzare, deve essere indicato con il **path assoluto**.
- **Arguments:** gli argomenti che vengono passati al comando specificato in Executable.
- **StdOutput:** il nome del file remoto che contiene lo stdout del comando eseguito.
- **StdError:** il nome del file remoto che contiene lo stderr del comando eseguito.

- **OutputSandbox:** Indica la lista dei file generati sul WorkerNode che desideriamo poter ritirare il locale sulla nostra UI.

Nel caso che non si sappia dove sia stato installato un eseguibile si può usare come **Executable** il comando `/usr/bin/env` con **Arguments** il comando e i parametri dello stesso, in questo modo il comando viene ricercato all'interno della variabile **PATH** del WorkerNode. Per esempio vediamo quale versione della JVM è installata in griglia.

Il file JavaVersion.jdl

```
[
Executable ="/usr/bin/env";
Arguments = "java -version";
StdOutput = "stdout";
StdError="stderr";
OutputSandbox = {"stdout", "stderr"};
]
```

3.4.2. ESECUZIONE DI UNO SCRIPT/PROGRAMMA PREPARATO IN LOCALE CHE CREA UN FILE DI OUTPUT

Dovremo quindi specificare nel file jdl il fatto che bisogna caricare un file dalla UI, per fare cio è presente l'attributo **InputSandbox**.

Il file RunInfo.jdl

```
[
Executable="RunInfo.sh";
InputSandbox="RunInfo.sh";
StdOutput="stdout";
StdError="stderr";
OutputSandbox={"stdout", "stderr", "risultato/dati.txt"};
]
```

InputSandbox: Indica lista di file che deve venir caricata in griglia, permette l'utilizzo di WildCard * e ?, con significato analogo a quello della Cshell

Inoltre il comando da eseguire è senza path in quanto verra eseguito nella directory creata sul worker node, dal meccanismo di jobsubmission.

!!!!!! job-submission.tex Attenzione: Al momento della sottomissione del job non sappiamo in quale directory del WN andrà a girare il nostro programma, infatti essa è determinata runtime in base, all'utente sui cui viene mappato il certificato ¹ e dal JOBID. Quindi file di output devono avere percorsi relativi e non assoluti. Cio' permette al WMS di ripulire tutto l'output prodotto durante l'elaborazione.

RunInfo.sh

```
#!/bin/sh
#
# Crea il file dati.txt riportante alcuni dati inerenti al worker node
# dove \e stato eseguito
#
mkdir risultato
echo "WorkerNode: 'hostname'" > risultato/dati.txt
echo "username: \${LOGNAME}" >> risultato/dati.txt
```

¹utente del pool account egrid sui WN di Padova oppure utente nominativo sui supernodi

3.4.3. ESECUZIONE DI UN JOB CHE ACCETTA DEI PARAMETRI IN STDINPUT E CREA UN NUMERO NON DEFINITO DI FILE DI USCITA

Consideriamo il caso in cui abbiamo un script o programma locale che legge dei parametri dallo standard Input e desideriamo lanciare in griglia senza modificarlo, per fare ciò è presente l'attributo **StdInput** che specifica il nome del file contenente i parametri che verranno passati all'eseguibile.

Il seguente script, chiede interattivamente il numero di file che deve creare, ed un loro prefisso, li crea con il nome prefisso.%.OUT e prepara un tar.gz per poter ritirare il risultato.

Poiché il numero dei file di uscita e il loro nome è funzione dei parametri di ingresso, presenti nel file dei comandi, e non è possibile utilizzare le wildcard nella OutputSandbox, conviene creare un file con un nome fisso, con cui poter ritirare il risultato, in modo da poter riutilizzare lo stesso file jdl per varie sottomissioni cambiando solamente i dati all'interno del file di comandi.

Lo script StandardInput.sh

```
#!/bin/sh
echo "Numero di file da creare"
read N
echo "prefisso del file"
read prefisso

for i in `seq 1 $N`
do
    echo "hello" > ${prefisso}.$i.OUT
done
tar cvfz OUT.tar.gz *.OUT
```

Prepariamo il file dei comandi che contiene uno per riga i parametri che interattivamente inseriamo dallo StandardInput.

```
StandardInputComandi.txt
10
prova
```

In questo caso creeremo i file prova.1.OUT ... prova.10.OUT

StandardInput.jdl

```
[
Executable="StandardInput.sh";
InputSandbox={"StandardInput.*"};          # Utilizziamo la wildcard * per i file
StdInput = "StandardInputComandi.txt";
StdOutput="stdout";
StdError="stderr";
OutputSandbox={"stdout","stderr","OUT.targ.gz"};
]
```

StdInput: Indica il file che verrà utilizzato come Standard Input dell'eseguibile. verrà eseguito il griglia il comando executable ; "standardinputfile"

3.4.4. UTILIZZO DI SOFTWARE DI EGRID

In griglia è possibile avere del Software specifico per ogni VO, la presenza di tale software è pubblicata sul BDII da ogni CE che lo renda disponibile.

Per ottenere la lista del software disponibile si può utilizzare il comando egrid-software che lo elencherà.

```
$ egrid-software
```

Egrid Software

First field is the Tag of the software

Second field is the name of the software

V0-egrid-python2.3	python2.3
V0-egrid-egrid-transform-dispatcher	egrid-transform-dispatcher
V0-egrid-transform-ls	transform-ls
V0-egrid-transform-grep-lfns	transform-grep-lfns
V0-egrid-transform-taq-trade	transform-taq-trade

Per utilizzare un Software di esperimento è necessario specificarlo nel file jdl, tramite l'attributo Requirements con la seguente sintassi

```
Requirements =Member("V0-egrid-Python_2.3_1",  
                    other.GlueHostApplicationSoftwareRunTimeEnvironment);
```

Cio' permette al RB nella fase di matchmaking di trovare i CE che lo hanno installato, chiedendo tale informazione al BDII.

Per poter usare ad esempio la versione 2.3 del Python, è necessario fare in modo che il l'eseguibile sia presente nel PATH di sistema del WN, perciò è necessario modificare l'enviroment di esecuzione, tramite l'attributo Environment del jdl. La variabile VO_EGRID_SW_DIR contiene la directory ove è installato il software dell'esperimento, per convenzione gli eseguibili si trovano in ;NOME_Programma/bin Ptest.jdl

```
[  
Executable = "ptest.py";  
StdOutput = "stdout";  
StdError = "stderr";  
InputSandbox = "./ptest.py";  
OutputSandbox = {"stdout","stderr"};  
Environment = "PATH=$PATH:$VO_EGRID_SW_DIR/python2.3/bin";  
Requirements = Member("V0-egrid-python2.3",  
                    other.GlueHostApplicationSoftwareRunTimeEnvironment);  
]
```

ptest.py

```
#!/usr/bin/env python2.3  
import sys  
print sys.version
```

3.4.5. GESTIONE DEI LFN IN GRIGLIA

Nota su InputSandbox e OutputSandbox

Questi due attributi servono per trasferire file di piccole dimensioni dell'ordine di qualche Megabyte: se si dovesse caricare file molto grandi si potrebbe saturare il disco del RB e di conseguenza l'intera griglia smetterebbe di funzionare (il problema della saturazione del RB nasce dal fatto che tutti gli utenti passano per un unico RB, da cui la facilità di riempimento). Anche se è possibile trasferire file su cui eseguire elaborazioni tramite la InputSandbox, e recuperare file creati dal programma tramite l'OutputSandbox, tale prassi, mostreremo come utilizzare i comandi di replica all'interno dei job che sottoponiamo in griglia in modo da effettuare lo spostamento e la registrazione dei risultati ottenuti come nomi logici, e gli attributi jdl **InputData** e **OutputData** per la gestione dei dati.

Consideriamo il seguente caso:

È presente nella home di griglia dell'utente nello storage principale il file lfn:/utenti/<username>/dati.txt che contiene dei valori che l'utente desidera processare, tramite lo eseguibile processa.sh presente in locale, e poi salvare il file di risultati nella sua home di griglia con il nome logico lfn:/utenti/username/risultati.txt sia sullo storage principale che sul suo supernodo.

Lo script dovrà

1. recuperare il file dallo storage Element principale tramite il comando `edg-rm`,
2. eseguire il comando `processa.sh`
3. copiare e registrare il risultato nello storage Principale
4. replicarlo sul supernodo dell'utente.

lo script `Replica.sh`

```
#!/bin/sh
# Definizione degli storage
# Per Padova egrid-se-01.pd.infn.it
PRINCIPALE=egrid-10.egrid.it
LOCALE=egrid-4.egrid.it

# Copia del file sul WN
edg-rm --vo egrid cp lfn:/utenti/username/dati.txt \
file:/// 'pwd'/dati.txt

# Creazione del file risultati
chmod +x processa.sh # solamante Executable ha permesso di esecuzione
./processa.sh dati.txt > risultati.txt # sono presenti nella stessa dir

# Registro il file sullo storage principale
edg-rm -vo egrid cr file://\${PWD}/risultati.txt \
-l lfn:/utenti/username/risultati.txt \
-d sfn://\${PRINCIPALE}/flatfiles/SE00/egrid/username/risultati.txt

# Replica del file sul supernodo locale
edg-rm -vo egrid rep lfn:/utenti/username/risultati.txt \
-d sfn://\${PRINCIPALE}/flatfiles/SE00/egrid/username/risultati.txt
```

Nota: Durante la fase di trasferimento del file `processa.sh` viene perso il permesso di eseguibilità, quindi bisogna ridarlo prima di poterlo utilizzare la fase di replica sul supernodo è possibile se esiste la stessa struttura di directory dello storage principale, in quanto il meccanismo di sicurezza di egrid si basa sui permessi UNIX [vedi sezione gestione dati].

`processa.sh`

```
#!/bin/sh
sorti -n \ $1
```

Per assicurarci che il job vada a finire in un CE vicino ai dati useremo l'attributo **InputData**, assieme al quale bisogna specificare anche l'attributo **DataAccessProtocol**, scegliamo il protocollo **gsiftp**, poichè permette l'implementazione del meccanismo di sicurezza dei dati.

`Replica.jdl`

```
[
InputData          = "lfn:/utenti/aterpin/dati.txt";
DataAccessProtocol = "gsiftp";
Executable         = "Replica.sh";
InputSandbox       = "Replica.sh,processa.sh";
StdOutput          = "stdout";
StdError           = "stderr";
OutputSandbox      = {"stdout","stderr"};
]
```

InputData: Specifica il LFN del file dei dati. **DataAccessProtocol:** Specifica il protocollo con cui accedere ai dati.

Nel caso che non si voglia effettuare la replica dei dati di uscita, ma soltanto la loro registrazione, è possibile usare l'attributo, **OutputData** che serve a specificare dove i file elaborati dal WN verranno salvati, ciò introduce un'asimmetria nel nostro script, infatti dovremmo scaricare il file specificato in **InputData**, ma la parte di copy and register di edg-rm verrà effettuata in automatico, poi bisognerà provvedere a replicare manualmente il file in locale.

Registrazione.sh

```
#!/bin/sh
# Copia del file sul WN
edg-rm --vo egrid cp lfn:/utenti/username/dati.txt \
file:/'pwd'/dati.txt
# Creazione del file risultati
chmod +x processa.sh          # solamente Executable ha permesso di esecuzione
./processa.sh dati.txt > risultati.txt # sono presenti nella stessa dir
```

Registrazione.jdl

```
[
  InputData          = "lfn:/utenti/aterpin/dati.txt";
  DataAccessProtocol = "gsiftp";
  OutputData={
    [
      OutputFile      = "risultati2.txt";
      StorageElement  = "sfn://egrid-10.egrid.it/flatfiles/SE00/egrid/aterpin/risultati2.txt";
      LogicalFileName = "lfn:/utenti/aterpin/risultati2.txt";
    ]
  };
  Executable         = "Registrazione.sh";
  InputSandbox       = "Registrazione.sh,processa.sh";
  StdOutput          = "stdout";
  StdError           = "stderr";
  OutputSandbox      = {"stdout", "stderr"};
]
```

OutputData: Specifica come verrà salvato il file di uscita tramite il comando edg-rm lanciato dal Wrapper, bisogna specificare, **OutputFile** il nome del file che deve essere salvato, **StorageElement** dobbiamo specificare il nome fisico completo, altrimenti il sistema di sicurezza non ci permetterà di scrivere il file. **LogicalFileName**.

È importante notare che avviene solamente la parte di registrazione dei dati, e che poi dovranno essere replicati in locale.

Lo stato della avvenuta Registrazione dei dati viene riportato in maniera automatica nel file DSUpload_{\$JOBID}.out che verrà scaricato dalla Sandbox, in caso di errore la notifica sarà presente solamente in questo file.

3.5. SOTTOMISSIONE DI UN JOB IN GRIGLIA

Dopo che abbiamo scritto il file jdl, assicurati di avere un proxy valido, dobbiamo sottomettere il job in griglia.

Il comando da usare

```
edg-job-submit [opzioni] <file.jdl>
```

utili sono l'opzione -o({jobid}) che permette di salvare in un file, l'identificativo del job, per poterlo poi utilizzare come input per gli altri comandi tramite l'opzione -ijobid, e l'opzione -r che permette di scegliere quale CE utilizzerà, bisogna da utilizzare con cautela in quanto elimina la fase di Matchmaking da parte del RB quindi bisogna essere sicuri che i **Requirements** specificati siano soddisfatti sul CE che sceglieremo ed anche la vicinanza dei dati dovrà essere considerata.

lanciando il comando il job verrà sottomesso

```
edg-job-submit -o jobid RunInfo.jdl
Selected Virtual Organisation name (from UI conf file): egrid
Connecting to host egrid-5.egrid.it, port 7772
Logging to host egrid-5.egrid.it, port 9002

===== edg-job-submit Success =====
The job has been successfully submitted to the Network Server.
Use edg-job-status command to check job current status. Your job identifier (edg_jobId) is:

- https://egrid-5.egrid.it:9000/Pctw8IBXvphFeTNfUSKXFQ

The edg_jobId has been saved in the following file:
/home/egrid/aterpin/tutorial/Esercizio2/jobid
=====
```

Spiegazione: Il comando `edg-job-submit` processo il file `RunInfo.jdl` aggiungendovi le parti relative alla configurazione contenute nei file `/opt/edg/etc/edg_wl_ui_cmd_var.conf` e `/opt/edg/etc/egrid/edg_wl_ui.conf`, il file `jdl` così creato, viene trasferendo al RB assieme a tutti i file specificati nella `InputSandbox` e salva il `JOBID` nel file specificato con `-o jobid`.

Contenuto del file `jobid`:

```
cat jobid
###Submitted Job Ids###
https://egrid-5.egrid.it:9000/nJcvLhYqVmENUNN-hFaJeA
```

Tale file può esserri riutilizzato per più sottomissioni, ogni nuovo `JOBID` verrà accodato alla fine del file.

3.6. STATI DI UN JOB IN GRIGLIA

Dal momento che un job viene sottomesso alla griglia passa attraverso una serie di stati, che descrivono il suo attuale stato, che sono:

SUBMITTED	Sottomesso dall'utente ma non processato dal NetworkServer
WAITING	Accettato dal NetworkServer ma non ancora processato dal Workload Manager
READY	Assegnato ad un CE ma non ancora trasferitoJobController
SCHEDULED	In attesa sul sistema di code del CE
RUNNING	In esecuzione su di un WN DONE
DONE	il job è stato eseguito, l'output è pronto per essere ritirato
ABORTED	il job è stato abortito dal WMS (esempio proxy scaduto, job troppo lungo)
CANCELLED	Il job è stato cancellato dal utente
CLEARED	L'utente ha ritirato l'output

Per poter sapere a che punto è il job che abbiamo sottomesso dobbiamo usare il comando `edg-job-status`, come parametro ha bisogno del `JOBID`.

```
edg-job-status https://egrid-5.egrid.it:9000/nJcvLhYqVmENUNN-hFaJeA
```

oppure visto che il `JOBID` è stato salvato nel file `jobid`

```
edg-job-status -i jobid
```

```
*****
```

BOOKKEEPING INFORMATION:

```
Status info for the Job : https://egrid-5.egrid.it:9000/nJcvLhYqVmENUNN-hFaJeA
Current Status:      Scheduled
Status Reason:       Job successfully submitted to Globus
Destination:         egrid-4.egrid.it:2119/jobmanager-lcgpbs-infinite
reached on:          Wed Sep 29 17:10:58 2004
```

L'output ci riporta lo stato attuale del job, ATTENZIONE le data è riferita al fuso orario di Greenwich quindi può esserle una o due ore indietro rispetto all'orario italiano.

3.7. RITIRO DEL RISULTATO DI UN JOB

Quanto il job Raggiunge lo stato Done si può ritirare l'output dalla Sandbox

```
edg-job-status -i jobid
*****
BOOKKEEPING INFORMATION:

Status info for the Job : https://egrid-5.egrid.it:9000/nJcvLhYqVmENUNN-hFaJeA
Current Status:      Done
Exit code            0
Status Reason:      Job terminated successfully
Destination:        egrid-4.egrid.it:2119/jobmanager-lcgpbs-infinite
reached on:         Wed Sep 29 17:12:43 2004
```

Importante: **Exit_code** contiene il codice di uscita dell'eseguibile lanciato in griglia, nel caso che sia diverso da zero vuol dire che si è verificato un errore durante la sua esecuzione, quindi conviene vedere lo stderr per capire cosa è successo.

Per ritirare l'output esiste il comando

```
edg-job-get-output [option] <JOBID>
```

sono disponibili l'opzione -i file e l'opzione -dir directory che permette di ritare il job in una directory di propria scelta al posto di quella di default che è /tmp/jobOutput/.

```
edg-job-get-output -i jobid
*****
JOB GET OUTPUT OUTCOME

Output sandbox files for the job:
- https://egrid-5.egrid.it:9000/nJcvLhYqVmENUNN-hFaJeA
have been successfully retrieved and stored in the directory:
/tmp/jobOutput/aterpin_nJcvLhYqVmENUNN-hFaJeA
```

I file definiti nella OutputSandbox si trovano ora nella directory /tmp/jobOutput/aterpin_nJcvLhYqVmENUNN-hFaJeA

3.8. CANCELLAZIONE DI UN JOB

È possibile cancellare un job usando, il comando edg-job-cancel

```
edg-job-cancel [option] <JOBID>
```

```
edg-job-cancel https://egrid-5.egrid.it:9000/TVaIMrfzZ1DLk8WGN_yJNg
```

```
Are you sure you want to remove specified job(s)? [y/n]n :y
```

```
===== edg-job-cancel Success =====
The cancellation request has been successfully submitted for the following job(s):
```

```
- https://egrid-5.egrid.it:9000/TVaIMrfzZ1DLk8WGN_yJNg
```

```
=====
```

3.9. ESEMPIO DI DEBUG

Nel caso in cui un job sia entrato nello stato Aborted, significa che qualche parte della griglia ha avuto problemi nella sua gestione, per capire dove sono sorti i problemi è disponibile il comando `edg-job-get-logging`.

Alfine di poter analizzare il log prodotti dal comando `edg-get-logging info` è necessario avere una visione complessiva dei passaggi che effettua un job quando viene imesso in griglia. La figura 2 riporta i servizi coinvolti e le loro interazioni in ordine temporale.

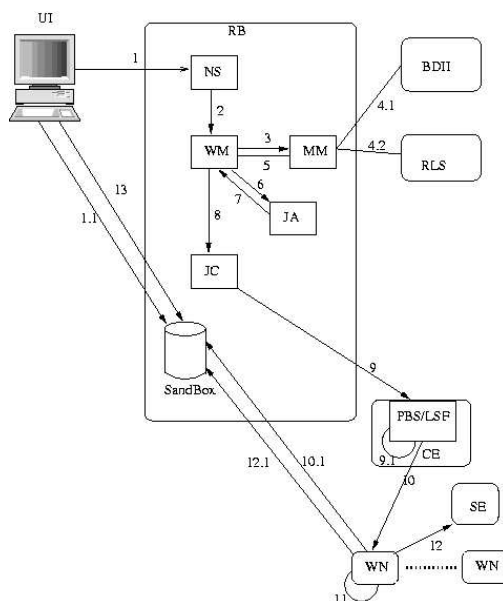


Figura 2: WMS

1. dalla UI l'utente ha lanciato il job tramite `edg-job-submit`
 - 1.1) Avviene lo spostamento dei file definitoo nella InputSandbox sullaSandbox del RB
2. Il NetworkServer contatta il WM
3. WM contata MM se non è stato specificato un CE in maniera esplicita con l'opzione `-r` di o nel file `jdl`. in questo caso passo subito alla fase 6
4. Fase MatchMaking
 - 4.1) Si contatta il BDII per avere informazioni sui CE di Produzione[specifca di configurazione] più ogni altro requirements, nel caso che nessuna risorsa possa soddisfare i requirement passa ad Aborted con il messaggio `StatusReason:Cannotplan:BrokerHelper:nocompatibleresources`
 - 4.2) Si contatta il RLS nel caso di OutputData e InputData per sapere ove sono disponibile i file .
5. Il MM comunica al WM il CE scelto per la sottomissione del job
6. Il WM comunica al JA i dati relativi al CE scelto Il JA prepara gli script di wrapper per la sottomissione del job
7. Il JA a finito la sua fase e ritorna al WM
8. Il WM contataa il JC le informazioni su CE su cui sottomettere il job
9. JC sottomette tramite CondorG il job al CE, costituito dal wrapper pordotto dal JobController
 - 9.1 Sul CE il wrapper viene inserito nella coda dello scheduler in attesa che vi sia un worker node disponibile.
10. Viene messo in esecuzione dal sistema di code sul primo worker node disponibile, il wrapper provvede a scaricare tramite `globus-ur-copy` i dati traseferiti tramite la input Sandox 10.1, e comunicare il passaggio di Stato a running in caso di successo altrimenti comunica

FAILED in caso di Directory non Scrivibile
FAILED se non definito globus Location
FAILED se non riesce a caricare il profile di Globus
FAILED se il file specificato nell' Attributo Executable non è presente e/o eseguibile
La comunicazione di un evento FAILED genero il passaggio di stato ad Aborted

11. Avviene il runnig dell'eseguibile richiesto.

12. alla fine viene trasferito via globus-url-copy i file specificati nella OutputSandbox nel RB, mentre quelli specificati con OutputData verranno trasferiti tramite il comando edg-rm CopyAndRegister
FAILED non riesce a effettuare l'upload sulla Sandbox
i problemi di registrazione sul SE non vengono trappati ma comunicati solamente nel file DSUpload_*JOBID*.out spostato nella SandBox Altrimente da OK e riporta il codice di uscita dell'eseguibile, lo stato passa a done in exit_code

13. L'utente dalla UI ritira l'output, il job passa in stato cleared

3.9.1. ESEMPIO DI DEBUG

Consideriamo il seguente file.jdl SBAGLIATO in cui è presente un errore di battitura

Il nome dell' eseguibile è diverso da quello della InputSandbox Errato.jdl

```
[
  Executable          = "Job_di_analisi_dati01.sh";
  InputSandbox        = "Job_di_analisi_dati01.sh";

  StdOutput           = "stdout";
  StdError             = "stderr";
  OutputSandbox       = {"stdout", "stderr"};
]
```

Vediamo come scoprire questo errore tipografico usando il comando edg-job-get-logging-info

I log risultano tagliati riportando solamente l'ultima parte, visto la loro lunghezza.

```
edg-job-get-logging-info https://egrid-5.egrid.it:9000/4kl4JI3NeL8URElPBCMVUg
```

```
*****
```

```
LOGGING INFORMATION:
```

```
Printing info for the Job : https://egrid-5.egrid.it:9000/4kl4JI3NeL8URElPBCMVUg
```

```
.
```

```
.
```

```
.
```

```
.
```

```
---
```

```
Event: RegJob
```

```
Event: Abort
```

```
- source = LogMonitor
```

```
- timestamp = Thu Sep 30 14:05:04 2004
```

```
*****
```

Otteniamo l'informazione che il LogMonitor ci ha avvisto dello stato di Aborted, ma non la causa, aumentiamo la verbosita.

```
edg-job-get-logging-info -v 1 https://egrid-5.egrid.it:9000/4kl4JI3NeL8URElPBCMVUg
```

LOGGING INFORMATION:

Printing info for the Job : <https://egrid-5.egrid.it:9000/4kl4JI3NeL8URElPBCMVUg>

.
.
.

Event: Abort

- host = egrid-5.egrid.it
- reason = Failure while executing job wrapper
- source = LogMonitor
- src_instance = unique
- timestamp = Thu Sep 30 14:05:04 2004
- user = /C=IT/L=Trieste - Italy/O=EGRID/OU=ICTP/CN=Alessio Terpin

Il campo reason ci dice che il problema si è verificato nell'esecuzione del Job Wrapper quindi, sul WN.

Aumentando il livello di dettaglio

edg-job-get-logging-info -v 2 <https://egrid-5.egrid.it:9000/4kl4JI3NeL8URElPBCMVUg>

LOGGING INFORMATION:

Printing info for the Job : <https://egrid-5.egrid.it:9000/4kl4JI3NeL8URElPBCMVUg>

.
.
.

Event: Done

- exit_code = 1
- host = egrid-5.egrid.it
- level = SYSTEM
- priority = asynchronous
- reason = ./Job_di_analisi_dati01.sh not found or unreadable
- seqcode = UI=000003:NS=0000000003:WM=000004:BH=0000000000:JSS=000003:LM=000007:LRMS=000000
- source = LogMonitor
- src_instance = unique
- status_code = FAILED
- timestamp = Thu Sep 30 14:05:04 2004
- user = /C=IT/L=Trieste - Italy/O=EGRID/OU=ICTP/CN=Alessio Terpin

Event: Abort

- host = egrid-5.egrid.it
- level = SYSTEM
- priority = asynchronous
- reason = Failure while executing job wrapper
- seqcode = UI=000003:NS=0000000003:WM=000004:BH=0000000000:JSS=000003:LM=000008:LRMS=000000
- source = LogMonitor
- src_instance = unique
- timestamp = Thu Sep 30 14:05:04 2004
user = /C=IT/L=Trieste - Italy/O=EGRID/OU=ICTP/CN=Alessio Terpin

Dalla reason sotto l'Event: Done si vede che il motivo è che il Job wrapper non ha trovato l'eseguibile, specificato nel file jdl, infatti Nel file.jdl ci si era confusi uno 0 con O.

3.10. COMANDI EGRID PER LA SOTTOMISSIONE DI JOBS

3.10.1. WRAPPER DI SOTTOMMISSIONE

Per rendere automatica la fase di sottomissione, controllo, ritiro del Output è disponibile un semplice wrapper egrid-submit

```
egrid-submit file.jdl [locale|principale|coda]
```

il primo parametro è il file jdl, il secondo parametro opzionale indica ove deve essere sottomesso il job; può essere

- locale: viene sottomesso in locale
- principale: viene sottomesso sul nodo principale
- coda: viene indicata una specifica coda su di un CE ad esempio egrid-4.egrid.it:2119/jobmanager-lcgpbs-infinite

Nota Bene tale opzioni non effettuano la fase di matchmaking

Il wrapper crea i seguenti file

file.jdl.ANNO_MESE_GIORNO_ORA.MINUTUTO.SECONDO
contenente l'identificatore del job

file.jdl.ANNO_MESE_GIORNO_ORA.MINUTUTO.SECONDO.nohup
contenente il logging dello status del job chiesto ogni 120 secondi.

La directory

file.jdl.ANNO_MESE_GIORNO_ORA.MINUTUTO.SECONDO.OUTPUT
contenete l'output Sandbox del job

In caso di Aborted genera il file file.jdl.ANNO_MESE_GIORNO_ORA.MINUTUTO.SECONDO.Aborted contenete il logging a livello di verbosità -v 2 per l'analisi del problema.

3.10.2. OPERAZIONE EGRID PER LA TRASFORMAZIONE DEI DATI GREZZI DEI CD/DVD, IN FORMATO ASCII

Una delle operazioni specifiche di EGRID è la trasformazione dei dati dei contratti che devono essere salvati in formato tar.gz è l'archivio deve contenere una directory di root in cui sono contenuti i vari dati con qualunque struttura, la trasformazione opera sui dati e salva i risultati in griglia registrandoli sul sistema di replica.

Comando:

```
egrid-transform [-n] <Trasformazione> <lfn> <lfnprefix> [config.file]
```

- Trasformazione la trasformazione che si desidera utilizzare
- lfn Nome logico del file di archivio in formato tar.gz
- lfnprefix Il prefisso della "directory" dei nomi logici di uscita
- -n Provvede a dare in output il file jdl usato per la sottomissione del job per un eventuale sottomissione manuale
- config.file file di configurazione per la trasformazione

Il job così creato verrà sottomesso in griglia e il suo jobid verrà salvato nel file \$HOME/egrid-jobs, per poter venir utilizzato nella fase di monitoring e retrieval dello stdout e stderr via sandbox.

Stdoutput del trasformatore è il file output.txt che contiene lo stdout del programma di trasformazione più la stringa 'egrid-transform-dispatcher.sh: saving file lfn e guid' per ogni file registrato.

Stderror del trasformatore è il file errors.txt e contiene lo standard error del programma di trasformazione file, più lo standard error del trasformatore nella forma 'egrid-transform.-dispatcher.sh errore' in caso che lfnprefix puntava ad una zona in cui l'utente non ha accesso si avrà

```
egrid-transform-dispatcher.sh GlobusURLCopy: the server sent an error response: 553 553 /flatfiles/SE00/egrid-transform-dispatcher.sh
```

Per sapere le trasformazioni installate usiamo il comando `egrid-software`, le trasformazioni installate sono individuate dalla tag `VO-egrid-transform`.

Utilizziamo come esempio la trasformazione di esempio `ls`, che effettua sull'archivio specificato col parametro `lfn` i seguenti comandi

- `ls` salvando il risultato nel file logico `lfnprefix/nomeArchivio-short.txt`
- `ls -l` salvando il risultato nel file logico `lfnprefix/nomeArchivio-medium.txt`
- `ls -Rl` salvando il risultato nel file logico `lfnprefix/nomeArchivio-long.txt`

```
egrid-transform ls lfn:/utenti/aterpin/CD1.tar.gz lfn:/utenti/aterpin/LS
```

Quindi il risultato dell'esecuzione della trasformazione verrà salvato automaticamente in griglia, mentre nella sandbox verrà soltanto riportato la lista dei file che ha salvato, ed eventuali errori.

Alla fine della esecuzione avremo nella standardoutput, la lista dei file prodotti che potranno essere recuperati con i comandi di data management, oppure utilizzati come input per ulteriori elaborazioni

```
$ more output.txt
```

```
egrid-transform-dispatcher.sh: saving file lfn:/utenti/aterpin/LS/CD1-long.txt: guid:87f160e8-1855-11d9-bfb
```

```
egrid-transform-dispatcher.sh: saving file lfn:/utenti/aterpin/LS/CD1-medium.txt: guid:8c047775-1855-11d9-a
```

```
egrid-transform-dispatcher.sh: saving file lfn:/utenti/aterpin/LS/CD1-short.txt: guid:900f0357-1855-11d9-8c
```

A EGRID LIVE UI

Il live CD *EGRID Live UI* contiene tutto il software necessario per accedere alla griglia EGRID, e più precisamente per sottoporre lavori, per copiare, spostare file dal computer locale verso gli storage element e viceversa, e per replicare file tra gli storage element.

EGRID Live UI ha una funzione di installazione con cui può essere permanentemente copiato su un hard-disk, con tutto il sistema operativo Debian GNU/Linux.

La configurazione di questo live CD è pronta per l'uso del testbed di EGRID a Trieste, ed include un certificato di esempio, valido solo per l'uso del testbed. Il certificato di esempio verrà rimosso nelle versioni configurate per la griglia di produzione.

EGRID Live UI è basato sul live CD Knoppix 3.6.

A1. REQUISITI PER IL FUNZIONAMENTO

Il software di griglia LCG-2/EDG è piuttosto esigente e non funzionerà, riportando errori misteriosi, se il sistema non soddisfa ai requisiti che sono qui sotto elencati.

Per aiutare nella diagnosi dei problemi, è fornito con *EGRID Live UI* il programma **egrid-host-self-test** che verifica lo stato del sistema, e riporta suggerimenti su come risolvere eventuali misconfigurazioni.

A1.1. DATA E ORA

l'ora di sistema deve essere esatta!

Per agevolare questo compito, *EGRID Live UI* tenta, all'avvio del sistema, di collegarsi ad un server di ora esatta (NTP), ma diverse condizioni possono decretare il fallimento di questo tentativo. È sempre bene controllare l'ora di sistema prima di cominciare le operazioni di griglia.

A1.2. NOME DNS

Il nome DNS della macchina deve essere completamente qualificato ("fully qualified", cioè deve includere il nome di dominio e non solo il nome dell'host). Inoltre, il nome DNS deve essere correttamente registrato, e devono essere funzionanti sia la risoluzione diretta che quella inversa del nome.

A1.3. FIREWALL

Le istruzioni aggiornate su come configurare eventuali firewall si trovano all'indirizzo <http://www.egrid.it/doc/utenti/firewall-testbed>, di cui riassumiamo qui i punti salienti. Indichiamo "UI" o "macchina UI" il computer su cui gira il software *EGRID Live UI*.

Devono essere possibili *connessioni TCP dalla macchina UI alle destinazioni elencate in tabella 2*

host / ip	porti
egrid-1.egrid.it / 140.105.46.51	10389
egrid-2.egrid.it / 140.105.46.52	2119
egrid-4.egrid.it / 140.105.46.54	2119, 2811
egrid-5.egrid.it / 140.105.46.55	7772, 9000, 9001, 9002
egrid-6.egrid.it / 140.105.46.56	2170
egrid-10.egrid.it / 140.105.46.60	2811
egrid-17.egrid.it / 140.105.46.67	7512, 8080

Tabella 2: Lista dei porti da aprire per accedere al testbed di EGRID

Devono essere possibili *connessioni TCP dalla rete 140.105.46.0/24 destinate alla macchina UI su tutti i porti dell'intervallo 20000--25000* (Globus GridFTP data connection).

Infine, per garantire il funzionamento del servizio di ora esatta, la macchina UI deve poter scambiare pacchetti sul porto UDP 123 (*NTP*, Network Time Protocol) con il resto del mondo.

A1.4. HARDWARE

EGRID Live UI è basata sul Live CD Knoppix, di cui qui riportiamo i requisiti di sistema:

- CPU Intel-compatibile (i486 o superiore);
- 128Mb di RAM (usando il software in modalità solo testo saranno sufficienti 20 MB di RAM);
- lettore CD-ROM/DVD di boot (IDE/ATAPI, Firewire, USB o SCSI);
- una scheda video SVGA-compatibile;
- un mouse seriale standard, o PS/2, o USB IMPS/2-compatibile;
- una scheda di rete compatibile con Linux

A2. USARE *EGRID Live UI*

A2.1. AVVIARE IL LIVE CD

Per utilizzare *EGRID Live CD* è sufficiente avviare il computer dal CD-ROM di *EGRID Live CD*; dopo qualche istante apparirà una schermata grafica ed un prompt boot: in basso a sinistra. Premere il tasto 'Invio' per avviare *EGRID Live UI*; il sistema si avvierà comunque dopo aver atteso 30 secondi.

Opzioni alternative di avvio Per avviare il sistema in modo testo (risparmia memoria), si risponda **textmode** al prompt boot:.

Le altre opzioni di avvio sono elencate nel file /cdrom/KNOPPIX/knoppix-cheatcodes.txt.

A2.2. IL SISTEMA GRAFICO

EGRID Live UI si avvierà con il sistema grafico KDE; l'uso del sistema grafico è solo una comodità per l'utente - tutti i programmi di griglia sono comunque funzionanti anche se *EGRID Live UI* viene avviato in modo solo testo.

Il programma **kdehelpcenter** permette di accedere alla documentazione delle applicazioni fornite con KDE.

All'avvio sul desktop grafico si apriranno due finestre, quella che contiene questo documento, e quella del programma di controllo della configurazione **egrid-host-self-test** - quest'ultima avviserà di eventuali problemi, o viceversa informerà che nulla è di ostacolo al funzionamento della griglia.

Configurazione della tastiera Il sistema *EGRID Live UI* è configurato per partire automaticamente con la tastiera italiana; per cambiare rapidamente questa impostazione con la tastiera US/Internazionale, si può fare clic con il tasto destro sulla piccola icona della bandiera italiana in basso a destra nel desktop.

A2.3. I COMANDI DI GRIGLIA

Tutti i comandi user interface di griglia di EGRID/LCG-2/EDG sono installati in *EGRID Live UI*. I comandi si usano da linea di comando, come di solito si fa; per alcune operazioni è anche disponibile un'interfaccia grafica.

Il menu identificato dall'icona permette di accedere ai comandi grafici specifici di *EGRID Live UI*.

Accesso a EGRID Questa voce di menu attiva il programma **egrid-proxy-ui**, che permette di attivare un certificato proxy, esaminarne i dettagli, rinnovarlo oppure distruggerlo, sempre usando un'interfaccia grafica. Di fatto, **egrid-proxy-ui** svolge le stesse funzioni di **grid-proxy-init**, **grid-proxy-info**, **grid-proxy-destroy**.

Autocontrollo di configurazione Questa voce di menu attiva il programma **egrid-host-self-test**, che esegue alcuni test di configurazione per controllare che la macchina UI soddisfi tutte le condizioni elencate nella sezione Requisiti. Il programma **egrid-host-self-test** presenta un'interfaccia grafica non-interattiva; al termine dell'esecuzione presenterà un messaggio "Tutti i test completati con successo." (figura 6), ovvero un messaggio di errore che tenta di suggerire un possibile rimedio (figura 7). Si tenga presente, però, che la

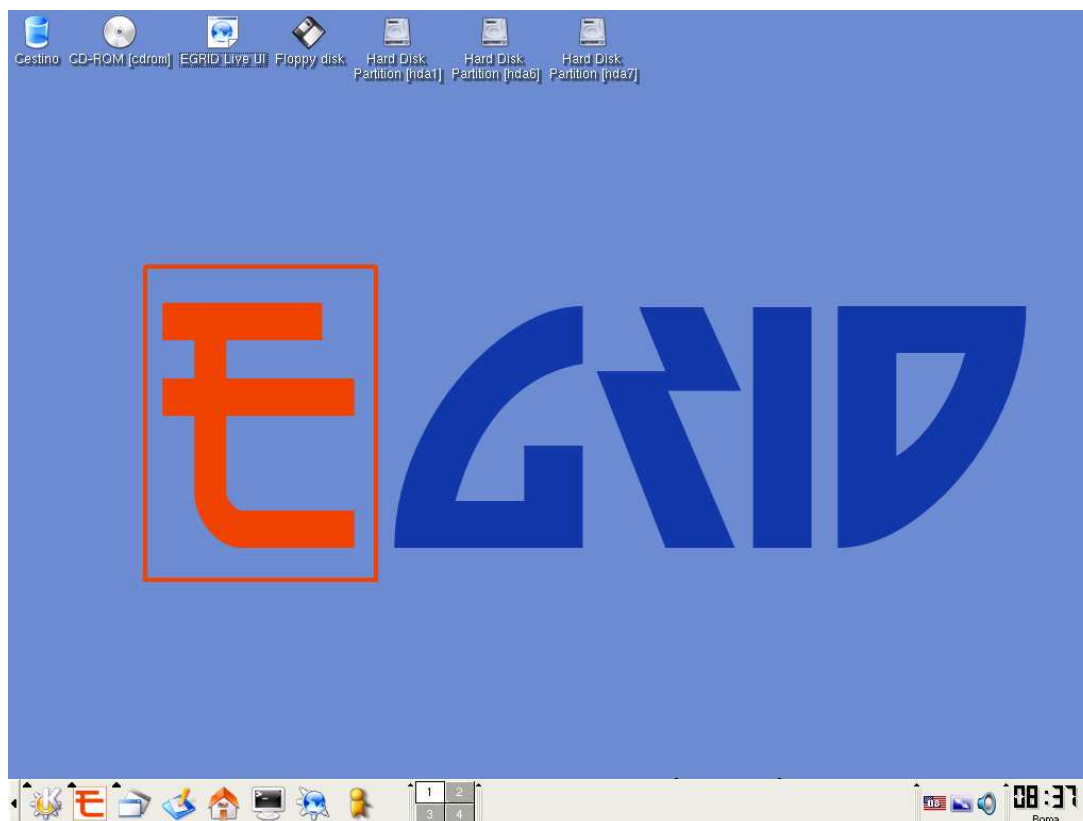


Figura 3: L'aspetto del desktop di *EGRID Live UI*



Figura 4: Cambiare l'impostazione della tastiera.



Figura 5: Il menu dei comandi EGRID

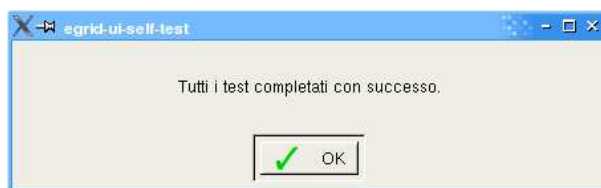


Figura 6: Il messaggio visibile su un computer correttamente configurato.

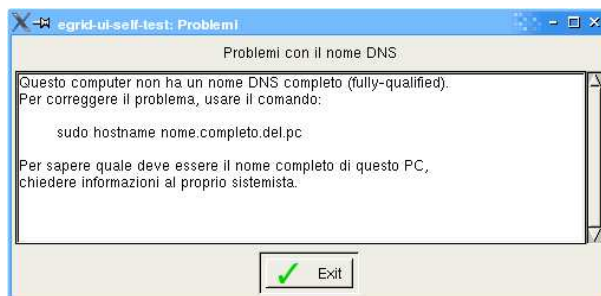


Figura 7: Un esempio di messaggio di errore.

diagnostica del programma è piuttosto elementare e spesso i problemi possono avere cause remote e non apparenti; sarà perciò spesso necessario comunque l'intervento di un sistemista.

Autocontrollo di funzionamento griglia Questa voce di menu attiva il programma **egrid-ui-self-test**, che esegue alcuni test basilari di funzionalità della griglia. Il programma **egrid-ui-self-test** presenta un'interfaccia grafica non-interattiva; al termine dell'esecuzione presenterà un messaggio "Tutti i test completati con successo.", ovvero un messaggio di errore che riporta l'output e gli errori dell'ultimo comando eseguito. tenta di suggerire un possibile rimedio; nessun tentativo viene fatto di diagnosticare le cause degli eventuali problemi.

Crea una home permanente Questa voce di menu richiama il programma **knoppix-mkpersistenthome** che permette di creare uno spazio dati permanente sul disco rigido, su una memory card USB o su un floppy, per salvare i propri dati tra un avvio del Live CD e l'altro. l'intero home filesystem verrà creato in un file **knoppix.img** sul medium scelto.

EDG JDL Editor GUI

EDG Job Monitor GUI

EDG Job Submitter GUI Queste voci di menu richiamano le interfacce grafiche scritte da EDG per sottomettere job in griglia (EDG Job Submitter), per monitorare lo stato di Job sottomessi (EDG Job Monitor), o per scrivere file di controllo JDL (EDG JDL Editor). Il programma EDG Job Submitter è in grado di controllare gli altri due. Per i dettagli, si rinvia alla documentazione prodotta da EDG.

Installa sul disco rigido Questa voce di menu richiama il programma **knoppix-installer**, che permette di installare il sistema operativo Debian GNU/Linux con **EGRID Live UI** sul disco rigido. La sezione Installazione riporta maggiori dettagli.

A3. INSTALLAZIONE

È possibile installare **EGRID Live UI** sul disco rigido: si elimina così la necessità di dover ogni volta utilizzare il CD-ROM, e si rendono permanenti i dati e le modifiche apportate. Con la funzione di installazione si ottiene un computer con installato Debian GNU/Linux ed anche tutto il software di accesso alla griglia scritto da EGRID, LCG, EDG e Globus.

È possibile usare lo stesso Live CD per installare più di un computer; ad ogni installazione verranno chiesti i dati che variano da un computer all'altro.

Attenzione! l'installazione richiede una partizione dedicata sull'hard disk, capiente almeno 3Gb; questa partizione verrà reinizializzata e tutti i dati presenti su di essa andranno *persi* - perciò è bene assicurarsi di averne una copia di riserva prima di procedere.

A3.1. PRIMA DI COMINCIARE

Prima di cominciare, è necessario raccogliere dal proprio Amministratore di Rete le seguenti informazioni sul computer che si va ad installare, perché saranno richieste dal programma di installazione:

- se il computer potrà usare il servizio DHCP per ottenere automaticamente la configurazione di rete;
- il nome DNS del computer;
- l'indirizzo IP assegnato;
- la maschera di sottorete (netmask);
- default gateway;
- il server di risoluzione dei nomi DNS;
- se sulla propria rete è presente un server NTP (*Network Time Protocol*), e, nel caso, quale sia il suo nome DNS o il suo indirizzo IP;

A3.2. PROCEDURA DI INSTALLAZIONE

Avviare il computer dal CD di *EGRID Live UI*. Lanciare il programma **knoppix-installer**; è disponibile una funzione di aiuto in linea, che può guidare attraverso i vari passi.

Alcuni punti a cui prestare particolare attenzione:

- l'installazione del sistema richiede una partizione dedicata sull'hard disk, capiente almeno 3Gb; questa partizione verrà reinizializzata e tutti i dati presenti su di essa andranno *persi* - perciò è bene assicurarsi di averne una copia di riserva prima di procedere.
- nel riquadro che chiede di scegliere il tipo di installazione da effettuare ("Choose system type"), selezionare Beginner: nonostante il nome, è il tipo di installazione più adatto in generale.
- nel riquadro che richiede l'immissione del nome della macchina, indicare il *nome DNS pienamente qualificato*, cioè completo del nome di domino, e non limitato al nome host (p.es., *egrid-16.egrid.it* invece di *egrid-16*).
- Verrà richiesto il nome di *un solo* utente da creare; questo utente potrà eseguire il comando **sudo** *senza password*, cioè potrà eseguire ogni tipo di comandi amministrativi del sistema.

A3.3. POST-INSTALLAZIONE

Dopo un'installazione eseguita correttamente, non dovrebbe esserci alcun bisogno di ulteriore configurazione manuale.

Riportiamo comunque qui alcuni dei parametri che potrebbe essere necessario cambiare; per favore, segnalateci ogni procedura utile che ritenete sia stata omessa.

Impostazione del server di ora esatta I server di ora esatta sono preimpostati a *pool.ntp.org* e *timehost.ictp.trieste.it*; per cambiare questa impostazione si agisce su due file:

/etc/default/ntpdate Controlla il programma *ntpdate*, che imposta l'ora esatta *all'avvio* del sistema.

/etc/ntp.conf Controlla il demone *ntpd* che mantiene l'ora esatta durante il funzionamento del sistema.

Impostazione dei parametri di rete Tutti i parametri di rete si possono impostare attraverso una interfaccia utente amichevole con il programma `netcardconfig`, che provvederà a modificare i rilevanti file di configurazione del sistema.

Cambio del nome dell'host Il nome dell'host ed il suo indirizzo IP primario sono impostati attraverso i due file:

`/etc/hostname` Questo file contiene il nome DNS completo.

`/etc/hosts` Consultare la pagina `man hosts(5)` per il formato ed il significato di questo file.

A4. DOWNLOAD

L'ultima versione di *EGRID Live UI* è disponibile all'URL <http://www.egrid.it/download/software/ui/>.

A5. LINK & ULTERIORI INFORMAZIONI

Documentazione di LCG-2 <http://grid-deployment.web.cern.ch/grid-deployment/cgi-bin/index.cgi?var=eis/docs>

Knoppix <http://www.knopper.net/knoppix>

A6. CONTATTI

Lo staff tecnico di EGRID rimane a disposizione per ogni ulteriore chiarimento, suggerimento o modifica; contattateci pure all'indirizzo e-mail `<staff@egrid.it>`.

ACKNOWLEDGEMENTS

Si ringraziano Massimo Sgaravatto, Enrico Ferro e Marco Verlatto per l'aiuto fornito nella installazione e nella realizzazione di questa prima release del progetto Egrid.

Questo lavoro è finanziato dal progetto MIUR E-grid.