

IGI Portal: portale web di accesso a risorse Grid e Cloud per le comunità scientifiche

Marco Bencivenni^{1,2}, Diego Michelotto¹, Andrea Ceccanti¹,
Andrea Cristofori¹, Enrico Fattibene¹, Giuseppe Misurelli¹,
Riccardo Brunetti³, Paolo Veronesi¹



¹INFN-CNAF, ²Università di Ferrara, ³INFN Torino

Abstract. Nell'ambito del progetto europeo EGI, la NGI Italiana ha sviluppato un portale web general purpose al fine di dare un accesso facilitato alle risorse Grid e Cloud. Un utente che si affaccia per la prima volta all'ambiente Grid si trova di fronte ad alcune barriere che rischiano di allontanarlo dalla Grid ancora prima di utilizzarla. Tra le varie difficoltà spiccano: la complessità di ottenere e gestire le credenziali Grid (Certificato e appartenenza a una *Virtual Organization*) e il tempo di apprendimento legato alla complessità dei comandi e alla possibilità di errore. Al fine di dare all'utente la massima facilità di utilizzo delle risorse è stata creata una sofisticata architettura che vede il portale connesso a diversi elementi, in modo tale che sia facilmente possibile autenticarsi alla Grid, sottomettere job/workflow/applicazioni a differenti *Middleware* e Cloud, gestire dati in Grid e Cloud. Affinché un utente possa facilmente autenticarsi al portale l'autenticazione è demandata a una federazione basata su SAML2 (IDEM, eduGAIN). È stato anche implementato un servizio per la gestione dei dati in Grid attraverso il quale gli utenti possono compiere operazioni in piena autonomia e con estrema facilità. Dal portale gli utenti autorizzati possono fare upload e download di file sulla Grid (*data transfer*), vedere quali file sono in Grid (*namespace browsing*) e gestire file sui quali hanno i privilegi (*file management*). Al fine di muovere grandi quantità di dati è stata creata una infrastruttura di storage connessa al portale che permette di eseguire i trasferimenti e la gestione dei file dall'interfaccia web, trasferendo i file esternamente al portale, in modo da non aggravarlo di questo consumo di risorse.

1. Introduzione

Da ormai qualche anno, l'infrastruttura Grid, nata allo scopo di servire la fisica delle alte energie, ha ampliato il proprio bacino di utenza fornendo potenza di calcolo e spazio storage a utenti diversificati per tipologia, dagli esperti informatici fino ai neofiti, e per ambito scientifico: biologia, astrofisica, ecc.

Oltre a questi nuovi scenari sono emerse anche esigenze da parte degli utenti che sempre più numerosi si affacciano alla tecnologia Cloud sia per elaborare dati sia per immagazzinarli. Mentre gli utenti LHC hanno avuto modo e tempo di accrescere la propria esperienza e ormai si muovono in modo collaudato tra l'utilizzo di certificati X.509 e la linea di comando, i nuovi utenti vedono questi meccanismi come barriere spesso insormontabili, tali da far abbandona-

re la tecnologia Grid ancora prima di provare a utilizzarla.

È proprio per fare fronte a queste difficoltà che il portale web IGI [1] nasce, cercando di fornire agli utenti dei potenti mezzi di calcolo ma con una facilità di utilizzo che non scoraggi coloro che d'informatica non sono esperti. Il portale è basato su Liferay [2] che, grazie al concetto di *portlet* [3] garantisce un'alta modularità e granularità: l'aggiunta di nuovi servizi corrisponde all'aggiunta di nuovi moduli che possono essere visibili a tutti o solo a specifici gruppi di utenti. Il portale dispone di moduli per la sottomissione di *job* o applicazioni, per la gestione dei dati, per l'autenticazione e altri che vedremo in seguito, ma può anche essere facilmente integrato con nuovi moduli grazie all'adozione dello standard JSR-286.

Il portale nasce come servizio che offre agli utenti una semplice interfaccia web con un'ampia di funzionalità. In modo trasparente all'utilizzatore finale il portale è interfacciato ad altri servizi esistenti e collaudati di tipo Grid e Cloud: diversi provider IaaS per la gestione di macchine virtuali on-demand, un server Dirac [4] per la sottomissione di Job, WS-Pgrade [5] per la sottomissione di complessi *workflow*.

2. Autenticazione e Autorizzazione

Nel nostro modello, l'autenticazione è demandata a una federazione di *Identity Provider*, in modo che ciascun utente possa utilizzare le credenziali che quotidianamente usa per usufruire dei servizi del proprio istituto. La federazione abilitata sul portale è eduGAIN [6] che comprende anche tutti gli istituti italiani che fanno parte di IDEM [7].

L'autorizzazione è invece basata sul possesso di un certificato X.509 e sull'appartenenza a una *Virtual Organization* (VO). Le credenziali Grid sono gestite con l'utilizzo di un MyProxy server, dove sono conservate in modo sicuro.

Un utente che attraverso la registrazione prova il possesso di un certificato X.509 e l'appartenenza a una VO può automaticamente e senza limiti iniziare a utilizzare le risorse offerte da essa offerte.

Il portale è stato disegnato anche per essere in futuro integrato con una CA online in grado di sfruttare gli attributi rilasciati dagli Identity Provider per generare *on demand* un certificato personale X.509 in modo trasparente all'utente. In questo scenario, l'utente non avrebbe l'onere di gestire il proprio certificato, ma è comunque in grado di utilizzare le risorse Grid perché in possesso delle credenziali necessarie.

3. Grid Computing

La Grid nasce per permettere agli esperimenti della fisica delle alte energie di eseguire grandi moli di calcolo sfruttando la distribuzione delle risorse. Altre discipline, come ad esempio la biologia, che effettua calcoli per l'annotazione proteica o sequenziamento del genoma, hanno

necessità di potenza di calcolo analoghe o anche maggiori. Queste comunità in genere utilizzano degli applicativi specifici per le proprie applicazioni.

In generale possiamo distinguere tra categorie di sottomissione: job semplice, workflow, applicazione; per ciascuna di esse il portale ha una sezione specifica.

2.1 Job Semplice

in questo caso, l'utente costruisce in piena libertà e autonomia il proprio JDL (*Job Description Language*) che sarà utilizzato per la sottomissione, impostando tutti i campi che ritiene utili o necessari.

2.2 Workflow

Si tratta di un insieme di nodi concatenati tra loro per permettere all'utente di elaborare calcoli più complessi, che passino da uno step all'altro in modo automatico. I nodi possono essere job semplici in cascata - ad esempio l'output di uno o più job può essere l'input di un altro - ma anche interazioni con database o altri servizi esterni. In questo caso l'utente può costruire graficamente il workflow e per ciascun nodo impostare tutti i parametri necessari.

2.3 Applicazione

Come accennato sopra, alcune comunità possono avere esigenze molto specifiche e quindi usare applicativi *ad hoc*. Specialmente in queste comunità possiamo trovare non esperti di informatica, ad esempio biologi o astrofisici, che hanno necessità di effettuare complessi calcoli che i server dei loro laboratori possono non essere in grado di supportare. La Grid può quindi dare loro la possibilità di sopperire alla mancanza di risorse e un'interfaccia di alto livello, quale un portale web, può permettere l'utilizzo di queste anche a non esperti informatici: questo binomio può garantire in breve tempo la possibilità ad ogni ambito scientifico di utilizzare l'infrastruttura Grid. Per ciascuna di queste applicazioni viene prima portata a termine un'analisi di portabilità in Grid: attraverso test pratici si verifica se il software può essere eseguito sui *Worker Node*. Una volta che il processo di porting è concluso, ci si concentra sulle esigenze dell'utente,



Fig 1 - Esempio di interfaccia *ad hoc* per l'invio di applicazioni

in modo da creare un'interfaccia web semplice (Fig. 1) in cui possa eseguire i propri calcoli e recuperare gli output.

In questa fase quindi si cerca di capire quali sono gli input e i parametri da impostare, se esiste l'esigenza di controllare in maniera interattiva l'andamento dell'elaborazione, quali sono gli output da recuperare e ogni altra cosa che l'applicazione specifica richiede. In questo modo si cerca di fornire una sorta d'intelligenza superiore a quella che si avrebbe con il semplice invio di job o *workflow*.

4. Cloud

Come si osserva da qualche anno a questa parte, la Cloud è un ambiente di calcolo sempre più utilizzato che, grazie a importanti *vendor* mondiali, sta conquistando una larga fetta di utenti che richiedono potenza di elaborazione. Esistono anche realtà *Open Source* sviluppate in

diversi istituti nell'ambito del progetto europeo o EGI [8]: ad esempio la soluzione sviluppata all'INFN WNoDeS [9], ma anche altre che si basano su OpenNebula (ad esempio quella sviluppata a FZJ) oppure su OpenStack (come quella sviluppata a CESNET).

Il portale, sfruttando lo standard OCCL, mette a disposizione una interfaccia (Fig. 2) comune a tutte le soluzioni supportate, attraverso la quale gli utenti possono usufruire dei servizi Cloud che queste forniscono.

In particolare, un utente può creare e avviare macchine virtuali scegliendo l'immagine da una lista che varia dinamicamente in base alle informazioni recuperate da un *marketplace* comune alle varie soluzioni Cloud. Per ciascuna immagine l'utente può scegliere tra quattro tipi di macchine: *small*, *medium*, *large* o *extra large*, che differiscono tra loro rispetto a numero di core, spazio disco e RAM; può inoltre de-



Fig 2 - Esempio di interfaccia per istanziare macchine virtuali

cidere se caricare la propria coppia di chiavi SSH per fare *login* senza *password* sulla macchina stessa o lasciare il compito della generazione delle chiavi al sistema e poi scaricarle una volta che la macchina è stata creata. L'utente, infine, ha a disposizione un terminale web attraverso il quale può compiere tutte le operazioni come se avesse a disposizione la *shell* localmente.

5. Data Management

Poiché ciascuna operazione di elaborazione richiede dati in input e produce dati in output, l'aspetto della gestione dei dati è non meno importante del computing. Al fine di evitare che il portale diventi un collo di bottiglia per il trasferimento dei dati, è stata disegnata un'architettura (Fig. 3) in cui i file sono trasferiti attraverso degli *Storage Element* (SE) esterni al portale stesso, ma sotto il suo dominio.

Dal punto di vista logico, i componenti principali sono due: un servizio *Data Mover* e un *cluster* di SE (in figura indicato come *SEs Portal*). Per le operazioni di upload e download gli SEs Portal agiscono come memoria cache: i file vengono conservati solo per lo stretto tempo necessario ad essere trasferiti sugli SE Grid (fase di upload) oppure recuperati dagli SE Grid (fase di download). Questa operazione è necessaria poiché al momento è possibile comunicare con gli SE Grid solo attraverso opportuni client. Il *Data Mover* invece controlla e gestisce ogni passo del trasferimento, in particolare per le operazioni sui file logici e le operazioni di replica interagisce direttamente con l'infrastruttura Grid.

Dal punto di vista implementativo il servizio *Data Mover* è integrato in un tool utilizzato anche per la visualizzazione di file, *AjaXplorer* [10], mentre il cluster di SE è realizzato utilizzando *SToRM* [11]. Con l'architettura implementata è possibile raggiungere un completo disaccoppiamento tra l'interfaccia web (Portale), il gestore dei trasferimenti (*Data Mover*) e lo

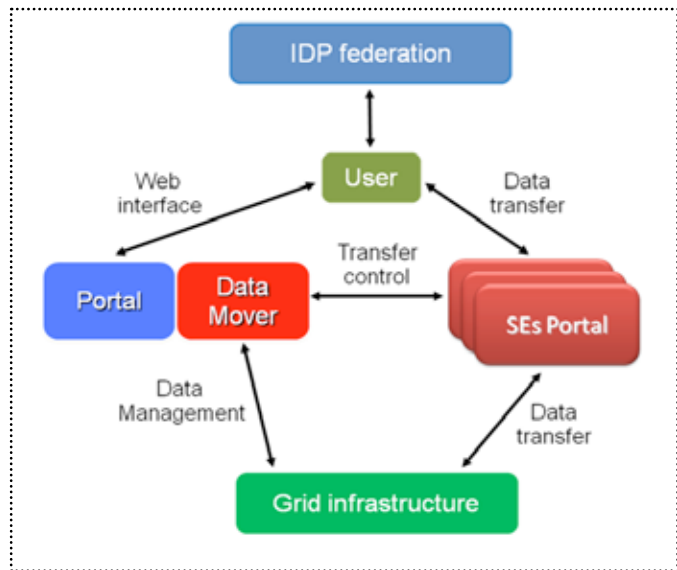


Fig 3 - Architettura del Data Management

spazio fisico interessato nel movimento dei dati (SEs Portal): in questo modo è anche possibile ottenere un certo grado di scalabilità, in quanto il cluster di SE può essere ampliato e anche distribuito geograficamente in modo da assicurare all'utente un trasferimento veloce.

Distinguiamo tre fasi principali quando si parla di dati: gestione, upload e download che ora andremo a esaminare in dettaglio.

5.1 Gestione Dati

Ciascun utente autenticato sul portale può accedere alla sezione Storage e navigare il contenuto del *Logical File Catalog* in base alla VO che sta utilizzando. Per ciascun file o directory sono fornite informazioni aggiuntive quali: ultima data di modifica, dimensione e livello di condivisione (con tutti i membri della VO, con solo alcuni utenti del portale oppure con nessuno). Sui file o directory sono possibili diverse azioni in grado di agire a livello logico, cioè provocare variazioni sul LFC (come la creazione di nuove cartelle, il cambiamento di nome di un file, la condivisione di file e tante altre), a livello fisico, cioè provocare variazioni sugli SE (come la replica o *download* di file dalla Grid) o su entrambi (come la cancellazione o upload di file su SE Grid).

5.2 Upload

Con *upload* si intende il passaggio di dati dall'u-

tente a un SE Grid via web. A tale scopo vengono utilizzati due tool: Plupload [12] per browser di tipo Internet Explorer e jQuery File Upload [13] per gli altri browser. jQuery File Upload è stato scelto perché permette l'upload di file di grandi dimensione, nell'ordine della decina di GB, utilizzando due meccanismi: il *chunking* e, per i browser che supportano la funzione *xmlhttprequest* di HTML5, il *Resumable Upload*, che permette di recuperare un trasferimento interrotto a partire dallo stato prima del fallimento. Plupload è stato scelto perché supporta SilverLight [14] che permette il meccanismo del *chunking* anche per le vecchie versioni di Internet Explorer. Il sistema è in grado di riconoscere il browser dell'utente e automaticamente scegliere quale tool utilizzare. Nella fase di upload il file è copiato sugli SE Grid e registrato nel *Logical File Catalog*, in particolare viene eseguito il meccanismo prima citato in cui il file transita momentaneamente sugli SEs Portal prima di essere definitivamente trasferito in Grid. Al momento dell'upload, l'utente può scegliere lo SE di destinazione. In base alla VO in uso e alle dimensioni del file, il sistema propone all'utente una serie di possibili destinazioni attraverso un menu a tendina.

Se l'utente sceglie uno specifico SE, allora il sistema proverà a trasferire il file sullo SE scel-

to. Nel caso questo fallisca, il sistema tenterà di trasferirlo sugli altri in modo randomico, fino a quando il trasferimento non si conclude con successo. Durante l'operazione di upload l'utente può controllare la velocità di trasferimento e la percentuale di completamento (Fig. 4).

Alla conclusione del trasferimento l'utente è informato sullo stato finale. Ci sono tre possibilità:

- *Transfer ok*: il file è stato correttamente trasferito in Grid sullo SE scelto.
- *Transfer ok but not on the selected SE*: il file è stato correttamente trasferito in Grid ma non sullo SE scelto.
- *Transfer failed*: non è stato possibile trasferire il file in Grid; è inoltre possibile avere maggiori dettagli sulle cause di fallimento su ciascun SE provato.

5.3 Download

Un'altra importante caratteristica è la possibilità di scaricare file dalla Grid a una destinazione locale o remota. In base alla dimensione dei file da recuperare il sistema dà la possibilità all'utente di scaricarli sul proprio PC oppure, se il file supera una certa soglia, di indicare un server di destinazione che supporti il protocollo FTP o SFTP. Il download del file per l'utente è completamente trasparente, ma anche questo consiste in due fasi: il file viene prima copiato sul-

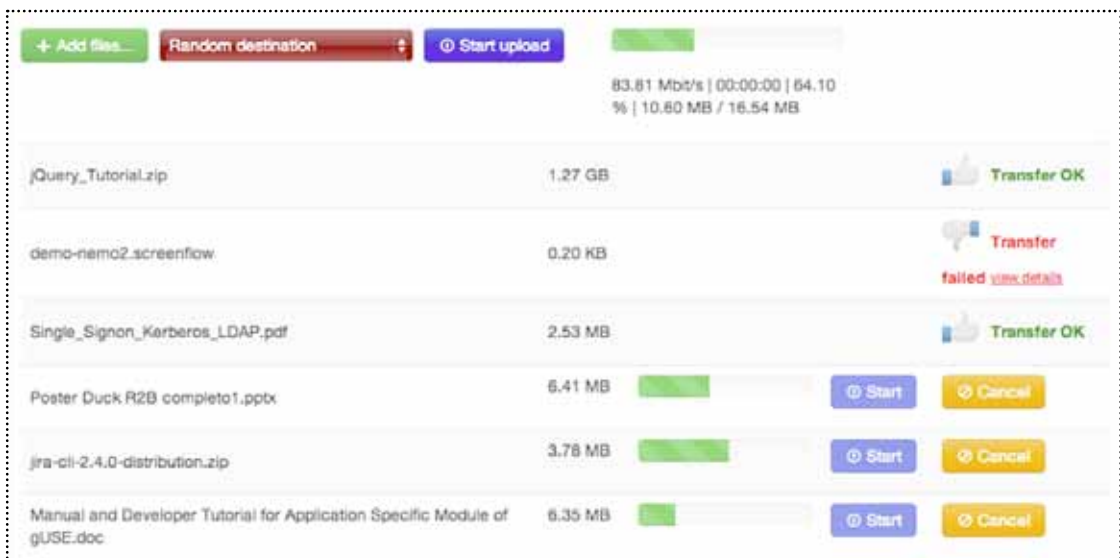


Fig 4 - Interfaccia di Upload

lo SEs Portal e quindi trasferito all'utente; anche in questo caso SEs Portal agisce come cache e alla fine del trasferimento il file viene eliminato. È anche possibile scaricare una lista di file in un'unica volta; in questo caso, se la dimensione totale dei file da scaricare è sotto la soglia, l'utente riceverà un unico archivio di tipo *tar*; se invece è sopra la soglia, man mano che i file sono recuperati dalla Grid vengono immediatamente copiati sulla destinazione indicata.

Riferimenti bibliografici

- [1] IGI - <http://www.italiangrid.it>
- [2] Liferay - <http://www.liferay.com>
- [3] Portlet - <http://jcp.org/en/jsr/detail?id=286>
- [4] DIRAC - <http://diracgrid.org>
- [5] WS-Pgrade - <https://guse.sztaki.hu/liferay-portal-6.0.5>
- [6] eduGAIN - <http://www.geant.net/service/edugain/pages/home.aspx>
- [7] IDEM - <https://www.idem.garr.it>
- [8] EGI - <http://www.egi.eu>
- [9] WNoDeS - <http://web2.infn.it/wnodes/index.php/wnodes>
- [10] AjaxPlover - <http://ajaxplorer.info>
- [11] STORM - <http://storm.forge.cnaf.infn.it>
- [12] jQuery File Upload - <http://blueimp.github.com/jQuery-File-Upload>
- [13] Plupload - <http://www.plupload.com/>
- [14] SilverLight - <http://www.microsoft.com/silverlight>



Marco Bencivenni

marco.bencivenni@cnaf.infn.it

Laureato in Ingegneria delle Telecomunicazioni presso l'Università di Bologna. Lavora presso il CNAF, partecipa a progetti IGI e EGI. Si occupa dello studio e dell'implementazione di servizi al fine di utilizzare le risorse Grid e Cloud in modo facile e sicuro attraverso la realizzazione di un portale web.