Consortium GARR

Workshop GARR CSD
Selected papers

WORKSHOP GARR

CALCOLO E STORAGE DISTRIBUITO

ROMA, 29-30 NOVEMBRE 2012

Consortium GARR

Workshop GARR CSD
Selected papers

WORKSHOP GARR

CALCOLO E STORAGE DISTRIBUITO

ROMA, 29-30 NOVEMBRE 2012

ISBN 978-88-905077-4-8

Editore: Associazione Consortium GARR
Via dei Tizii, 6, 00185 Roma, Italia
http://www.garr.it
Tutti i diritti riservati.

Curatori editoriali: Giuseppe Attardi, Marta Mieli, Federica Tanlongo, Vincenzo Vagnoni
Progetto grafico: Carlo Volpe
Impaginazione: Marta Mieli, Federica Tanlongo, Carlo Volpe

Prima stampa: Giugno 2013
Numero di copie: 1500
Stampa: Tipografia Graffietti Stampati snc
S.S. Umbro Casentinese Km 4.500, 00127 Montefiascone (Viterbo)

Tutti i materiali relativi al Workshop GARR Calcolo e Storage Distribuito (CSD) sono disponibili all'indirizzo: http://www.garr.it/ws-garr-csd

Indice

Introduzione 5
G. Attardi, V. Vagnoni

Progetto DECIDE: un esempio di infrastruttura al servizio della comunità biomedica 7
V. Ardizzone

IGI Portal: portale web di accesso a risorse Grid e Cloud per le comunità scientifiche..... 14
M. Bencivenni, D. Michelotto, A. Ceccanti, A. Cristofori, E. Fattibene, G. Misurelli, R. Brunetti, P. Veronesi

Authentication e authorization federate nelle Cloud: estensioni a Shibboleth per l'applicazione in contesti di Cloud Computing..... 20
A. Biancini, L. Prete, S. Vocella

GaaS: Grid personalizzate per il calcolo su Cloud..... 27
V. Boccia, G.B. Barone, R. Bifulco, D. Bottalico, L. Carracciuolo, R. Canonico

Lo Science Gateway del progetto agINFRA per l'accesso a una data infrastructure per le Scienze Agrarie..... 33
R. Bruno, G. Allegri, G. Andronico, R. Barbera, F. Bitelli, A. Budano, A. Calanducci, E. A. C. Costantini, M. Fargetta, A. Fornaia, G. L'Abate, S. Monforte, A. Puliafito, R. Ricceri, F. Ruggieri, D. Saitta, M. Villari

Software-Defined Networking: Esperienze OpenFlow e l'interesse per Cloud..... 40
M. Campanella, F. Farina, L. Prete, A. Biancini

Octopus: una Cloud self-service di macchine virtuali..... 46
A. Cisternino, M. Davini, M. Mura

Grandi infrastrutture di storage per calcolo ad elevato throughput e Cloud..... 50
M. Di Benedetto, A. Cavalli, L. dell'Agnello, M. Favaro, D. Gregori, M. Pezzi, A. Prosperini, P.P Ricci, E. Ronchieri, V. Sapunenko, V. Vagnoni, V. Venturi, G. Zizzi

Cloud Computing in ENEA-GRID: Macchine Virtuali, Roaming Profile e Online Storage..... 57
G. Ponti, A. Rocchi, A. Colavincenzo, G. Giannini, A. Secco, G. Bracco, S. Migliori

Distributed Open Cloud Computing, Storage e Network con WNoDeS: Esperienza ed Evoluzione..... 63
D. Andreotti, M. Caberletti, V. Ciaschini, G. Dalla Torre, A. Italiano, E. Ronchieri, D. Salomoni

Sull'interoperabilità tra risorse locali, Grid e Cloud per la realizzazione di un'infrastruttura di calcolo distribuito in Italia.....69
D. Scardaci, G. Andronico, R. Barbera, R. Bruno, M. Fargetta, A. Fornaia, G. La Rocca, S. Monforte, R. Ricceri, R. Rotondo, D. Saitta

Realizzazione di un'infrastruttura Cloud pilota basata su OpenStack 76
L. Fanò Illic, E. Fattibene, M. Manzali, H. Riahi, D. Salomoni, A. Valentini, P. Veronesi, V. Venturi

Prototipo per un servizio di Cloud Storage federato per il mondo accademico e della ricerca..... 84
S. Vocella, A. Biancini, C. Valli, M. Reale, F. Farina

Introduzione

Giuseppe Attardi, Vincenzo Vagnoni

Chair del Comitato di Programma del Workshop GARR CSD 2012

Le infrastrutture distribuite di calcolo e storage, siano esse griglie computazionali (Grid) o Cloud, sono oggi strumenti indispensabili nell'ambito di un sistema evoluto di ricerca. Tutti i grandi Paesi al mondo si sono già dotati, o stanno dotandosi, di sistemi in grado di soddisfare le crescenti richieste di calcolo e storage poste dalla ricerca di frontiera in tutti i campi.

Il calcolo di tipo Grid ha efficacemente soddisfatto nell'arco dell'ultimo decennio le necessità di talune comunità, come i fisici del CERN e dell'INFN, dotate al loro interno di elevate competenze nel settore ICT e caratterizzate da una struttura di grande scala. Nel contempo, tuttavia, questo modello ha non poco faticato a penetrare in molti altri ambiti. Di fatto, le comunità di ricerca più piccole, per arrivare sino ai singoli gruppi o ai singoli ricercatori, che non hanno potuto permettersi investimenti in formazione, personale e mezzi, sono rimaste al di fuori del circuito, e hanno continuato a lavorare con strumenti tradizionali senza poter avere accesso alla capacità di eseguire complicati calcoli in tempi brevi e memorizzare ingenti quantità di dati. Uno dei limiti principali della Grid risiede infatti nella complessità di utilizzo da parte degli utenti, che ha portato alla necessità di progettare nuove soluzioni.

Col termine di Cloud si identificano oggi una serie di soluzioni finalizzate a mettere a disposizione servizi di elaborazione in maniera flessibile e scalabile: piattaforme *(Platform as a Service, PaaS)*, infrastrutture *(Infrastructure as a Service, IaaS)*, software *(Software as a Service, SaaS)* e alcune combinazioni per offrire servizi di storage *(Storage as a Service, STaaS)* o di condivisioni di dati *(Data as a Service, DaaS).*

Il Cloud Computing rappresenta quindi un insieme di tecnologie e di interfacce sviluppate al fine di conseguire maggior dinamismo nell'utilizzo di risorse di calcolo distribuito, siano esse CPU, dispositivi di storage o infrastrutture di rete. Sebbene la metafora del Cloud Computing abbia avuto origine degli anni '60, il modello ha cominciato a diffondersi solo negli ultimi 6 anni, dapprima attraverso servizi commerciali, e in seguito attraverso la sua adozione anche in ambiti di ricerca pubblica.

In termini commerciali, la Cloud rappresenta un modello di business per la vendita di servizi e sistemi informatici on-demand da parte di grandi e piccoli provider di risorse private. La stessa modalità può però essere applicata all'interno di un'organizzazione di dimensioni significative, per condividere e rendere disponibili le risorse evitando sprechi e duplicazioni.

In alternativa a mettere in comune servizi di base, per le esigenze della ricerca scientifica si può pensare a rendere disponibili come servizi accessibili via Web raccolte di funzionalità specifiche che rispecchiano le necessità e le consuetudini d'uso di ciascuna comunità di ricerca. È l'approccio degli *Science Gateway*, tramite i quali è possibile eseguire tutta una serie di operazioni, come eseguire applicativi

Workshop GARR Calcolo e Storage Distribuito - Selected Papers. Roma 29-30 novembre 2012
ISBN: 978-88-905077-4-8 © 2013 Associazione Consortium GARR

su infrastrutture Grid o Cloud, ricercare informazioni all'interno di database remoti, movimentare grandi quantità di dati, ecc., senza la necessità di dover imparare un linguaggio di programmazione specifico e senza avere alcuna particolare conoscenza del funzionamento complessivo dell'infrastruttura stessa.

Nell'ambito del workshop GARR Calcolo e Storage Distribuito, nato dall'idea a lungo perseguita da Enzo Valente e dal GARR di riunire le tante anime presenti nelle varie e variegate comunità della ricerca Italiana al fine di esplorare e possibilmente mettere in campo sinergie e condivisioni di competenze e risorse, sono stati presentati 26 contributi da parte di ricercatori e tecnologi impegnati nei vari settori ICT, includendo due keynote speech, da parte di Fabrizio Gagliardi (Direttore del Technical Computing, Microsoft Research), e Ignacio Llorente (Direttore del progetto OpenNebula). Nella fase conclusiva del workshop sono stati anche presentati tre position paper da parte di ENEA, INAF e INFN, contenenti la loro visione strategica per il calcolo e lo storage distribuiti, con l'obiettivo di stimolare una discussione aperta ai circa 130 partecipanti all'evento. Da questa discussione è emersa come fondamentale la necessità di perseguire un sistema integrato di e-Infrastructure della ricerca Italiana, che possa capillarmente raggiungere con i suoi benefici anche la vastissima comunità di piccoli gruppi di ricerca e singoli ricercatori. I grandi attori del settore della ricerca Italiana, come CNR, ENEA, INAF, INFN, INGV insieme ovviamente al sistema universitario e al GARR, possono fare un grande lavoro di condivisione di competenze e infrastrutture che possa finalmente modernizzare gli strumenti ICT, rendendoli fruibili a tutti, e conseguentemente le modalità di fare ricerca nel nostro Paese.

In questo volume sono stati selezionati i contributi ritenuti più significativi da parte del comitato di programma del workshop.

Buona lettura!

Chair del Workshop

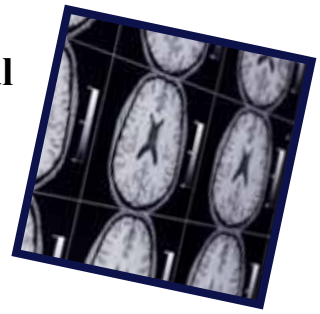
Giuseppe ATTARDI - Università di Pisa (co-chair)
Vincenzo VAGNONI - INFN (co-chair)

Comitato di programma

Roberto BARBERA - INFN
Giovanni BRACCO - ENEA
Luca DELL'AGNELLO - INFN
Marco PAGANONI - INFN

Tutte le presentazioni e maggiori informazioni sono disponibili sul sito dell'evento:
http://www.garr.it/ws-garr-csd

Progetto DECIDE: un esempio di infrastruttura al servizio della comunità biomedica



Valeria Ardizzone

Consortium GARR

Abstract. Il morbo di Alzheimer è la causa più comune di demenza, stimata nel 40-70% dei casi tra la popolazione oltre i 65 anni di età, e la sua diffusione è destinata ad aumentare notevolmente nei prossimi decenni. La demenza ha un enorme impatto sociale sulle famiglie, sui governi e sui loro settori sanitari e sociali. Per questo motivo, la malattia di Alzheimer non è solo una priorità europea, ma planetaria. Il modo in cui i ricercatori guardano al morbo di Alzheimer e alle demenze in generale è cambiato notevolmente: la demenza ora è vista come una fase avanzata dell'evoluzione della malattia, mentre lo scopo prioritario è quello di identificare la malattia di Alzheimer nella sua fase iniziale di sviluppo, utilizzando una combinazione di risultati prodotti dall'imaging strutturale (MRI), funzionale (FDG-PET), molecolare (PIB-PET) e dai test biochimici (analisi del CSF). Anche le analisi di test elettroencefalografici (EEG), in termini di densità di potenza e coerenza spettrale, stanno acquisendo sostegno da parte della comunità scientifica, poiché possono essere utilizzati per lo screening preliminare di grandi campioni di popolazioni.

1. Introduzione

Il progetto DECIDE [1] è incentrato sul fornire un valido supporto ai neurologi e alle varie figure mediche coinvolte nella diagnosi e prognosi delle malattie neurodegenerative. Il progetto, impiegando la e-Infrastructure basata su Grid, prevede la realizzazione di un servizio la cui comunità di riferimento sia in primo luogo quella clinica piuttosto che quella della ricerca. Il principale obiettivo è di fornire ai medici degli ospedali, e non solo dei grandi centri specializzati di ricerca e ricovero, strumenti efficaci per determinare i marcatori clinici per la diagnosi precoce dei disturbi neurologici (malattie neurodegenerative come l'Alzheimer) e psichiatrici (schizofrenia), insieme con la loro rilevanza prognostica.

Il raggiungimento dello scopo richiede lo sviluppo di una nuova infrastruttura nella cui progettazione, essendo il progetto focalizzato più sull'utenza della comunità clinica, è stata tenuta in grande considerazione la presenza dei vincoli specifici in termini di facilità d'uso, di standardizzazione, di sicurezza, di riservatezza dei dati. Particolare attenzione è stata inoltre prestata alle questioni etiche e giuridiche connesse alla gestione, all'elaborazione ed alla distribuzione dei dati. Il servizio offerto a questa comunità ha bisogno di essere compatibile con la routine clinica, e quindi essere semplice da usare, robusto, ragionevolmente veloce e non richiedere troppa interazione con il sistema.

2. Metodologia

L'idea ispiratrice alla base del progetto non è stata quella di fare qualcosa "per" una comunità, ma piuttosto "insieme" ad una comunità. Gli utenti del servizio sono stati coinvolti da subito nella fase di sviluppo del servizio, al fine di garantire che le loro esigenze fossero prese in considerazione per raggiungere la piena fruibilità del servizio finale nell'ambiente clinico.

La piattaforma DECIDE è costituita da tre strati differenti, e cioè le reti della ricerca, le risorse Grid e le applicazioni specifiche del dominio di riferimento.

- La connettività di rete fornita dalla dorsale europea GÉANT [2] e dalle NRENs (le reti nazionali della ricerca e dell'istruzione) dei paesi che partecipano al progetto, interconnette con collegamenti ad alta capacità diversi tipi di strutture, quali centri clinici e di ricerca ed istituti universitari di ricerca;
- L'infrastruttura Grid è utilizzata come piatta-

Workshop GARR Calcolo e Storage Distribuito - Selected Papers. Roma 29-30 novembre 2012
ISBN: 978-88-905077-4-8 © 2013 Associazione Consortium GARR

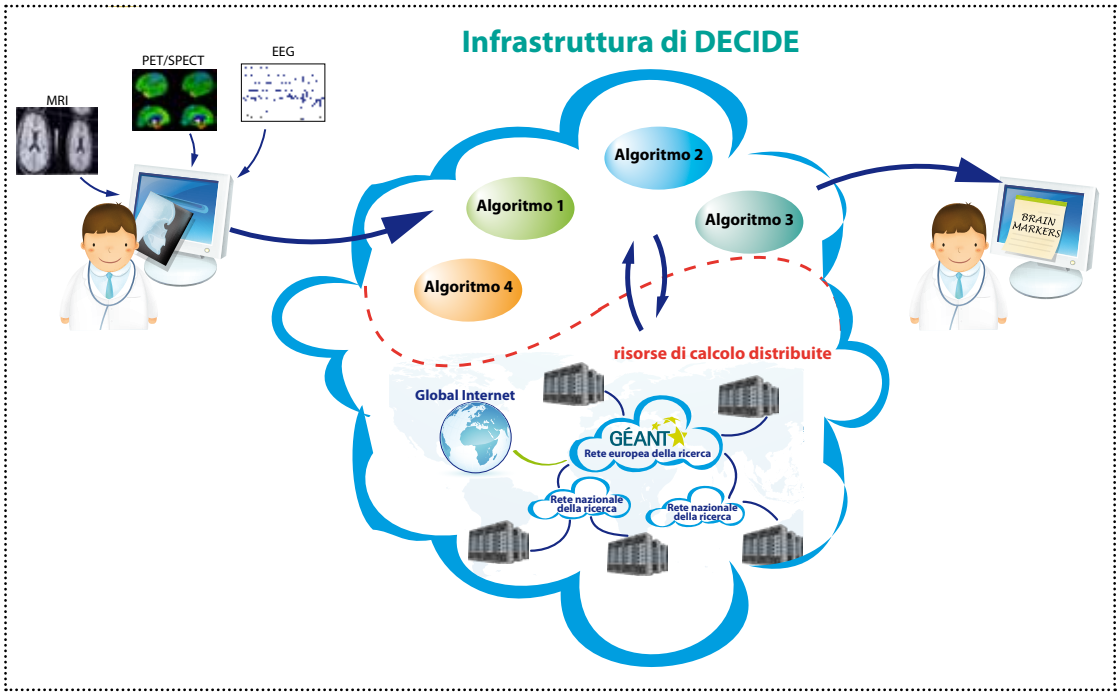


Fig 1 - Rappresentazione della eInfrastructure DECIDE

forma per consentire la collaborazione tra tutti i partner, una sorta di "collante" tecnologico atto ad armonizzare e unificare gli sviluppi, e come bacino di risorse di calcolo e storage in cui grandi volumi di dati possono essere ospitati in modo sicuro e le analisi possono essere eseguite;

- L'uso dei dati medici diversamente acquisiti *(Magnetic Resonance Imaging - MRI, Positron Emission Tomography - PET e Elettroencefalografia - EEG)* consente la combinazione di approcci diagnostici complementari sulla diagnosi delle malattie neurodegenerative, consentendo sinergie tra i diversi ambiti clinici e sostenendo possibili studi di correlazione tra i diversi approcci neurologici.

Una decisione chiave, presa sin dalle prime fasi del progetto, è stata quella di adottare, ove possibile, standard esistenti, al fine di contribuire a ridurre i costi di sviluppo e manutenzione, semplificare l'adozione e l'integrazione con altri servizi, e di ampliare la comunità di utenti. Il servizio DECIDE si basa su standard a tutti i livelli:

- dal punto di vista ICT, questo è vero dal livello di middleware e di rete (EMI / gLite [3], adottato su siti di produzione ufficiali EGI [4], la Grid Infrastructure europea), fino al livello del portale del progetto, il cosiddetto "Science Gateway" [5] basato su *Liferay portal framework* [6] (JSR 168/286 [7] per le portlets, SAML [8] per l'autenticazione, LDAP per il database di utenti, PKCS #11 per la crittografia e SAGA [9] per l'interfaccia con il *middleware*);
- dal punto di vista clinico, il progetto ha documentato e reso disponibile al pubblico le procedure di preparazione del paziente, preparazione esami, acquisizione e controllo di qualità dei dati, con il duplice obiettivo di migliorare la qualità e il contenuto informativo dei dati acquisiti e garantire che essi siano coerenti con i dataset di riferimento del progetto.

A livello tecnologico, si è deciso di sviluppare e implementare i servizi sulla base dell'infrastruttura Grid e del middleware sfruttando due caratteristiche:

- disponibilità di strumenti per integrare le risorse geograficamente distribuite, dove per "risorse" intendiamo principalmente le banche dati di immagini necessarie per addestrare gli algoritmi per il calcolo del volume di regioni cerebrali da immagini MRI, o quelle necessarie per fare confronti statistici di immagini FDG-

PET con i casi di pazienti "normali";
- capacità di stabilire criteri di autorizzazione fino al livello del singolo utente, requisito importante sia perché consente ai proprietari dei dati di mantenerne il controllo, pur facendo utilizzare da altri utenti l'informazione in essi contenuta tramite le applicazioni integrate, sia perché permette di controllare con precisione quale utente può avere accesso a una data applicazione.

In passato, l'adozione e l'impiego della tecnologia Grid, soprattutto da parte di utenti non esperti di IT, sono stati severamente limitati dalla scarsa usabilità e dai vincoli imposti dall'utilizzo di certificati personali, rigide procedure di sicurezza, interfacce a riga di comando, script di esecuzione scritti in linguaggi particolari. In DECIDE, queste problematiche sono state risolte con l'introduzione di uno *Science Gateway* (SG), ossia un portale web che integra un insieme di strumenti e di applicazioni, realizzati su misura del progetto, per soddisfare le esigenze della comunità. Il problema dell'identificazione degli utenti senza far uso di certificati personali Grid, è stato risolto con l'integrazione nel portale SG di certificati robot e mediando l'accesso ad essi tramite le Federazioni di Identità (quale l'italiana IDEM [10]). Questa configurazione combinata permette l'identificazione sicura degli utenti e risolve automaticamente i problemi della loro gestione in relazione al ciclo di vita delle identità digitali. È importante notare che nel corso degli ultimi due anni, l'efficacia dell'approccio SG è stata ampiamente riconosciuta, come testimonia il fatto che più di dieci Autorità europee di Certificazione hanno aggiornato le loro procedure per il rilascio anche di certificati di robot, e dalla presenza di numerosi (~ 30 nel solo EGI) SG che servono le varie comunità.

Gli utenti del servizio DECIDE sono stati classificati in tre gruppi, secondo le funzionalità messe a disposizione dal sistema:

1. "Neurologi": questi professionisti si prendono cura dei pazienti durante l'intero processo diagnostico, dalla diagnosi alla terapia. Hanno bisogno solo di richiedere degli esami da far eseguire ad altri utenti del servizio e recuperare i referti, che poi saranno combinate e usate per fare la diagnosi.
2. "Medici": questi professionisti (radiologi, neurofisiologi, medici nucleari) forniscono informazioni diagnostiche ai neurologi, per il test specifico di competenza.
3. "Scienziati": questi utenti possono avere diversi profili scientifici (fisici, matematici, statistici, ecc.). Hanno a che fare con gli algoritmi diagnostici e collaborano con i medici, fornendo la conoscenza e la comprensione della metodologia sottostante.

A ciascun gruppo di utenti corrisponde una vista personalizzata del portale e un crescente grado d'interazione con il servizio al quale sono autorizzati. Una specifica attività formativa è propedeutica all'uso del servizio per ciascuna categoria di utenti.

3. Descrizione della tecnologia

Come accennato nel paragrafo precedente, l'attuale implementazione del servizio è basata su middleware Grid: come primo passo, la Virtual Organization (VO) vo.eu-decide.eu è stata registrata nell'EGI Operations Portal ed i siti del progetto sono stati configurati per supportare tale VO ed offrire un insieme di servizi Grid (topBDII, WMS, LFC, CE, SE, UI). Per garantire il livello di servizio richiesto per la categoria degli utenti clinici, si è deciso di fare affidamento esclusivamente sui siti di produzione certificati EGI, poiché obbligati a rispettare certi livelli di disponibilità e affidabilità. Per quanto riguarda invece l'uso del servizio in termini di ricerca, le applicazioni possono funzionare su qualsiasi sito che supporti la VO del progetto. La scelta dei siti per l'esecuzione del lavoro è attivata richiedendo uno specifico software tag, che è pubblicato dal VO *Software Manager* tramite apposita procedura.

Il portale SG è uno strumento estremamente potente che rende la Grid utilizzabile dalle diverse comunità di utenti. Esso si basa sul f*ramework Liferay* ed è un contenitore di *portlet* 2.0 che supportano lo standard JSR-286. All'in-

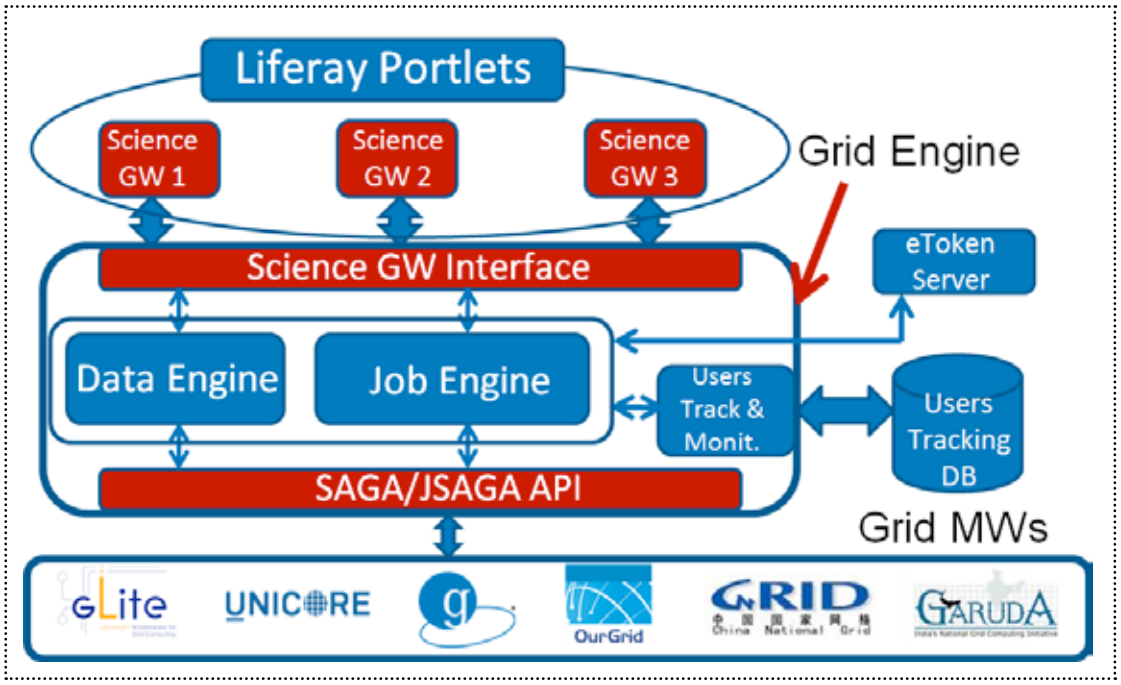


Fig 2 - Rappresentazione schematica delle componenti del portale SG

terno di questo framework, le applicazioni sono costruite e personalizzate in base alle esigenze degli utenti, da esperti sviluppatori di software, combinando, ove possibile, le portlet esistenti o scrivendone delle nuove. L'interazione con i servizi di Grid è mediata dal *Grid Engine*, uno strato di software compatibile con lo standard SAGA, in particolare con la sua applicazione JAVA (JSAGA) [11], progettato per interagire con una serie di middleware. Il Grid Engine isola efficacemente le applicazioni dai livelli sottostanti: invece, il portale SG può sottomettere con successo jobs a diverse infrastrutture basate su differenti middleware quali gLite, Unicore, Globus (in uso presso EGI), OurGrid (Brasile), CNGrid (Cina) e Garuda (India). Questo isolamento renderebbe l'eventuale passaggio ad altri paradigmi di calcolo (ad esempio, il cloud) piuttosto semplice.

Per evitare l'uso dei certificati personali degli utenti, il portale SG interagisce con il cosiddetto eToken server, grazie ad una leggera criptolibrary [12]: tale server contiene i certificati robot (uno per ogni applicazione/ruolo, memorizzati su una smartcard USB fisicamente collegata al server) e gestisce la creazione di *proxy* per conto dell'utente. Poiché con quest'approccio ogni utente è incanalato attraverso lo stesso tipo di proxy, l'associazione di ogni attività Grid all'identità digitale dell'utente è realizzata attraverso la registrazione ed il suo tracciamento da uno speciale database, lo *UserTracking DB*: questo rende il portale SG compatibile con le procedure di sicurezza EGI. Per quanto riguarda gli aspetti di sicurezza, l'eToken server appartiene ad una rete privata ed è accessibile solo dal portale SG; inoltre, a differenza di altri Science Gateway di uso comune, i certificati digitali non viaggiano mai sulla rete, solo i proxy di breve durata lo fanno (12 ore al massimo).

Nel portale SG i meccanismi di autenticazione e autorizzazione di un utente sono stati disaccoppiati. Per l'autenticazione, SG supporta diverse Federazioni d'Identità, come eduGAIN [13] e IDEM. Una Federazione d'Identità "*catch-all*" chiamata GrIDP [14] è stata creata e mantenuta dal progetto, per consentire la registrazione di utenti non appartenenti alle Federazioni. Lo SG può essere facilmente esteso ad altre Federazioni d'Identità che supportano lo standard SAML2 nelle sue implementazioni di Shibboleth e SimpleSAMLphp [15]. Il supporto alle Federa-

zioni d'Identità è un potente motore per la diffusione e l'adozione del servizio, grazie al fatto che non appena dei nuovi Provider d'Identità si uniscono ad una Federazione, questi si traducono in nuovi bacini di potenziali utenti del portale DECIDE.

Una volta che un utente è autenticato, deve essere autorizzato ad accedere ai servizi DECIDE. Per prima cosa si deve registrare al portale, operazione questa che innesca la creazione di una voce in un database LDAP, poi avrà bisogno di frequentare un corso di formazione anche online sull'uso del servizio scelto ed il ruolo richiesto, al fine di ottenere la relativa qualifica. Solo a questo punto alla voce relativa all'utente nel database LDAP viene associato il ruolo corrispondente appena conseguito. Dal punto di vista del portale SG, i benefici di questo meccanismo di autenticazione e di autorizzazione sono:

- offloading della gestione delle identità: non c'è bisogno di mantenere, proteggere, aggiornare, e verificare la validità delle credenziali utente sul SG;
- esecuzione delle procedure per garantire che tutti gli utenti del servizio siano stati adeguatamente formati;
- pieno controllo su chi è autorizzato ad utilizzare il servizio.

Dal momento che l'infrastruttura del progetto richiede la gestione di particolari tipi di dato dei pazienti, molta attenzione è stata posta sui problemi di sicurezza. Una tipologia di dato è particolarmente critico: le immagini FDG-PET che compongono il dataset di riferimento per i casi normali. A differenza delle immagini MRI, infatti, non esistono banche dati pubbliche per la loro conservazione,e forti vincoli legali, etici e anche finanziari limitano notevolmente la possibilità per gli ospedali e per i centri di ricerca di acquisirli. Una delle applicazioni offerte dal servizio DECIDE esegue un confronto tra l'immagine del paziente con un "cervello medio" costruito combinando le immagini di un certo numero di soggetti normali, al fine di evidenziare le regioni statisticamente significative di "ipometabolismo del glucosio". I migliori risultati sono stati ottenuti con l'uso di 50-100 immagini di pazienti normali, ma il dataset è stato allargato in modo da permettere di filtrare su parametri quali il sesso, l'età del paziente, il produttore e il modello dello scanner di acquisizione delle immagini. Le immagini dei soggetti normali sono una ricchezza immensa per un ospedale: basti pensare che un ospedale abbastanza grande può riuscire ad ottenerne mediamente una decina all'anno, sempre che abbia i fondi necessari per acquisirle.

Per permettere ai centri di ricerca medica coinvolti nel progetto di condividere efficacemente tali preziose risorse senza rinunciare alla proprietà, è stato adoperato il servizio middleware "*SecureStorage*" [16], una soluzione Grid-aware per memorizzare i dati crittografati su degli Storage Element: in questo modo, neppure gli amministratori di sistema degli Storage Element sono in grado di accedere ai dati riservati. I file dei dati riservati sono criptati da un *server KeyStore* (KS), ospitato nel dominio amministrativo del proprietario dei dati (ospedale o clinica) e poi copiati su uno o più *Storage Element*: i job delle applicazioni in esecuzione sui *Worker Nodes* possono contattare il KS per recuperare la chiave di decriptazione e, se le autorizzazioni possedute sono soddisfatte, l'applicazione può avere accesso al dato ed utilizzarlo per l'analisi statistica.

Le informazioni relative a ciascun file di dati sono memorizzate in un catalogo di metadati attraverso il servizio gLibrary [17], garantendo l'accesso e la gestione da parte di utenti ed applicazioni.

4. Conclusioni

I servizi che sono stati realizzati nel progetto DECIDE sono stati offerti agli utenti finali attraverso un nuovo tipo di portale SG, basato su Liferay e sugli standard più comuni, arricchito da un sofisticato meccanismo di autenticazione ed autorizzazione in grado di facilitare l'accesso e l'uso della tecnologia Grid, garantendo nel contempo il rispetto ed il controllo di ruoli e privilegi personalizzabili. Il DECIDE SG, inoltre, con-

sente la creazione e la gestione di grandi librerie digitali distribuite di immagini mediche e offre il servizio per la loro criptazione prima di memorizzarli sull'infrastruttura.

La sostenibilità dell'infrastruttura è garantita dal fatto che tutti i siti che fanno parte del servizio di produzione appartengono a organizzazioni che fanno parte delle iniziative Grid nazionali stabilite nei vari Paesi. Diverse iniziative programmate sono state previste per raggiungere la sostenibilità a lungo termine e descritte nel piano di sostenibilità realizzato prima del termine del progetto (corsi di formazione sull'uso dei servizi offerti, nuovi finanziamenti ottenuti partecipando a varie call nazionali ed europee, ecc.) e alcune delle azioni previste hanno già portato all'ottenimento di nuovi finanziamenti per proseguire il lavoro, a testimonianza del forte interesse suscitato da questo progetto.

La lezione principale che scaturisce da questa esperienza è che il coinvolgimento degli utenti già dalla fase di progettazione è fondamentale per identificare i bisogni e le esigenze specifiche delle comunità da raggiungere. In una fase successiva, la presenza di una comunità di motivati *early adopter* e il sostegno di alcuni opinion leader nel campo della ricerca e cura della Malattia di Alzheimer assicurano che il servizio fornito sarà accettabile per la più ampia comunità. Inoltre, problemi di sostenibilità devono essere seriamente considerati, poiché l'orizzonte temporale di un ospedale o di un centro di ricerca è tipicamente di diversi anni. A questo proposito, la scelta di aderire a standard e di sviluppare il servizio utilizzando un approccio stratificato garantisce che il prodotto risultante possa essere facilmente adattato alle future tecnologie.
Riteniamo che l'approccio DECIDE giocherà un ruolo significativo nel mettere a disposizione dei cittadini europei procedure cliniche di qualità scientificamente avanzata.

Ringraziamenti

Il lavoro descritto in questo articolo è stato realizzato all'interno del progetto DECIDE (*Diagnostic Enhancement of Confidence by an International Distributed Environment*), finanziato dalla Commissione Europea nell'ambito del 7° Programma Quadro per la Ricerca e lo Sviluppo Scientifico e Tecnologico. Si ringrazia il consorzio del progetto per aver reso possibile questi risultati. Maggiori informazioni su DECIDE sono disponibili su: www.eu-decide.eu

Riferimenti Bibliografici

[1] DECIDEProject:https://www.eu-decide.eu/

[2] GEANT Project: http://www.geant.net.

[3] gLite middleware: http://glite.cern.ch , European Middleware Initiative, http://www.eu-emi.eu/

[4] European Grid Infrastructure (EGI): http://www.egi.eu.

[5] Science Gateways are discussed in: Wilkins-Diehr N., Gannon D., Klimeck G., Oster S., Pamidighantam S. (2008), TeraGrid Science Gateways and Their Impact on Science, IEEE Computer 41(11), 32-41.

[6] The Liferay portal framework: http://www.liferay.com.

[7] The JSR 286 standard: http://www.jcp.org/en/jsr/detail?id=286.

[8] The SAML standard: http://saml.xml.org.

[9] The SAGA OGF Standard Specification: http://www.gridforum.org/documents/GFD.90.pdf.

[10] The IDEM identity Federation: http://www.idem.garr.it.

[11] The JSAGA website: http://grid.in2p3.fr/jsaga/

[12] La Rocca G., Barbera R,, Ciaschini V, Falzone A., Monforte S. (2011). A new "lightweight" Crypto Library for supporting a new Advanced Grid Authentication Process with Smart Cards. Proceedings of Science (ISGC 2011 & OGF 31), 29.

[13] The eduGAIN inter-federation: http://www.edugain.org.

[14] The GrIDP federation: http://gridp.ct.infn.it.

[15] The Shibboleth System: http://shibboleth.internet2.edu

[16] Scardaci D., Scuderi G. (2007), A Secure Storage Service for the gLite Middleware, Proceedings of the Third International Symposium on Information Assurance and Security, p. 261-266.

[17] Calanducci A. et al. (2007), A Digital Library Management System for the Grid, Fourth International Workshop on Emerging Technologies for Next-generation GRID (ETNGRID 2007) at 16th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE-2007), GET/INT Paris, France, June 18-20, 2007.

Valeria Ardizzone
valeria.ardizzone@garr.it

Laureata presso l'Università degli Studi di Catania nel 2003, ha lavorato presso l'Istituto Nazionale di Fisica Nucleare nell'ambito della tecnologia Grid Computing, partecipando alle tre edizioni del progetto europeo "Enable Grid for E-sciencE". Dal 2008 al 2010 è stata responsabile dell'Attività Formazione in Commissione Calcolo e Reti dell'INFN.

Dal 2011 è dipendente presso il Consortium GARR come Technical Coordinator del Progetto DECIDE.

È stata membro del Program Committee in più di una ventina di Scuole Internazionali sul Grid Computing, ha partecipato come tutor esperto della tecnologia Grid Computing ad oltre 100 eventi di training in tutto il mondo ed ha collaborato attivamente alla realizzazione di diverse National Grid Initiative (NGI).

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

Marco Bencivenni[1, 2], Diego Michelotto[1], Andrea Ceccanti[1], Andrea Cristofori[1], Enrico Fattibene[1], Giuseppe Misurelli[1], Riccardo Brunetti[3], Paolo Veronesi[1]

[1]INFN-CNAF, [2]Università di Ferrara, [3]INFN 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 abbandonare 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.

Workshop GARR Calcolo e Storage Distribuito - Selected Papers. Roma 29-30 novembre 2012
ISBN: 978-88-905077-4-8 © 2013 Associazione Consortium GARR

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 machine 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 sulll'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 di fetta di utenti che richiedono potenza di elaborazione. Esistono anche realtà *Open Source* sviluppate in diversi istituti nell'ambito del progetto europeo 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 OCCI, 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-

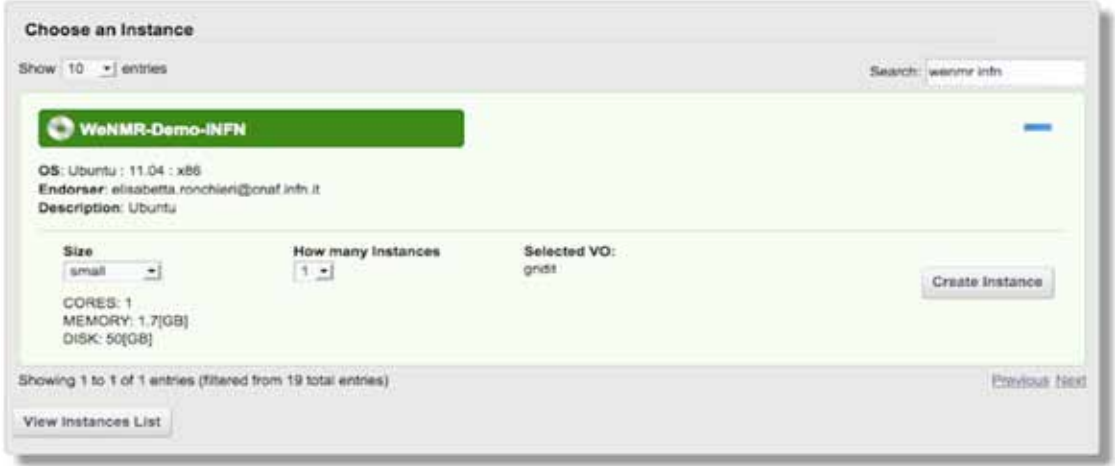


Fig 2 - Esempio di interfaccia per istanziare macchine virtuali

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.

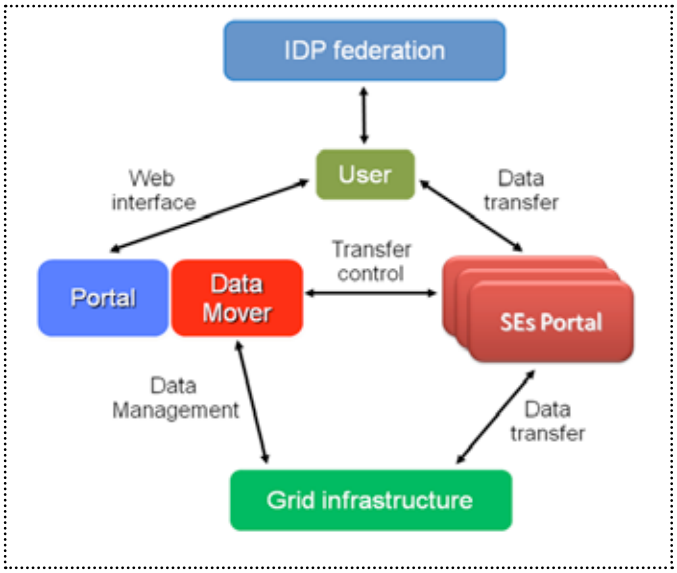


Fig 3 - Architettura del Data Management

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 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 scelto. 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-

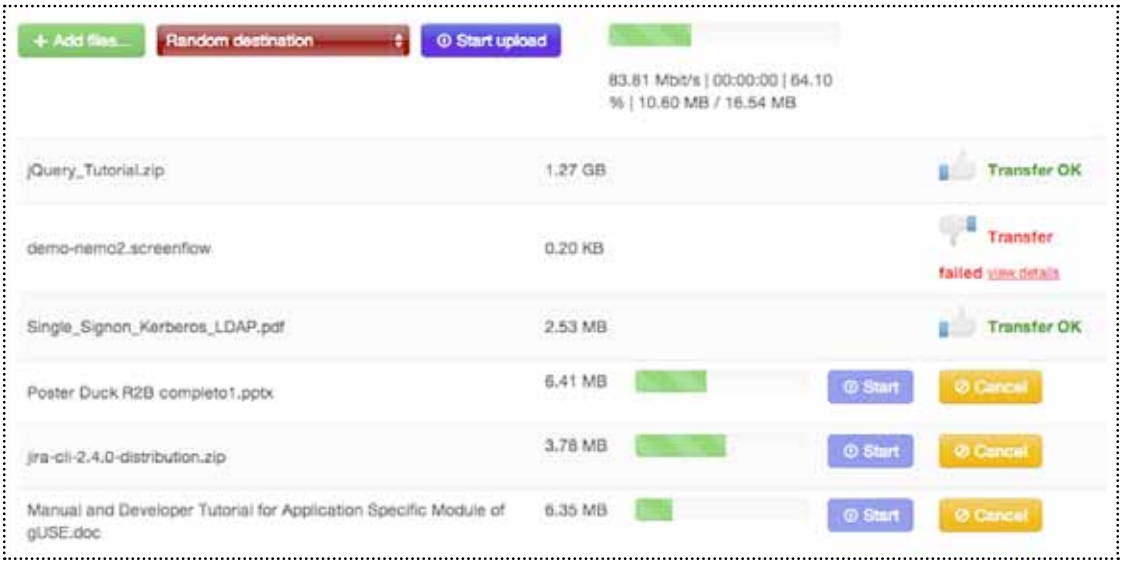


Fig 4 - Interfaccia di Upload

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] AjaxPlorer – 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.

Authentication e authorization federate nelle cloud: Estensioni a Shibboleth per l'applicazione in contesti di cloud computing

Andrea Biancini[1] , Luca Prete[2] , Simon Vocella[2]

[1]INFN – Sezione di Milano Bicocca, [2]Consortium GARR

Abstract. Molti sistemi informativi fanno uso di tecnologie cloud che sfruttano un paradigma distribuito ed espongono interfacce non necessariamente *web-based*. Contemporaneamente si va diffondendo l'uso di tecnologie per l'autenticazione e l'autorizzazione federate, che invece sono tipicamente web-based. Sono stati fatti vari tentativi per adattare i meccanismi federati alla tecnologia cloud, tuttavia le soluzioni introdotte sono ancora complesse e difficilmente integrabili. L'articolo propone tre nuove estensioni di Shibboleth che permettono: l'integrazione di sistemi non web-based, basati su Java e Python; l'autenticazione di sistemi Linux basati su PAM e NSS; l'integrazione del protocollo per l'archiviazione dati nelle cloud Amazon S3. Dopo un breve confronto con le precedenti tecnologie, si discute in dettaglio l'architettura di tali moduli, quanto sia già possibile fare e gli sviluppi futuri.

1. Introduzione

Oggi gli ambienti informatici fanno sempre più uso di applicazioni distribuite in differenti domini applicativi. In questi ambienti l'autenticazione e l'autorizzazione devono offrire meccanismi di *Single Sign-On* (SSO)[1]. GARR promuove un approccio federato alla gestione delle identità degli utenti attraverso la Federazione IDEM [2], che sfrutta l'implementazione di SAML 2 [3] offerta da Shibboleth [4]. Gli approcci di *Authentication and Authorization Infrastructure* (AAI) ben si applicano agli emergenti paradigmi di cloud computing, termine con il quale si è soliti indicare l'insieme di modelli di accesso a risorse informatiche (capacità computazionale, storage, applicazioni, ecc.) on demand attraverso Internet. Grazie alla virtualizzazione, i sistemi cloud fanno della trasparenza la loro caratteristica fondamentale. Per trasparenza s'intende la capacità di astrarre dallo strato fisico, disaccoppiando la gestione delle risorse fisiche dal loro utilizzo. Una soluzione cloud per l'università e la ricerca dovrebbe porsi l'obiettivo di estendere i modelli di federazione esistenti, sfruttandone i benefici.

Nell'ambito *storage cloud*, sono nate in Italia esperienze relative alla realizzazione di un servizio di questo tipo [5]. Dall'esperienza progettuale sono emerse alcune limitazioni tecnologiche di IDEM per il supporto specifico alle soluzioni cloud:

1. Cloud, per definizione, è multiprotocollo, quindi non accessibile solo da *web browser*. In questo caso, l'approccio standard offerto da Shibboleth può essere limitante e poco flessibile per applicazioni *mobile* e *client*.
2. In ambito cloud sono emersi nuovi protocolli, diventati standard *de facto*. Alcuni di essi descrivono e disciplinano gli aspetti di autenticazione e autorizzazione. I meccanismi federati di Shibboleth devono offrire interfacce compatibili con questi nuovi standard.

Tali aspetti saranno discussi dettagliatamente nel seguito dell'articolo, che è organizzato come segue: la prima sezione contiene una breve panoramica dei lavori correlati ai temi di autenticazione e autorizzazione federata; la seconda presenta la soluzione realizzata; la terza sezione presenta alcuni dettagli sulla realizzazione tecnica del software; l'ultima sezione presenta le conclusioni e i possibili sviluppi futuri.

2. Lavori correlati

Per l'accesso a schemi d'identità federata da

Workshop GARR Calcolo e Storage Distribuito - Selected Papers. Roma 29-30 novembre 2012
ISBN: 978-88-905077-4-8 © 2013 Associazione Consortium GARR

parte di applicazioni non basate sul web, sono in corso di realizzazione diverse iniziative. In particolare nell'ambito del software Shibboleth, il progetto Moonshot [6] si prefigge di affrontare questo problema. Il progetto mira esplicitamente a sviluppare una singola tecnologia che sia in grado di estendere i benefici delle federazioni d'identità a servizi non basati sul web.

La soluzione proposta da Moonshot è tanto completa quanto articolata. Essa si avvale di diverse tecnologie: Kerberos (per l'autenticazione in ambiete distribuito), le librerie *Generic Security Services* (GSS) e un server *Remote Authentication Dial In User Service* (RADIUS). Per quanto la soluzione sia valida e affidabile, essa si basa su un'architettura piuttosto complessa e richiede notevoli cambiamenti sulle macchine client, sul *Service Provider* (SP) e l'*Identity Provider* (IdP). Ciò rischia di rendere l'approccio di difficile adozione nelle federazioni esistenti.

La necessità di sfruttare federazioni d'identità Shibboleth in applicazioni non basate su web è particolarmente sentita nei servizi di Grid Computing [7], ispirati dall'articolo originario di Foster [8]. Questi ambienti hanno introdotto soluzioni specifiche ai problemi di sicurezza che differiscono da quelli proposti dal sistema SSO di Shibboleth. A causa della complessità del progetto Moonshot, l'integrazione dei due schemi di sicurezza è stata realizzata con soluzioni differenti: ad esempio quelle proposte da Wang [9] e Jensen [10] permettono agli utenti di spostarsi in modo trasparente tra applicazioni Shibboleth e ambienti Grid utilizzando le stesse credenziali. Queste soluzioni riescono a facilitare l'esperienza utente e a fornire risultati efficaci. Tuttavia si deve rilevare come loro non siano in grado di fornire soluzioni alle esigenze di AAI per servizi non basati sul web.

A oggi, la letteratura offre scarse informazioni sullo sviluppo di meccanismi d'integrazione di AAI cloud in federazioni già esistenti. Il Cloud computing si è sviluppato come paradigma autonomo, implementato attraverso soluzioni verticali molto specifiche, quindi scarsamente integrabili con il pregresso tecnologico. Tra le iniziative sviluppate intorno a Shibboleth per implementare meccanismi di autenticazione diversi e non basati solamente su *username* e *password*, nessuna ha preso ancora direttamente in considerazione i nuovi protocolli sviluppati in ambito Cloud.

3. Soluzioni proposte

Per risolvere i problemi descritti sono state realizzate, in ambito cloud federato per il mondo dell'università e della ricerca, tre estensioni di Shibboleth:

1. Autenticazione e autorizzazione per applicazioni non web-based (Java e Python).

Permette di sfruttare i meccanismi di autenticazione tipici delle realtà federate anche in applicazioni non basate sull'uso di un browser internet, quali ad esempio applicazioni desktop tradizionali. La soluzione garantisce il SSO, accedendo a risorse web da applicazioni che ne integrino le API. I meccanismi di autenticazione Shibboleth sono stati implementati in una libreria che comunica con il SP e l'IdP, utilizzando HTTPS e *Basic Authentication*.

La libreria è stata sviluppata sia per Java, come modulo JAAS [11], sia per Python, a oggi i linguaggi principali per realizzare applicazioni utente ad alto livello.

2. Autenticazione e autorizzazione di sistemi Linux (tramite PAM e NSS).

Questa estensione sfrutta un meccanismo simile a quello descritto nel punto precedente. Anch'essa è basata su HTTPS e *Basic authentication* e permette l'autenticazione di utenti in sistemi Linux attraverso Shibboleth. Grazie ai moduli sviluppati, in fase di login, SP e IdP sono contattati per autenticare l'utente e ottenere i dati necessari alle logiche di autorizzazione. In tal modo il sistema e le applicazioni riconoscono gli utenti della federazione trasparentemente e senza nessuna ulterior modifica.

Inoltre, anche applicativi come SSH, NFS o Apache, ereditano la possibilità di autenticare gli utenti attraverso Shibboleth. Ciò accade poiché molte applicazioni Linux delegano i compiti di autenticazione al sistema operativo attra-

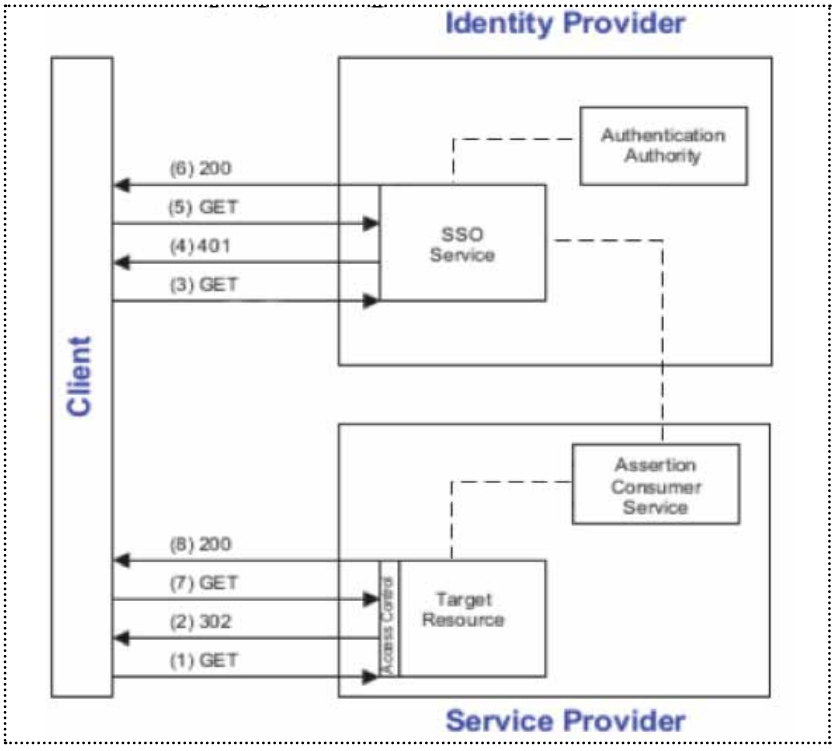


Fig 1 - Procedura di login di Shibboleth

verso i moduli PAM, aggiunti ad hoc nell'implementazione dell'estensione.

Per la realizzazione dell'estensione sono stati implementati degli specifici moduli per PAM e NSS, i meccanismi standard di user authentication e naming service di Linux

3. Autenticazione e autorizzazione tramite lo schema di sicurezza del protocollo S3.

Questa estensione permette di effettuare l'autenticazione Shibboleth di utenti che utilizzino il protocollo S3 [12] per la gestione dei dati. Il protocollo S3 è stato sviluppato da Amazon per il proprio servizio di storage cloud ed è oggi uno standard de facto, sia in ambito commerciale sia Open Source.

L'autenticazione di S3 non avviene tramite l'uso di username e password, ma attraverso la condivisione di un segreto tra l'utente e l'applicazione server. A ogni richiesta, il client cripta una stringa con una chiave segreta condivisa con il server, che non è mai trasmessa o distribuita. Il server, eseguendo la stessa operazione, confronta i risultati della cifratura autenticando l'utente in caso di corrispondenza. Per integrare tale modello di autenticazione in Shibboleth è stato sviluppato un nuovo modulo che permetta di verificare le credenziali utente utilizzando il segreto condiviso nel modo normato dal protocollo.

3.1 Architettura

La soluzione proposta sfrutta i meccanismi di autenticazione e autorizzazione di Shibboleth [13], permettendo di eseguire un login all'IdP usando il meccanismo Basic HTTP [14]. Il diagramma delle interazioni è quello mostrato in Figura 1.

Tuttavia sono necessarie le seguenti precisazioni:

1. La richiesta originale è indirizzata a una pagina web sul SP che si trova dietro autenticazione Shibboleth. La pagina produce un elenco di righe chiave=valore contenente le informazioni degli attributi da inserire nella sessione utente.
2. Il server web sul SP risponde con un redirect ad una pagina dell'IdP.
3. Il client segue il reindirizzamento e apre la pagina di login dell'IdP.
4. La richiesta ottiene una risposta HTTP 401 ("autorizzazione richiesta") e richiede l'autenticazione tramite HTTP Basic.
5. Il client esegue la stessa richiesta, specificando nell'header HTTP il nome utente e la password, verificati dall'IdP.
6. L'IdP interagisce conil SP per creare una sessione valida contenente gli attributi della sessione utente.
7. Il client apre l'URL originale, fornendo il cookie ottenuto dall'IdP dopo l'autenticazione. Ottiene quindi la pagina con le righe chiave=valore e inizializza la sessione utente sul client.

Il meccanismo simula ciò che accade all'interno del browser durante un'autenticazione Shibbo-

leth via web. Tutte le interazioni HTTP sono tuttavia automatiche e sono "nascoste" all'utente.

4. Implementazioni

Le implementazioni delle soluzioni descritte sono raggruppabili in tree aree: le implementazioni Java e Python, l'implementazione PAM e NSS e l'implementazione per l'autenticazione S3. Nel corso del capitolo sono presentati i dettagli relativi ai tre moduli realizzati.

4.1 Java e Phyton

Java, nella versione enterprise, è divenuta una piattaforma molto conosciuta e utilizzata per realizzare differenti applicazioni. Da Java 1.4 è stato introdotto un insieme di servizi per l'implementazione di meccanismi AAI: J*ava Authentication and Authorization Service* (JAAS, [11]). JAAS può essere utilizzato per due scopi:

- Autenticare gli utenti per determinare in modo affidabile e sicuro chi sta eseguendo il codice Java, indipendentemente dal fatto che il codice sia eseguito come applicazione *client*, *applet* o *servlet*;
- Autorizzare gli utenti e verificare che abbiano opportuni diritti di accesso.

Per la sua natura modulare, JAAS può essere esteso per integrare meccanismi di autenticazione differenti. Grazie al modulo realizzato è possibile integrare l'autenticazione Shibboleth e la gestione degli attributi nelle applicazioni Java anche non web-based. Il modulo JAAS implementato segue lo schema descritto nella sezione 3.1 per la validazione delle credenziali utente. Internamente, esso implementa uno ShibbolethPrincipal che contiene una mappa associativa degli attribute Shibboleth dell'utente e che, quindi, rappresenta la sessione di login dell'utente stesso.

Come ulteriore esempio, è stata sviluppata una libreria per l'integrazione del meccanismo di AAI descritto con applicazioni Python. Python è un linguaggio di programmazione ad alto livello utilizzato per lo sviluppo di applicazioni client. Differentemente da Java, Python non offre un insieme di API per gestire l'autenticazione e l'autorizzazione. Quindi, il modulo che realizza l'integrazione di Shibboleth è stato implementato da zero, senza implementare nessun *framework* di sicurezza preesistente. Tale modulo è contenuto nel *package shibauth* ed espone un metodo login che restituisce il nome di un utente autenticato e un dizionario contenente gli attributi dell'utente per la sessione Shibbolet corrente. La realizzazione di un'applicazione che sfrutti il modulo è quindi semplice e immediata.

4.2 Moduli PAM e NSS

I meccanismi standard utilizzati dai moderni sistemi Linux per autenticare gli utenti sono PAM e NSS. Le due librerie hanno diversi scopi e funzionalità. *Pluggable Authentication Modules* (PAM) è il cuore dei meccanismi di autenticazione e permette ai programmi di autenticare trasparentemente gli utenti, come descritto nella Linux-PAM *System Administrator's Guide* [15]. La libreria è stata estesa con l'implementazione di un modulo specifico al fine di garantire il SSO su macchine Linux. Il modulo realizzato autentica utilizzando HTTP Basic secondo lo schema generale descritto nella sezione 3.1. Esso utilizza le librerie *libcurl [*16] per gestire i dialoghi HTTP e i *cookie* al fine di effettuare l'autenticazione HTTP Basic. Un modulo PAM è uno *shared object* con un'interfaccia pubblica specificata negli header di PAM, collegata ai programmi utente per compiere le operazioni di autenticazione, accounting e per recuperare i valori da inserire nella sessione utente. Il modulo PAM sviluppato può essere utilizzato nelle catene PAM presenti sui sistemi Linux. Per esempio, modificando il file */etc/pam.d/login* e aggiungendo il modulo Shibbolet realizzato, è possibile permettere il login alla macchina Linux a utenti definiti su un IdP federato.

Per consentire a un sistema Linux di utilizzare directory esterne per gli utenti e i gruppi, è necessario implementare un altro modulo per la libreria N*ame Service Switch* (NSS), [17]. Questa libreria permette di definire i servizi per l'accesso a database contenenti diverse informazioni. Per quanto riguarda le attività descritte nell'articolo, i database rilevanti sono: passwd, con-

tenente gli utenti e i *group*, database con tutti i gruppi. Il servizio realizzato per integrare NSS con Shibboleth è basato sull'adozione di una *servlet*, distribuita sull'IdP. La servlet si connette al database di autorizzazione sottostante (di solito un database LDAP) e recupera le informazioni sugli utenti e i gruppi riconosciuti dall'IdP.

Grazie ai nuovi moduli, Linux riconosce gli utenti Shibboleth nello stesso modo di quelli locali. Un utente può accedere alla macchina Linux con le proprie credenziali Shibboleth, avere diritto di accesso ai file, utilizzarne le ACL e così via. Dopo il login, l'utente trova nel suo ambiente tutti i valori della sessione Shibboleth: gli attributi scaricati, infatti, sono inseriti nella sessione e possono essere utilizzati per eseguire SSO con altre applicazioni non web-based.

4.3 Integrazione con il protocollo S3

Amazon Simple Storage Service (S3) è un servizio cloud che permette l'archiviazione dei dati e files sul web. Esso rientra nell'insieme di servizi detti *Amazon Web Services* (la cloud realizzata da Amazon) e si presenta come uno standard *de facto*. Il protocollo S3 basa la comunicazione tra client e server su chiamate REST e, oltre agli aspetti di trasferimento e accesso ai file, definisce anche quelli relativi all'autenticazione. Per la parte di sicurezza, S3 sfrutta un meccanismo basato sulla condivisione di una chiave segreta (la cosiddetta *secret key*). A ogni chiamata S3, il client non scambia direttamente il segreto condiviso, ma crea un *token* ottenuto criptando un messaggio (la *canonical string*) attraverso un algoritmo noto e usando la *secret key* come chiave. La stringa criptata è inviata al server che è in grado di eseguire le stesse operazioni (utilizzando la stessa *secret key* condivisa con il client), quindi di autenticare l'utente.

Il meccanismo descritto non si presta a essere facilmente integrato con Shibboleth, che utilizza maschere di login che richiedono all'utente di inserire i propri nome utente e password. Per ovviare a questo problema è stato realizzato un nuovo modulo *ad hoc*, che permette a Shibboleth di verificare l'identità degli utenti secondo quanto stabilito nel protocollo S3. L'IdP è in grado di generare la *secret key* partendo da diversi attributi LDAP dell'utente (tra cui l'*hash* criptato e la sua password) e di comunicarlo all'utente stesso tramite l'invio di una mail. A questo punto il client opera in modo trasparente, senza nessuna modifica al suo comportamento standard, criptando la canonical string con la *secret key*. Il modulo realizzato sull'IdP, quindi, implementa un nuovo *Login Handler* specifico, che è in grado di processare le richieste in arrivo per autenticarle. Ad ogni richiesta da parte del client questo modulo genera nuovamente la secret key, cripta la canonical string e confronta il risultato con quanto inviato dal client. Se le due stringhe coincidono, allora l'utente è in possesso della secret key corretta e quindi può essere autenticato. In tal modo, l'implementazione realizzata è totalmente trasparente per i client e gli utenti possono continuare ad usare i tradizionali client S3. Tuttavia l'autenticazione avviene a questo punto ad opera di un IdP Shibboleth opportunamente esteso e quindi preservando l'architettura e gli schemi di identità federati in essere.

5. Conclusioni e sviluppi futuri

L'articolo ha descritto come siano state realizzate delle estensioni ai meccanismi di autenticazione e autorizzazione degli utenti per meglio adattare le federazioni d'identità all'erogazione di servizi cloud. In particolare le estensioni realizzate hanno mirato a garantire l'accesso alle risorse cloud, sia tramite il web, sia attraverso applicativi client non basati sul web. Un'ulteriore estensione è stata realizzata per permettere l'integrazione di schemi di autenticazione standard in ambito cloud all'interno di federazioni d'identità esistene. Come esempio per questo secondo punto, è stata descritta l'implementazione di un modulo S3 per l'autenticazione degli utenti all'interno di Shibboleth tramite il protocollo definito da Amazon.

Utilizzando le estensioni descritte è possibile implementare servizi cloud che sfruttino ed estendano tutti i meccanismi e le regole prescrit-

te da IDEM, la federazione di identità adottata dal mondo accademico e della ricerca in Italia. Il lavoro svolto rappresenta un primo passo verso meccanismi più complessi di estensione delle federazioni d'identità ai modelli e alle tecnologie cloud. Sebbene il lavoro descritto risolva già le esigenze specifiche di alcuni progetti in ambito di erogazione di servizi cloud, sono indubbiamente necessarie ulteriori indagini ed attività per raggiungere un livello di servizio migliore, in particolare:

- Supporto a WAYF. Le estensioni proposte non supportano il protocollo WAYF, per contattare l'IdP dell'ente di affiliazione dell'utente a seguito della verifica della sua identità. Occorre una nuova estensione per effettuare il discovery dei provider d'identità, utilizzando i profili descritti da SAML 2.
- Gestione multidominio per l'autenticazione in sistemi Linux. Qualora l'estensione per l'autenticazione utente su macchine Linux dovesse essere utilizzata in contesti federati di grandi dimensioni, sarebbe necessario risolvere un problema di mappatura degli utenti. I sistemi Linux associano a ogni utente uno *user id* numerico univoco. Gli id utente utilizzati dagli IdP dovrebbero essere associabili biunivocamente agli identificativi utenti dei sistemi.
- *Accounting*. Occorre capire come coniugare gli strumenti di monitoring dei singoli servizi cloud con quanto offerto dalle federazioni d'identità, ad esempio come gestire in modo federato l'uso di risorse virtuali fino all'esaurimento delle disponibilità assegnate a un utente.
- Performance. Relativamente al modulo per l'integrazione del protocollo S3, in ambiente di produzione è necessario introdurre meccanismi di *caching* sicuri sul SP. In tal modo non sarà necessario, una volta autenticato il client, dover contattare nuovamente l'IdP per una nuova richiesta di autenticazione.

Riferimenti bibliografici

[1] Shaer, C. 1995. Single sign-on. Network Security 1995, 8, 11–15

[2] https://www.idem.garr.it/

[3] Cantor, S., Kemp, J., Philpott, R., and Maler, E. 2005. Assertions and protocols for the OASIS security assertion markup language (SAML) v2.0. Tech. rep. 03.

[4] Morgan, R. L., Cantor, S., Carmody, S., Hoehn, W., and Klingenstein, K. 2004. Federated security: The shibboleth approach. EDUCAUSE Quarterly 27, 4, 12–17.

[5] Valli, C., Biancini, A., Reale, M., Farina, F., Vocella, S., Galeazzi, F. (2012). GARR Cloud Storage GARRBox. Proceedings of TICAL 2012.

[6] Howlett, J. and Hartman, S. (2005). Project Moonshot. Tech. rep. 07

[7] Kesselman, C. and Foster, I. 1998. The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann Publishers

[8] Foster, I., Kesselman, C., Tsudik, G., and Tuecke, S. 1998. A security architecture for computational grids. In Proceedings of the 5th ACM conference on Computer and communications security. CCS '98. ACM, 83–92

[9] Wang, X. D., Jones, M., Jensen, J., Rirchards, A., Wallom, D., Ma , T., Frank, R., Spence, D., Young, S., Devereux, C., and Geddes, N. 2009. Shibboleth access for resources on the national grid service (sarongs). In Proceedings of the 2009 Fifth International Conference on Information Assurance and Security - Volume 02. IAS '09. IEEE Computer Society, 338–341

[10] Jensen, J., Wallom, D., Spence, D., Tang, K., Meredith, D., and Trenthefen, A. 2007. Shibgrid, a shibboleth based access method for the national grid service. In UK e-Science All Hands Meeting

[11] http://java.sun.com/products/jaas

[12] http://aws.amazon.com/s3

[13] Erdos, M. and Cantor, S. 2005. The Shibboleth architecture. http://shibboleth.internet2.edu

[14] Franks, J., Hallam-Baker, P., Hostetler, J., Lawrence, S., Leach, P., Luotonen, A., and Ste-

wart, L. 1999. Http authentication: Basic and digest access authentication. RFC 2617.

[15] Morgan, A. G. and Kukuk, T. 1996. The Linux-pam system administrators' guide. http://kernel.org

[16] Stenberg, D. 1996. curl and libcurl. http://curl.haxx.se

[17] http://www.gnu.org

Andrea Biancini
andrea.biancini@mib.infn.it

Ha sviluppato la carriera all'interno di società dei settori Finance e Information Technology ove si è occupato di: gestione progetti IT, pianificazione e governo su temi di controllo e gestione (budget/costi), gestione del portafoglio progetti.

Da sempre interessato alla dimensione umana, ha organizzato eventi formativi e corsi. Sta concludendo un secondo corso di laurea in psicologia che mi sta permettendo di sviluppare ulteriori competenze.

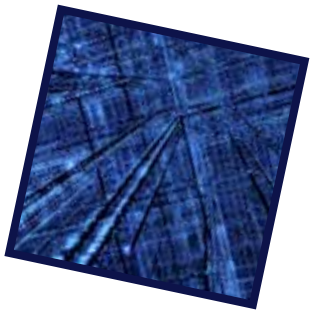
Luca Prete
luca.prete@garr.it

Svolge da diversi anni attività di consulenza in campo informatico, con particolare attenzione ai sistemi e alle reti. Interessato al networking, alle tecnologie cloud e quelle di virtualizzazione. Da poco più di un anno collabora con GARR occupandosi di Software Defined Networking.

Simon Vocella
simon.vocella@garr.it

lavora da diversi anni come software developer. Interessato a nuove tecnologie emergenti, in particolare a sistemi distribuiti e tecnologie cloud. Ha collaborato con GARR lavorando nei progetti europei FEDERICA e NOVI e nel progetto cloud storage GARRbox.

GaaS: Grid personalizzate per il calcolo su Cloud



Vania Boccia[1], Giovanni Battista Barone[2], Roberto Bifulco[2], Davide Bottalico[2], Luisa Carracciuolo[3], Roberto Canonico[2]

[1]INFN, [2]Università degli Studi di Napoli Federico II, [3]CNR

Abstract. Negli ultimi anni, l'utilizzo di ambienti distribuiti basati sul paradigma di Grid Computing ha consentito alle comunità scientifiche di risolvere problemi sempre più complessi, grazie alla condivisione di un numero elevato di risorse mediante protocolli di gestione robusti ed interfacce di accesso ben definite. Tuttavia il Grid è caratterizzato da un modello di aggregazione delle risorse piuttosto statico, in cui non è consentito agli utenti di intervenire in alcun modo né sull'organizzazione logica delle risorse né tantomeno sulla loro configurazione, sulla base delle proprie esigenze. Al modello di aggregazione Grid manca, dunque, quella dinamicità che è invece propria dell'on demand Computing. Da qualche anno la comunità scientifica ha mostrato interesse verso l'utilizzo del Cloud Computing che promette un ambiente di lavoro molto più flessibile, in cui l'infrastruttura di calcolo si modifica sulla base delle reali esigenze degli utenti. Tuttavia il passaggio dal modello Grid a quello Cloud, se da un lato porta indiscussi vantaggi in termini di flessibilità, presenta l'inevitabile svantaggio per l'utente di doversi adattare a nuove interfacce di accesso alle risorse. Di qui l'idea di realizzare GaaS (*Grid as a Service*), un insieme di servizi che consentono, combinando i paradigmi Grid e Cloud, di realizzare infrastrutture di Grid Computing configurabili *on demand.*

1. Introduzione

ll Grid è ormai largamente diffuso in molti contesti scientifici come modello di aggregazione, condivisione ed accesso a risorse di calcolo e storage. Molto è stato fatto negli ultimi dieci anni, in contesti nazionali ed internazionali, per realizzare e consolidare protocolli e strumenti. Tuttavia, il Grid Computing propone un modello di aggregazione delle risorse piuttosto statico. A titolo esemplificativo si possono individuare, tra le operazioni che gli utenti dell'infrastruttura non possono effettuare, l'aggiunta o la rimozione di nodi di calcolo, la modifica dell'organizzazione logica dei nodi di calcolo sulle code di esecuzione e la personalizzazione dei nodi di calcolo sulla base di esigenze specifiche (senza l'intervento dell'amministratore del sito Grid). Non è possibile inoltre modificare dinamicamente il numero delle risorse disponibili sulla base del reale carico dell'infrastruttura per migliorarne sia i livelli di efficienza sia la sostenibilità. Per superare i limiti appena esposti è necessario un modello di infrastruttura più "elastico" che consenta agli utenti di richiedere risorse (e servizi) *on demand*. Secondo la definizione del NIST [3] *"Cloud Computing is a model for enabling ubiquitous, convenient, on demand network access to a shared pool of configurable computing resources ... that can be rapidly provisioned and released with minimal management effort or service provider interaction"*. Da quanto detto, dunque, il Cloud Computing sembrerebbe fornire una risposta al problema posto.

Tuttavia il passaggio dal modello Grid a quello Cloud, se da un lato porta indiscussi vantaggi in termini di flessibilità, presenta l'inevitabile svantaggio per l'utente di doversi adattare a nuove interfacce di accesso alle risorse. Da queste considerazioni nasce l'idea di combinare i due modelli Grid e Cloud. Negli ultimi anni, diversi gruppi di ricerca hanno dedicato la propria attività alla risoluzione di questo problema, ognuno differenziando il proprio lavoro rispetto agli altri o per il modo di combinare i due modelli (Grid-su-Cloud, Cloud-su-Grid) o per la scelta delle tecnologie di deployment. In questo documento è descritto GaaS (*Grid as a Service*), un insieme di servizi che puntano a fornire agli utenti Grid risorse dell'infrastruttura di calcolo (IaaS) in un modo nuovo, che assomiglia al modello PaaS (*Platform as a Service*) [3].

Nel paragrafo 2 è descritta l'architettura Grid

Workshop GARR Calcolo e Storage Distribuito - Selected Papers. Roma 29-30 novembre 2012
ISBN: 978-88-905077-4-8 © 2013 Associazione Consortium GARR

di riferimento per il lavoro, cui segue nel paragrafo 3 una panoramica sui lavori di integrazione Grid-Cloud di alcuni gruppi di ricerca europei. Nel paragrafo 4 è riportata una descrizione dei servizi che costituiscono GaaS focalizzando, per ciascuno di essi, l'attenzione sui possibili casi d'uso. Nel paragrafo 5 è descritta la proof-of-concept per GaaS. In ultimo sono riportate alcune conclusioni, attività in corso e sviluppi futuri.

2. L'architettura Grid di riferimento

L'architettura Grid di riferimento considerata è basata sul middleware EMI, sviluppato nel contesto di EGI (*European Grid Infrastructure*), che consente il *deployment* di un'infrastruttura Grid, accessibile alle varie comunità di utenti organizzate in *Virtual Organization* (VO), ed offre servizi di alto livello e servizi di sito (Fig. 1). I servizi di alto livello sono utilizzati per la gestione dell'ambiente distribuito: autenticazione e autorizzazione (ad es. VOMS e MyProxy), diffusione e recupero delle informazioni relative ai siti Grid (ad es. BDII), allocazione delle risorse e job *scheduling* (ad es. LB/WMS). A livello di sito, per quel che concerne il calcolo, i nodi (WN) sono aggregati logicamente in code di esecuzione mediante un *Local Resource Management System* (ad es. Torque/Maui oppure LSF) e resi disponibili agli utenti mediante l'utilizzo di *Computing Element* (CE). Un utente dell'infrastruttura accede ai vari servizi da una *User Interface* (UI).

Nell'ambiente descritto, una tipica sessione di lavoro prevede che l'utente:

1. si autentichi;
2. definisca un job specificandone le richieste in termini di risorse di calcolo da utilizzare, task da eseguire, librerie o software da utilizzare per l'esecuzione, dati da trasferire, ecc.;
3. verifichi che sull'infrastruttura siano presenti risorse compatibili con le richieste espresse
4. sottoponga il job all'infrastruttura di calcolo;
5. controlli lo stato di un job inviato;
6. ritiri i dati di output ed eventualmente i log dell'esecuzione.

Lo scenario di utilizzo appena descritto non lascia spazio a nessun tipo di intervento di per-

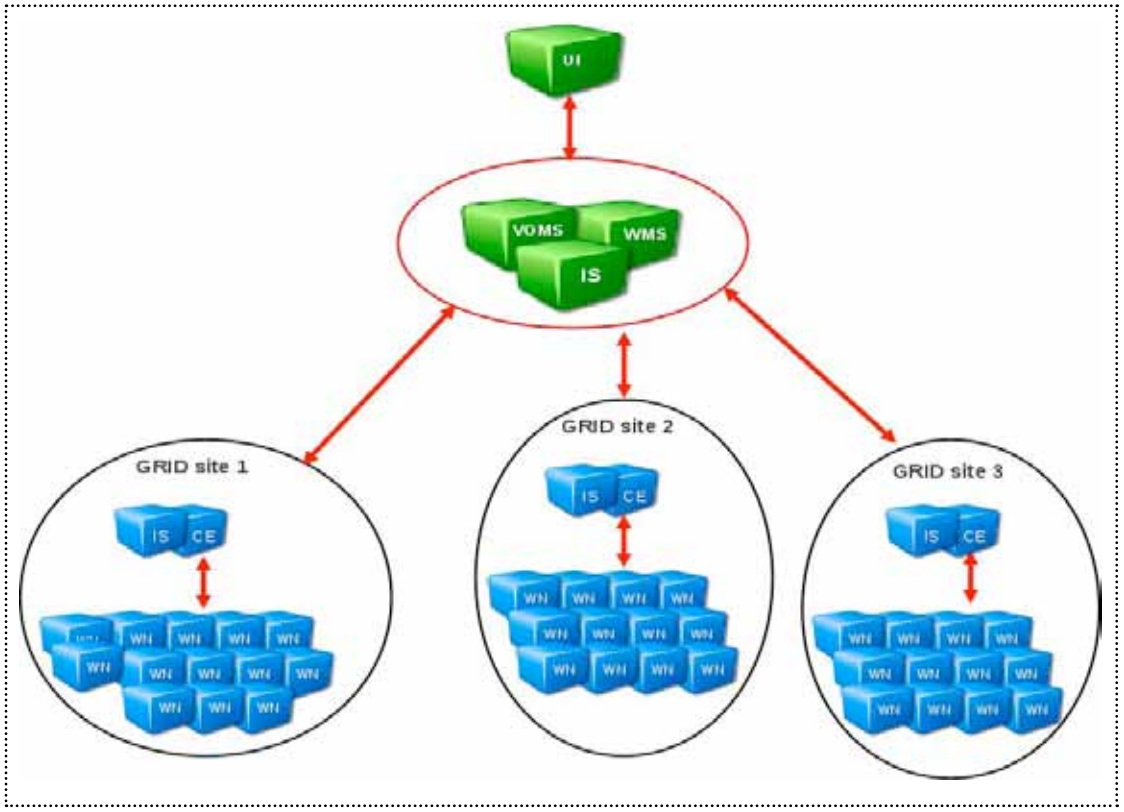


Fig 1 - Esempio di infrastruttura Grid basata sul middleware EMI

sonalizzazione dell'infrastruttura Grid da parte dell'utente. Infatti, se dalla verifica del punto 3 emerge che non ci sono sufficienti risorse di calcolo, non è possibile aggiungere *Worker Nodes* a una coda di esecuzione già presente. Oppure, se non è presente una libreria necessaria all'esecuzione del job, non è possibile per l'utente installare quella libreria senza l'intervento dell'amministratore del sito. Questa caratteristica di staticità dell'ambiente Grid può essere superata mediante tecniche di integrazione dei modelli Grid con quelli Cloud.

3. Stato dell'arte dell'integrazione dei modelli Grid e Cloud

Negli ultimi anni, diversi gruppi di ricerca hanno dedicato la propria attività all'integrazione dei modelli Grid e Cloud, ognuno differenziando il proprio lavoro rispetto agli altri o per il modo di combinare i due modelli (*Grid-su-Cloud*, *Cloud-su-Grid*) o per la scelta delle tecnologie di deployment.

Utilizzando un modello di integrazione di tipo Cloud-su-Grid, l'Istituto Nazionale di Fisica Nucleare (INFN) ha sviluppato WNoDeS [4], una soluzione che virtualizza risorse di calcolo e le rende disponibili mediante interfacce locali Grid o Cloud. L'architettura di riferimento è un ambiente Grid basato sul middleware EMI. I WN virtuali sono istanziati mediante l'utilizzo di Grid job speciali detti job di *power on*, definiti dall'utente selezionando immagini di macchine virtuali personalizzate, che vengono poi eseguite sulle risorse di calcolo (WN fisici) gestite dai CE.

Un modello di integrazione Grid-su-Cloud è invece quello scelto da StratusLab [2] che punta a sviluppare un'architettura Cloud completa ed open source che può essere inserita in ambienti Grid di produzione sia accademici che industriali. StratusLab fornisce servizi Grid utilizzando le risorse della sua infrastruttura Cloud (StratusLab IaaS). L'infrastruttura Grid prodotta sfrutta quindi la natura dinamica del Cloud Computing per fornire, on demand, risorse e servizi selezionati e personalizzati dall'utente. Le risorse e i servizi personalizzati sono memorizzati e resi disponibili grazie all'utilizzo di un *marketplace*.

Nel paragrafo 4 è descritta invece la soluzione GaaS (Grid as a Service) per l'integrazione dei modelli Grid e Cloud. In particolare è riportata l'esperienza fatta per la progettazione e la realizzazione di un'infrastruttura di calcolo flessibile che fa uso di risorse Cloud locali o remote (fisiche o virtuali), allo scopo di adattare un ambiente Grid di produzione alle reali esigenze dell'utenza, in nome di un utilizzo più efficiente e consapevole dell'infrastruttura di calcolo stessa.

4. GaaS: il modello e i servizi

GaaS nasce dalla duplice esigenza di fornire da un lato un modello di accesso Grid alle risorse già familiare a molte comunità di utenti, dall'altro di avere un'infrastruttura flessibile come quelle Cloud. Dal momento che il servizio fornito è un ambiente Grid, personalizzato sulla base delle esigenze degli utenti ed integrato in un ambiente di produzione preesistente, il nostro modello può essere classificato, secondo il NIST, come un Platform as a Service (PaaS) applicato ad ambienti Grid.

Attualmente il lavoro riguarda la possibilità di estendere la flessibilità degli ambienti Grid relativamente ai soli servizi di calcolo. A tal fine sono stati sviluppati i quattro servizi illustrati in Fig. 2.

Il servizio GaaS_WNS (*Worker Node Service*) consente l'integrazione di nodi di calcolo in code di esecuzione già attive su un CE. Il sistema acquisisce risorse da Cloud pubbliche o private, le configura come WN e avvia una procedura di riconfigurazione del CE al fine di inserire queste risorse in una coda di esecuzione e renderle in tal modo accessibili. Questo servizio è utile in situazioni in cui è necessario ampliare on demand l'insieme di nodi di calcolo.

Il servizio GaaS_QS (*Queue Service*) gestisce l'integrazione di nuove code di esecuzione su un CE. In questo caso l'utente ha la possibilità di modificare lo schema logico di aggregazione delle risorse di calcolo. Il sistema richiede la riconfigurazione del CE. Questo servizio è utile in casi in cui è necessario, su una parte dei nodi

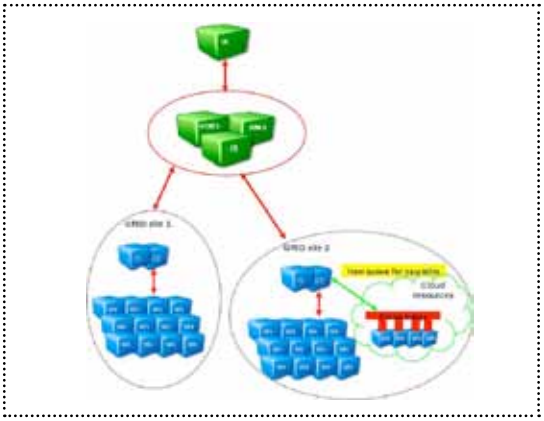


Fig 2a - GaaS_WNS

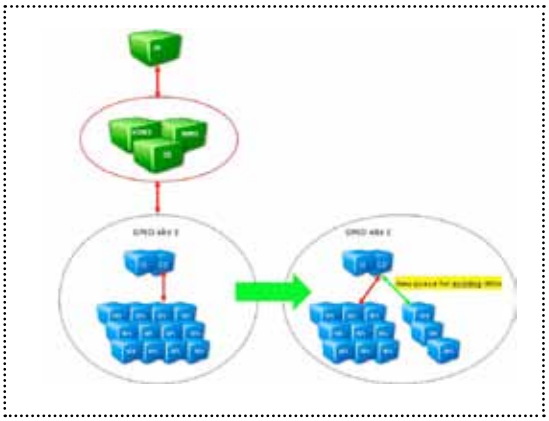


Fig 2b - GaaS_QS

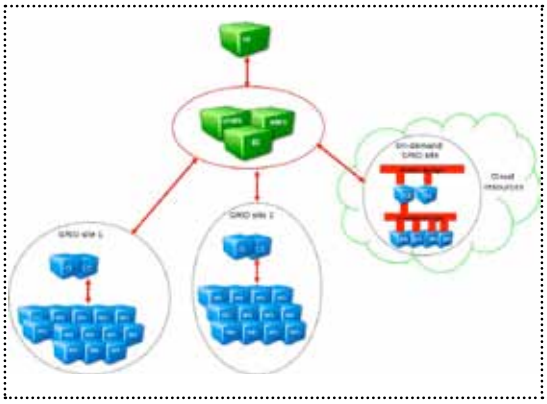


Fig 2c - GaaS_GSS

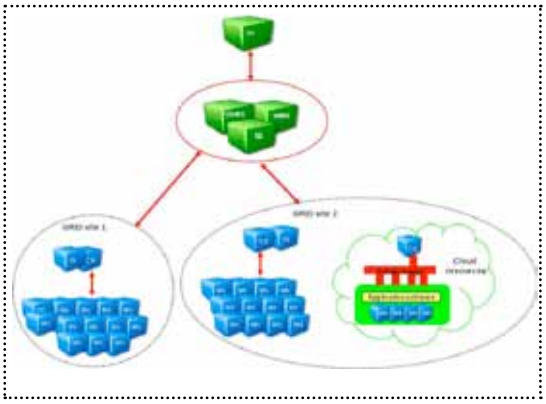


Fig 2d - GaaS_AES

di calcolo, modificare on demand e per un certo periodo, le politiche di accesso e di utilizzo delle risorse di calcolo (ad es. aumentando parametri come il tempo massimo di esecuzione, o modificando le priorità dello scheduler).

Il servizio GaaS_GSS (*Grid Site Service*) gestisce l'integrazione, in un'infrastruttura Grid completa e dotata di servizi di alto livello, di un sito Grid completo (CE e WN, SiteBDII) che sarà reso disponibile ad una VO esistente. Il sistema acquisisce le risorse Cloud (pubbliche o private), istanzia i servizi GaaS_WNS e GaaS_QS, e configura un sistema informativo al fine di integrare i servizi Grid configurati nell'infrastruttura Grid ospitante. Un possibile caso di utilizzo di questo servizio è relativo ad esempio a una comunità di utenti che necessita di condividere risorse con altre comunità, nell'ambito di un progetto di durata limitata e non desidera, o non può, affrontare i costi di acquisizione e gestione di risorse hardware da dedicare al progetto.

Il servizio GaaS_AES (*Application Environment Service*) fornisce, combinando opportunamente i servizi GaaS_WNS, GaaS_QS ed eventualmente GaaS_GSS, un ambiente Grid in cui sui nodi di calcolo è presente uno strato software ottimizzato, e più o meno ricco, configurato sulla base delle richieste degli utenti. Il risultato è un ambiente di calcolo che, dalle configurazioni del sistema operativo fino alle librerie per il calcolo scientifico, è realizzato su misura per l'utente che ne fa richiesta.

5. Caso di studio e strategie di deployment

I servizi descritti nel paragrafo 4 sono stati sviluppati ed integrati in un prototipo inserito nel contesto del datacenter S.Co.P.E. dell'Università di Napoli "Federico II". Le risorse di S.Co.P.E sono inserite in una Grid di produzione che è parte di IGI (Italian Grid Infrastructure) e di EGI.
Inoltre essa è dotata di tutti i servizi di alto livello, utilizzati dalle comunità dell'Ateneo e rap-

presenta, pertanto, il caso di studio ideale per il deployment e la verifica di GaaS.

Per la realizzazione del prototipo di GaaS, sono stati valutati e utilizzati, nei due anni di lavoro, differenti hypervisor (VMWARE, XEN) e sistemi di gestione Cloud (Eucaliptus, OpenNebula). Durante la fase realizzativa del prototipo, molto lavoro è stato dedicato:

- alla scrittura dei file di contestualizzazione, necessari alla personalizzazione delle macchine virtuali (VM) come servizi Grid (CE, WN e BDII);
- all'individuazione e all'utilizzo di metodologie più adatte a garantire il fast provisioning delle VM.

In generale il tempo necessario all'operazione di copia, configurazione ed allocazione delle VM configurate sulla macchina fisica che le ospiterà, rimane un fattore critico nel deployment di infrastrutture Cloud. Esso dipende da molti fattori connessi sia alle dimensioni delle immagini da copiare, sia alle caratteristiche dell'infrastruttura di rete e di storage sottostanti. Per ottimizzare l'utilizzo delle risorse e, allo stesso tempo, ridurre il tempo di *provisioning*, sono state adottate alcune strategie:

- è stata utilizzata un'unica immagine "minimale" della distribuzione Linux di riferimento per il middleware EMI (Scientific Linux) in modo da avere un unico template di VM (memorizzato in un repository);
- sono stati poi realizzati i file di contestualizzazione (anch'essi memorizzati nel *repository*) per la configurazione dei servizi Grid a partire dall'immagine di base della VM;
- sono state sfruttate le peculiarità del *Logical Volume Manager* (LVM) di Linux, che mediante *snapshots* e "delta meta-data", è in grado di ricostruire l'immagine attuale di un *file system* a partire da un Volume Logico di riferimento e dalle successive copie incrementali.

In tal modo è stato possibile ottenere il deployment dei servizi in qualche decina di secondi. L'approccio utilizzato ricorda quello seguito da StratusLab. Sussistono tuttavia alcune differenze che riguardano essenzialmente:

- le strategie di *provisioning*;
- l'utilizzo di un'unica immagine di VM da configurare sulla base dei diversi file di contestualizzazione in alternativa ad un marketplace di immagini di VM distinte (una per ogni ruolo Grid).

6. Conclusione e sviluppi futuri

Il lavoro svolto ha consentito di realizzare GaaS: un insieme di servizi che consentono, combinando i paradigmi Grid e Cloud, di produrre infrastrutture di Grid Computing configurabili on demand. Il prototipo realizzato, utilizzando come caso di studio il datacenter S.Co.P.E. dell'Università degli Studi di Napoli "Federico II", può essere considerato come una valida prova di principio dell'approccio utilizzato. Inoltre esso costituisce un utile banco di prova per la risoluzione di problemi ancora aperti legati all'integrazione Grid-Cloud e all'utilizzo dei data center in ottica Green [1].

Riferimenti bibliografici

[1] Barone, G., Bifulco, R., Boccia, V., Bottalico, D., Carracciuolo, L.: Toward a flexible, environmentally conscious, on demand high performance computing service. In: Data Compression, Communications and Processing (CCP), 2011 First International Conference on. pp. 136-138 (june 2011)

[2] Loomis, C., Airaj, M., Begin, M., Floros, E., Kenny, S., O'Callaghan, D.: StratusLab Cloud Distribution. In: Petcu, D., Vazquez-Poletti, J. (eds.) European Research Activities in Cloud Computing, pp. 260-282. Cambridge Scholars Publishing (2012)

[3] Mell, P., Grance, T.: The NIST definition of Cloud Computing

[4] Salomoni, D., Italiano, A., Ronchieri, A.: WNoDeS, a tool for integrated Grid and Cloud access and computing farm virtualization. In: Proceedings of the International Conference on Computing in High Energy and Nuclear Physics (CHEP 2010). pp. 18-35 (2011)

Ringraziamenti

Il presente lavoro è parte delle attività di un gruppo interdisciplinare denominato GTT (Gruppo Tecnico Trasversale) che è responsabile della gestione e dell'utilizzo efficiente del datacenter S.Co.P.E. dell'Università degli Studi di Napoli Federico II. Il lavoro è anche svolto nel contesto delle attività dell'*Italian Grid Infrastructure* (IGI).

Vania Boccia
vania.boccia@na.infn.it

PhD in Scienze Computazionali e Informatiche. La sua attività di ricerca scientifica e tecnologica si è svolta nell'ambito di diversi progetti nazionali (FIRB Grid.it e i PON SPACI, S.Co.P.E.) ed europei (EGEE ed EGI Inspire). Attualmente è tecnologo a tempo determinato presso l'INFN di Napoli.

Lo Science Gateway del progetto agINFRA per l'accesso a una data infrastructure per le Scienze Agrarie

Riccardo Bruno[1], Giovanni Allegri[2], Giuseppe Andronico[1], Roberto Barbera[1,3], Federico Bitelli[1,4], Antonio Budano[1], Antonio Calanducci[1], Edoardo A .C. Costantini[5], Marco Fargetta[1], Adrea Fornaia[6], Giovanni L'Abate[5], Salvatore Monforte[1], Antonio Puliafito[7], Rita Ricceri[1], Federico Ruggieri[1], Davide Saitta[6], Massimo Villari[7]

[1]INFN, [2]GIS3W s.a.s., [3]Dipartimento di Fisica e Astronomia dell'Università di Catania, [4]Dipartimento di Fisica dell'Università di Roma Tre, [5]Consiglio per la Ricerca e la Sperimentazione in Agricoltura, Centro di ricerca per l'agrobiologia e la pedologia (CRA-APB), [6]Consortium GARR, [7]Facoltà di Ingegneria, Università di Messina

Abstract. agINFRA è un progetto che ha lo scopo di sviluppare strumenti per supportare la comunità delle scienze agrarie basati sull'uso di infrastrutture dati distribuite a livello europeo e intercontinentale, insistenti sulle reti nazionali ed internazionali della ricerca.

Quest'articolo descrive l'architettura dello *Science Gateway* che è stato sviluppato per agINFRA basandosi sul Catania *Science Gateway Framework* e illustra le sue caratteristiche di ambiente integrato per l'accesso e la gestione dei dati, offerte mediante l'adozione dei paradigmi sia di Grid computing che di cloud computing. In particolare vengono descritti degli standard adottati, quali SAGA e SAML, e la loro importanza per la sostenibilità dell'infrastruttura e dei suoi servizi.

A titolo di esempio, saranno mostrate alcune delle applicazioni disponibili agli utenti attraverso lo Science Gateway, tra cui l'*Italian Soil Information System* (ISIS).

1. Introduzione

agINFRA [1] è un progetto co-finanziato dalla Commissione Europea nell'ambito del Settimo Programma Quadro che ha lo scopo di sviluppare strumenti per supportare la comunità delle scienze agrarie, basati sull'uso di infrastrutture dati distribuite a livello europeo e intercontinentale, insistenti sulle reti nazionali ed internazionali della ricerca. Scopo di agINFRA è progettare e sviluppare una grande infrastruttura digitale per dati scientifici relativi all'agricoltura, su cui creare e offrire servizi dedicati alla comunità scientifica di riferimento. Tali servizi dovranno essere fruibili in maniera tale da non violare le regole di sicurezza imposte dalle infrastrutture di tipo Grid e Cloud coinvolte.

Nell'ambito delle scienze agrarie esiste una crescente quantità di argomenti multidisciplinari, i cui soggetti possono variare dallo studio delle piante all'orticoltura e all'ingegneria agraria, dall'economia agraria agli studi di tipo ambientale. Un numero sempre crescente di studi si sta focalizzando sulle interconnessioni tra questi temi, come ad esempio il collegamento tra cambiamento climatico e produzione o perdita di biodiversità, oppure la pressione esercitata da una specie sull'altra. Questi studi richiedono l'accesso trasparente e ubiquo a enormi volumi di dati da parte di una comunità globale di ricercatori.

Per soddisfare questa esigenza, l'Istituto Nazionale di Fisica Nucleare ha realizzato, nell'ambito di agINFRA, uno *Science Gateway* dedicato [2], basato sul Catania *Science Gateway Framework* [3,4], che permette ai membri della comunità di riferimento di accedere ad applicazio-

Workshop GARR Calcolo e Storage Distribuito - Selected Papers. Roma 29-30 novembre 2012
ISBN: 978-88-905077-4-8 © 2013 Associazione Consortium GARR

ni e servizi che vengono eseguiti trasparentemente su infrastrutture sia di tipo Grid che di tipo Cloud.

La descrizione dello Science Gateway di agINFRA è data nel paragrafo seguente, insieme alla presentazione di alcuni delle applicazioni e servizi disponibili al suo interno, mentre nel paragrafo 3 sono riportate le conclusioni.

2. L'agINFRA Science Gateway

Lo Science Gateway di agINFRA è un portale web che offre allo stesso tempo la possibilità di accedere ad applicazioni e servizi che girano su ambienti Grid o anche Cloud. Il portale consente di autenticare gli utenti tramite le credenziali possedute all'interno delle Federazioni d'Identità [5,6] alle quali appartengono e gestisce in modo separato l'autenticazione dall'autorizzazione, in modo da offrire a ogni utente la possibilità di accedere ai diversi servizi offerti dal portale in maniera coerente e rispettosa dei suoi ruoli all'interno della comunità di riferimento e, quindi, dello Science Gateway. La figura 1 mostra l'architettura dell'agINFRA Science Gateway la quale si compone di 3 componenti principali.

Il modulo AAI si occupa dell'autenticazione e dell'autorizzazione degli utenti. La prima si basa sull'uso delle cosiddette identità federate. Ciò è reso possibile dall'adozione dello standard SAML 2.0 [7] e della sua implementazione Shibboleth [8]. Al fine di massimizzare il numero di potenziali utenti, lo Science Gateway di agINFRA è stato abilitato quale *Service Provider* della federazione IDEM [6] e, attraverso questa, dell'inter-federazione internazionale eduGAIN [9]. Inoltre, coloro che non appartengono a nessuna federazione possono utilizzare la federazione *catch-all* GrIDP [10] e l'IDP Open [11] che sono co-gestiti dal GARR e dalla Sezione di Catania dell'INFN.

La parte del modulo AAI relativa all'autorizzazione è invece gestita tramite un server LDAP, il quale registra per ciascun utente i ruoli ricoperti all'interno delle varie organizzazioni facenti parte della comunità e le relative autorizzazioni che questi deve avere all'interno dello Science Gateway. Per soddisfare un preciso requisito di sicurezza, la registrazione delle autorizzazioni è gestita manualmente, garantendo o negando in modo preciso l'accesso alle risorse offerte dal portale.

Il secondo elemento dell'architettura dello Science Gateway di agINFRA è l'insieme delle diverse interfacce utente offerte dal portale per le varie applicazioni e servizi che sono integrati in esso. Queste sono implementate attraver-

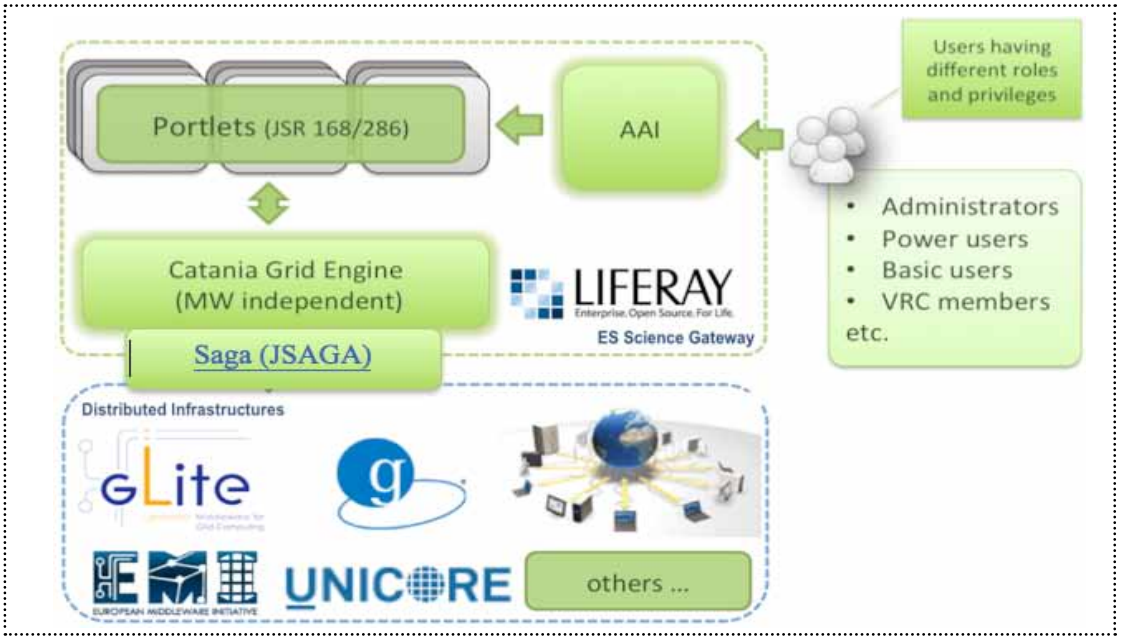


Fig 1 - Architettura dello Science Gateway di agINFRA

so *portlet* conformi allo standard JSR286 [12] e rese disponibili all'utente attraverso il *portlet framework Liferay* [13]. Le portlet fanno da tramite tra gli utenti del portale e le diverse infrastrutture distribuite attraverso il terzo e ultimo modulo; il Catania *GridEngine* [3]. Quest'ultimo elemento si occupa dell'accesso fisico alle infrastrutture distribuite, attraverso l'adozione dello standard SAGA [14] e della sua implementazione in Java chiamata JSAGA [15], e del tracciamento delle attività degli utenti, in conformità alle politiche che regolano l'accesso all'*European Grid Infrastructure* attraverso portali [16,17]. JSAGA, garantisce la possibilità di accedere a infrastrutture Grid sulle quali insistono middleware differenti, mediante l'utilizzo di librerie specifiche chiamate *adaptors*, che nascondono le specificità dei vari *middleware* allo sviluppatore di portlet. Sono queste ultime librerie che gestiscono l'accesso fisico alle risorse per una specifica infrastruttura, interfacciandosi poi con un unico insieme di chiamate API di alto livello.

Fig 2 - Home page dello Science Gateway di agINFRA

2.1 Applicazioni e servizi

Lo Science Gateway di agINFRA, la cui pagina d'accesso è riportata in figura 2, raggruppa l'accesso alle diverse applicazioni e servizi attraverso le voci di menu: "Applications" e "Cloud Services".

Sotto la voce "Applications" sono raggruppate le applicazioni installate sull'infrastruttura Grid del progetto, che accedono a risorse Grid utilizzabili attraverso il middleware EMI-gLite [18] e sono brevemente descritte di seguito:

1. AGRIS AP XML2RDF: converte file dal formato AGRIS AP XML sviluppato dalla FAO nel formato RDF, standard del web semantico, lanciando una serie di esecuzioni parallele su Grid.
2. AGROVOC *Tagging*: elabora i file di output provenienti da un servizio di *web crawling* e restituisce in output un file di triple in formato RDF.
3. *R Statistical Analysis*: R è uno strumento per l'analisi statistica di dati e la loro visualizzazione grafica.
4. *RiceInfo*: è un servizio di ricerca sviluppata dall'*Indian Statistical Institute* che offre risultati riguardanti tutte le tipologie di parassiti del riso quali: virus, vermi, insetti, ecc. Queste informazioni sono memorizzate in una *knowledge base* che contiene anche altre tipologie di dati riguardanti i difetti di nutrizione, la tossicità, i meccanismi di controllo per gli agenti contaminanti. I risultati delle ricerche sono riportati come insiemi di triple in formato N3. Il servizio integra anche risultati di ricerche semantiche effettuate su

DBpedia e GoogleScholar.

5. Weka: è una suite di algoritmi di machine learning che possono essere applicati a problemi di data mining in scienze agrarie. Gli algoritmi di Weka sono sviluppati in Java e questa applicazione permette di eseguire diversi algoritmi su un singolo file di input, distribuendo l'operazione su un'infrastruttura Grid.

Oltre alle applicazioni appena esaminate, lo Science Gateway di agINFRA offre al momento altre due applicazioni di diversa tipologia. Queste sono trattate nel successivo sotto-paragrafo.

2.2.1 Italian Soil Information System

Il servizio *Italian Soil Information System* (ISIS), la cui interfaccia è mostrata in figura 3, utilizza lo standard WebGIS per offrire mappe del territorio sulle quali presentare dati relativi alle varie tipologie dei suoli dele regioni italiane, correlandoli poi con i dati del suolo del territorio europeo, rispettivamente alle tipologie di suolo unitarie "*Soil Typological Units*" (STUs), a livello nazionale, e di sotto-sistemi a livello regionale.

ISIS è stato sviluppato da ricercatori Centro di Ricerca per l'Agrobiologia e la Pedologia del Consiglio per la Ricerca e la Sperimentazione in Agricoltura (CRA) ed è mantenuto presso la Sezione di Catania dell'INFN. Lo Science Gateway di agINFRA consente di accedere al servizio remoto in maniera del tutto trasparente all'utente nelle due modalità descritte di seguito.

2.2.2 Soil Map Annotator

Questa applicazione è in stretta relazione all'applicazione ISIS in quanto gestisce delle informazioni di tipo metadati sulle mappe del suolo utilizzate dall'applicazione ISIS. L'applicazione inoltre, permette di aggiungere delle annotazioni definite dall'utente sulle mappe e successivamente di eseguire ricerche non solo sui metadati associati alle mappe ma anche sulle annotazioni definite dagli utenti. L'utilizzo delle annotazioni è una forma di collaborazione sempre più in uso nell'ambito di progetti che coinvolgono numerosi partner o più in generale organizzazioni virtuali.

2.3 Cloud Services

Oltre a quella del *Grid computing*, il progetto agINFRA utilizza tecnologie di tipo Cloud, per tutti quei servizi che devono essere disponibili "24x7" e non possono quindi essere associati risorse non dedicate. L'applicazione ISIS è un esempio di servizio che gira su una Cloud il cui accesso da parte degli utenti è solamente possibile tramite lo Science Gateway. Inoltre, come mostrato in figura 4, l'interfaccia web offerta dal portale permette due diverse modalità di accesso a seconda delle autorizzazioni dell'utente. La prima permette un accesso di tipo pubblico e le funzionalità offerte dal servizio ISIS sono limitate ad un insieme minimo di opzioni. La seconda richiede che l'utente abbia eseguito il sign-in e offre ulteriori fun-

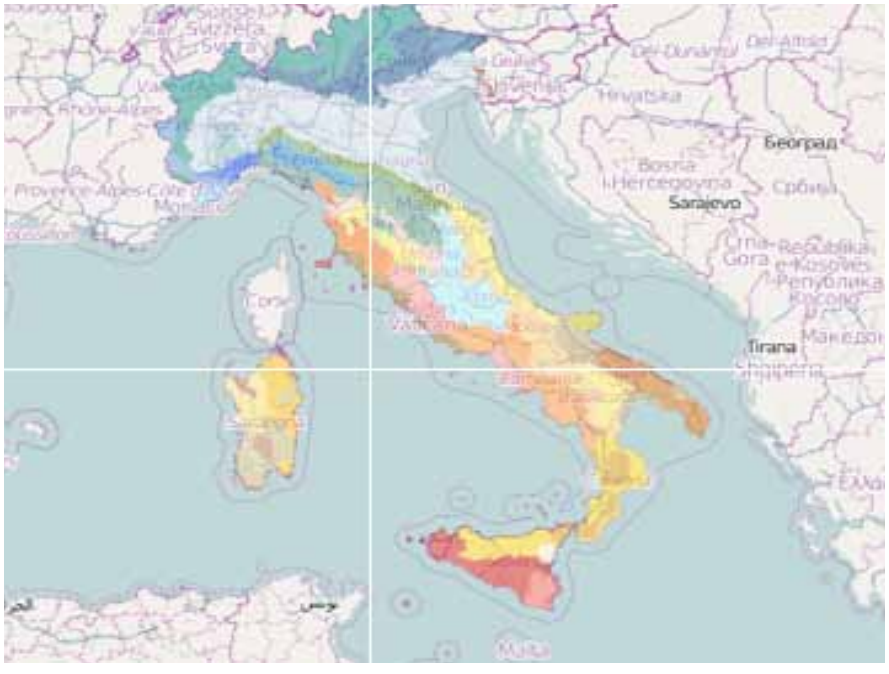


Fig 3 - Interfaccia WebGIS del servizio ISIS

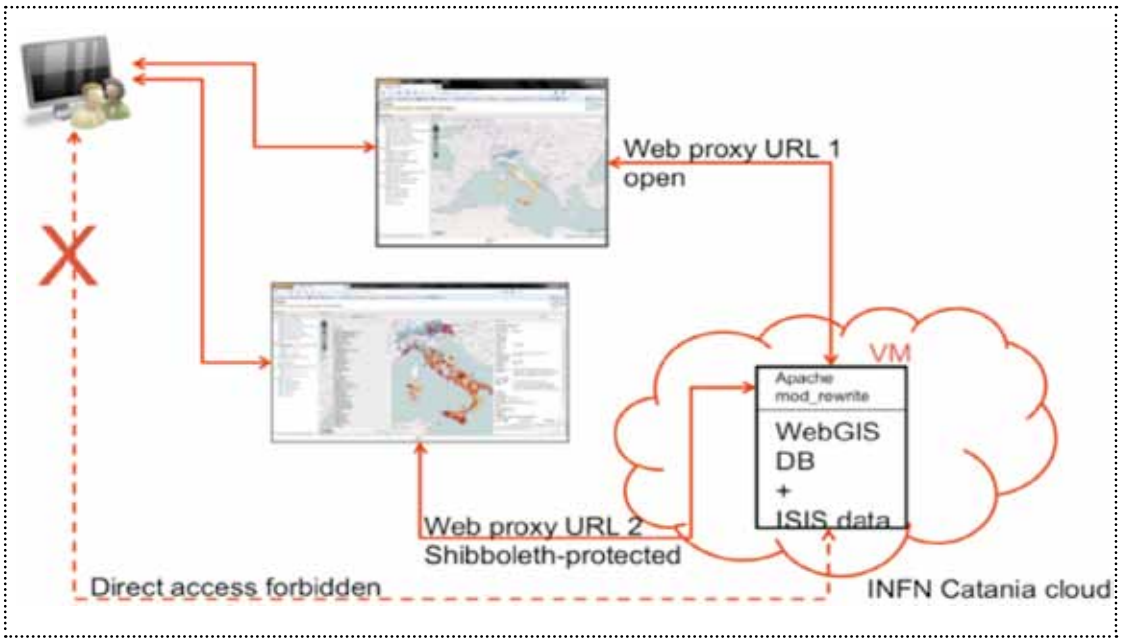


Fig 4 - Integrazione del servizio cloud ISIS nello Science Gateway di agINFRA

zionalità, come ad esempio la possibilità di eseguire interrogazioni al geo-database per ricercare informazioni sul suolo dei siti italiani in esso memorizzati.

Il progetto agINFRA utilizza come tecnologia Cloud il middleware CLEVER [19], che verrà brevemente descritto di seguito.

2.3.1 CLEVER

CLEVER è un innovativo middleware cloud, progettato e realizzato da ricercatori della Facoltà di Ingegneria dell'Università di Messina, che prevede la gestione di dispositivi virtuali come astrazione delle risorse fisiche. Questo facilita la gestione dei servizi cloud di tipo privato o ibrido. Tramite un'interfaccia grafica è possibile interagire con differenti infrastrutture di computer interconnessi via rete.

CLEVER offre la possibilità di gestire intere infrastrutture di tipo virtuale (IaaS). Il middleware è basato su un'architettura cluster di tipo distribuito, dove ogni cluster può appartenere a due diversi livelli gerarchici, come mostrato in figura 5.

Ciascun nodo CLEVER contiene un modulo di gestione host chiamato *Host manager* (HM) ed inoltre può anche includere un modulo di gestione cluster chiamato *Cluster Manager* (CM). Il Cluster manager è responsabile del trattamen-

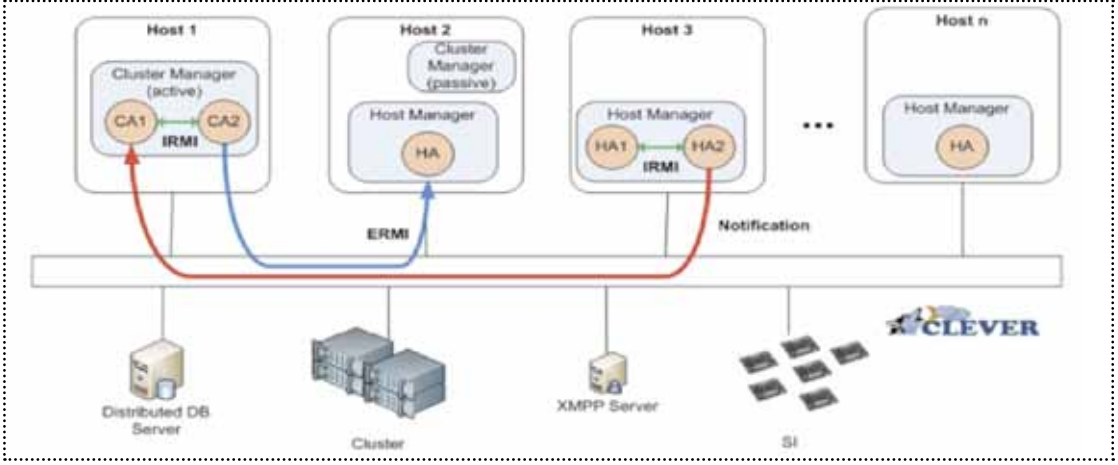


Fig 5 - Architettura del middleware CLEVER

Fig 6 - Vista della portlet MyCloud

to e dell'analisi del flusso dei dati mentre l'HM, a più basso livello, agisce come agente remoto per il CM. In questo modo nel Cluster si avrà almeno un CM attivo di alto livello e diversi HM di basso livello dipendenti da esso. I nodi CLEVER sono gestiti, all'interno dello Science Gateway di agINFRA, dalla portlet MyCloud, accessibile solo agli utenti del portale identificati come Cloud Manager sul server LDAP.

Come mostrato in figura 6, questa interfaccia permette di installare servizi diversi sui nodi della cloud con semplici azioni di trascinamento.

Una volta installato il servizio richiesto su uno dei nodi di CLEVER, questo sarà reso dinamicamente fruibile da parte del portale ai suoi utenti, esattamente come avviene per l'applicazione ISIS descritta sopra.

3. Conclusioni

Lo Science Gateway di agINFRA offre al progetto diverse funzionalità, che spaziano dall'accesso a file e ai relativi metadati all'esecuzione di applicazioni che richiedono numerose risorse di computazione su infrastrutture di calcolo distribuite geograficamente. Vi è pure l'opportunità di gestire in modo semplice e dinamico, servizi ospitati da nodi cloud e renderli poi fruibili attraverso l'interfaccia web del portale, riuscendo a separare accessi di tipo pubblico da quelli ristretti.

Il portale è stato sviluppato con i più moderni standard riguardanti l'autenticazione e autorizzazione, l'esecuzione di applicazioni su infrastrutture diverse, che vengono così rese interoperabili tra loro, e il trattamento di dati e metadati su infrastrutture distribuite. L'adozione di standard è un punto cruciale per la sostenibilità tecnologica del portale e di conseguenza per i servizi offerti dal progetto agINFRA. La sostenibilità tecnologica avrà sicuramente ripercussioni positive su quella economica.

Riferimenti Bibliografici

[1] www.aginfra,eu

[2] http://aginfra-sg.ct.infn.it

[3] V. Ardizzone et al. J. Grid Computing (2012) 10:689-707, DOI 10.1007/s10723-012-9242-3

[4] www.catania-science-gateways.it

[5] Per maggiori informazioni sulle Federazioni d'Identità esistenti nel mondo della ricerca scientifica, si visiti il sito web https://refeds.org

[6] Per maggiori informazioni sulla Federazione d'Identità italiana, si visiti il sito web www.i-dem.garr.it

[7] http://saml.xml.org

[8] http://shibboleth.net

[9] www.edugain.org

[10] http://gridp.garr.it

[11] http://idpopen.garr.it

[12] www.jcp.org/en/jsr/detail?id=286

[13] www.liferay.com

[14] www.gridforum.org/documents/GFD.90.pdf

[15] http://grid.in2p3.fr/jsaga

[16] https://documents.egi.eu/public/ShowDocument?docid=80

[17] https://documents.egi.eu/public/ShowDocument? docid=81

[18] www.eu-emi.eu

[19] http://clever.unime.it

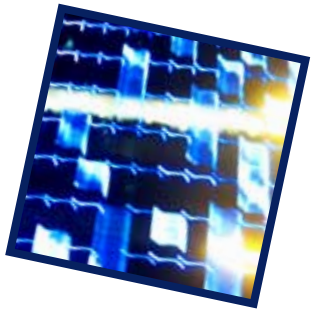
Ringraziamenti

Si ringraziano tutti i collaboratori dell'INFN sezione di Catania che hanno fatto del Catania Science Gateway Framework un prodotto affidabile al servizio delle comunità scientifiche italiane e internazionali nell'ambito di diversi progetti europei. Si ringrazia inoltre tutto lo staff tecnico e amministrativo del progetto agINFRA nonché il team del Consiglio delle Ricerche e Sperimentazioni in Agricoltura (CRA) per il prezioso contributo offerto durante l'integrazione dell'applicazione ISIS.

Riccardo Bruno
riccardo.bruno@ct..infn.it

Laureato in Scienze dell'Informazione presso l'Università degli Studi di Catania, ha lavorato presso una società operante nel settore della telefonia mobile. Dal 2006 svolge attività di ricerca presso l'INFN Sez. di Catania in relazione ad ambienti di calcolo e storage distribuito: Grid/Cloud.

Software-Defined Networking: Esperienze OpenFlow e l'interesse per Cloud



Mauro Campanella[1], Fabio Farina[1], Luca Prete[1], Andrea Biancini[2]

[1]Consortium GARR, [2]INFN – Sezione di Milano Bicocca

Abstract. Il protocollo OpenFlow è divenuto uno standard *de facto* nell'ambito del *Software-Defined Networking* (SDN). L'articolo descrive l'esperienza di ricerca condotta da GARR su tali tecnologie e il potenziale uso di SDN e OpenFlow nelle cloud. È descritta la creazione di un testbed virtuale *OpenFlow* e l'ideazione e realizzazione di un modulo software per l'utilizzo di topologie magliate a livello di data link nelle reti con controller OpenFlow.

1. Introduzione

Con l'avvento del paradigma *cloud* e dell'uso diffuso della virtualizzazione nei sistemi di calcolo, la gestione delle infrastrutture di rete diventa sempre più complessa, richiedendo maggior dinamicità e automazione. La densità e il numero delle risorse fisiche di calcolo disponibili sono in costante aumento grazie agli sviluppi delle microtecnologie. Il paradigma Cloud fa un uso esteso delle tecnologie di virtualizzazione che, slegando le risorse informatiche dalle loro componenti fisiche, ne moltiplica il numero e permette di fruirne in modo più efficace, flessibile e dinamico, per esempio con affidabilità tramite migrazione in tempo reale dei servizi. Lo stesso cambio di paradigma è richiesto e sta avvenendo a livello delle reti che astraggono sempre più dal livello fisico di implementazione e si presentano come infrastrutture virtualizzate.

Per essere efficace, un servizio basato su cloud deve armonizzare la virtualizzazione di calcolo e *storage* con la virtualizzazione delle infrastrutture di rete in ambiente ad alta densità. L'idea di migrare verso reti definite via software viene incontro a questa esigenza. Dal 2008 il nuovo paradigma, chiamato *Software-Defined Networking* (SDN) [1] propone di semplificare i nodi di rete (*router*|*switch*) disaccoppiando il piano d'instradamento (*switching*, realizzato tipicamente in hardware), dal piano di controllo (in software), favorendo l'utilizzo di hardware e software standard e *open source*. Il primo protocollo SDN sviluppato è *OpenFlow*.

SDN permette di programmare a tutti i livelli della pila protocollare le logiche di funzionamento della rete in base ai requisiti di amministrazione, non facendo più unicamente affidamento sui protocolli di uso comune. Un altro vantaggio offerto da tale tecnologia consiste in una gestione semplificata della rete da parte dell'amministratore, grazie a strumenti di *orchestration* centralizzati. Per quanto riguarda i produttori di apparecchiature, l'effetto derivante del modello SDN è una semplificazione delle tecnologie di costruzione di router e switch, con un conseguente abbassamento dei costi.

Le tecnologie SDN come OpenFlow [2][3], attraverso la possibilità di programmare a livello software la rete, introducono la gestione della de-materializzazione delle risorse network e di virtualizzazione. Il modello SDN, quindi, può essere visto come un fattore abilitante per la realizzazione di servizi cloud in cui la rete possa rappresentare un elemento di valore aggiunto e ridurne i costi di realizzazione. Inoltre, la gestione di centri di calcolo e di memorizzazione di massa di tipo cloud è un compito estremamente complesso e articolato. Gli effetti della complessità di gestione sull'infrastruttura di rete sono ampi e SDN può essere un modello per l'otti-

Workshop GARR Calcolo e Storage Distribuito - Selected Papers. Roma 29-30 novembre 2012
ISBN: 978-88-905077-4-8 © 2013 Associazione Consortium GARR

mizzazione dell'uso delle risorse e, più in generale, dell'attività di gestione e amministrazione delle reti nei *data center* cloud.

La principale implementazione del paradigma SDN è OpenFlow. Tale protocollo ha un modello di gestione e funzionamento delle apparecchiature di rete che disaccoppia completamente il piano di *forwarding* da quello di controllo. Oltre a essere stato il primo protocollo SDN disponibile, è aperto e permette di gestire insiemi eterogenei di apparati hardware.

Da circa un anno GARR sta svolgendo un'attività finalizzata alla comprensione e alla valutazione delle potenzialità del protocollo OpenFlow. I primi risultati sono stati la creazione di un testbed virtuale dedicato e lo sviluppo di una variante del protocollo *Spanning Tree* (STP) [4].

Gli aspetti descritti saranno discussi in dettaglio nel seguito dell'articolo, che è organizzato come segue: la prima sezione contiene una descrizione più dettagliata del protocollo OpenFlow; la seconda presenta l'architettura del testbed realizzato; la terza sezione presenta il modulo software sviluppato; l'ultima sezione presenta le conclusioni e i possibili sviluppi futuri.

1. Il protocollo OpenFlow

OpenFlow è un protocollo aperto sviluppato dall'università di Stanford a partire dal 2007, che utilizza il concetto di flusso per la re-direzione dei pacchetti. In questo contesto, per flusso s'intende una sequenza unidirezionale di pacchetti aventi caratteristiche comuni, che attraversa il nodo entro un intervallo temporale, avendo sorgente e destinazione fisse. Attualmente la tecnologia OpenFlow permette di definire un flusso utilizzando caratteristiche dei pacchetti dal livello 2 al livello 4 (compresi) della pila protocollare e supporta Ethernet, IP, TCP e UDP. OpenFlow permette di definire, attraverso l'uso di una maschera, quali siano i campi nell'header del pacchetto in base ai quali esso possa essere considerato parte di uno specifico flusso.

OpenFlow gestisce apparecchiature di *switching* e routing disaccoppiando il piano di controllo da quello di *forwarding*. Infatti, mentre nei router e negli switch standard il piano di forwarding e quello di controllo sono presenti nello stesso apparato, gli switch OpenFlow separano le due funzioni. La parte di *datapath (*chiamata anche di *forwarding* o *switching*) è svolta sugli apparati, mentre quella di controllo è spostata su un controller esterno, tipicamente un server.

Rispetto ai classici dispositivi con CPU e logica integrate, switch, router e access point possono essere così gestiti da un piano di controllo distinto, grazie a uno o più *Network Operating System* (NOS) comuni. Agendo attivamente su questi ultimi si potrà decidere come i flussi siano elaborati e instradati all'interno dell'intera rete.

Il supporto a OpenFlow è già disponibile per diverse apparecchiature di rete, quali switch, router e access point WiFi. Tali dispositivi espongono un'interfaccia standard OpenFlow che permette di interagire con i controller esterni, senza bisogno che i diversi produttori rivelino i dettagli dei loro apparati di rete. OpenFlow è implementato in hardware da una gran parte di essi. Il datapath di un apparato OpenFlow utilizza una tabella di flussi memorizzati; ogni record della tabella contiene una serie di regole che permettono di filtrare determinati pacchetti, quindi i flussi, e applicare a essi un'azione del tipo *send-out-port*, *modify-field* o *drop*. Quando uno switch OpenFlow riceve un pacchetto che non ha mai visto prima, per il quale non vi siano regole di pattern matching attive nella tabella dei flussi, esso manda il pacchetto al *controller,* che decide come gestirlo: può ad esempio scartarlo o aggiungere invece un nuovo record contenente una regola alla tabella dello switch, istruendolo così su quali azioni applicare a pacchetti analoghi in futuro. Secondo le configurazioni scelte sul controller, una regola d'indirizzamento può essere installata in modo proattivo, cioè installata a prescindere dal verificarsi di un evento o in seguito a un evento. Inoltre, ogni regola inserita in tabella può scadere dopo un certo intervallo temporale o essere persistente fino allo spegnimento del nodo.

2. Il testbed virtuale GARR

Da circa un anno si sta svolgendo un'attività finalizzata alla comprensione e alla valutazione delle potenzialità del protocollo OpenFlow. A tale scopo è stato realizzato un testbed virtuale, su un server che utilizza VMware ESXi 5.1 [5], capace di emulare in modo realistico una rete gestita da un sistema OpenFlow.

Nell'infrastruttura sono presenti:

- tre macchine virtuali per la simulazione del traffico utente
- quattro switch virtuali basati su OpenFlow
- due router software per consentire l'accesso a Internet, sia agli *host*, sia agli apparati stessi per manutenzione.

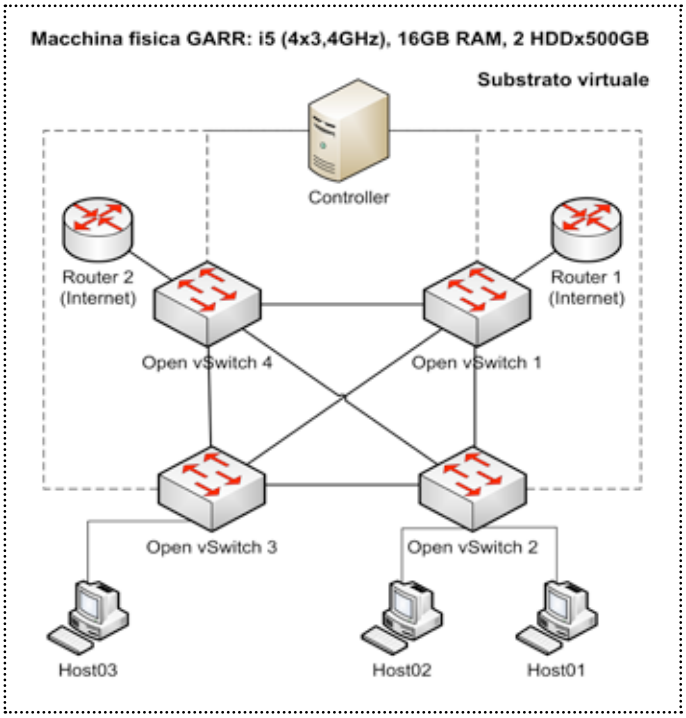


Fig 1 - Schema del testbed virtuale

Il testbed è formato da due piani coesistenti: quello dedicato al traffico utente (rappresentato con linee continue nel diagramma) e quello di monitoring e controllo (linee tratteggiate) dedicato alla comunicazione tra gli apparati OpenFlow e il *controller*.

Si è deciso di focalizzare la ricerca e le prove su protocolli di gestione, per verificare le funzionalità, evitando di tenere conto dei livelli di prestazioni massime raggiungibili, non essendo gli switch in hardware.

Il collegamento tra le porte di rete delle apparecchiature del testbed virtuale è realizzato per mezzo di switch software dell'*hypervisor* con due porte e in "*promiscuous mode*", equivalenti a dei comuni repeater e non partecipanti a OpenFlow.

OpenFlow è implementato da quattro switch emulati da macchine virtuali che eseguono il sistema Open vSwitch e collegati tra loro attraverso una magliatura completa. I possibili cammini tra gli switch sono ridondati per la sperimentazione di meccanismi di *fail-over*, equivalenti di STP, offerti dal protocollo OpenFlow. Ogni switch ha un certo numero di porte, dalle quali transita il traffico utente, collegate ai router, agli host e agli altri switch; inoltre ogni switch OpenFlow ha un'interfaccia di *loopback* per la configurazione e il controllo remoto, la comunicazione con i controller e il collegamento a internet per gli eventuali aggiornamenti.

Nella scelta del controller OpenFlow, si è adottato inizialmente Beacon [6] per le sue grandi potenzialità, affidabilità e diffusione. In seguito, l'attività è stata impostata su FloodLight [7], nato come evoluzione di Beacon e seguito da una comunità utente più ampia e rilasciato sotto licenza Apache2 [8].

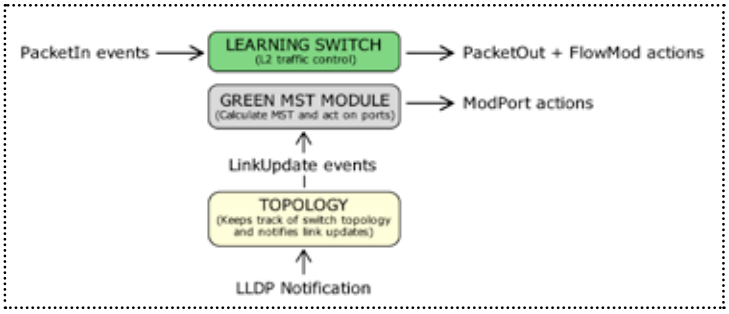


Fig 2 - Il funzionamento dei moduli

3. Sviluppo del modulo GreenMST

Prima di parlare dell'implementa-

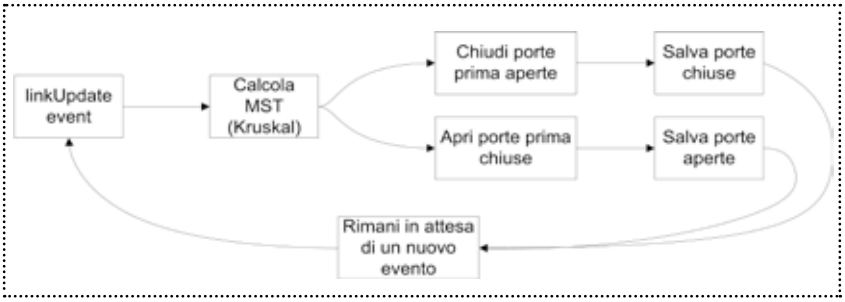


Fig 3 - L'algoritmo Kruskal

zione del modulo *Green Minimum Spanning Tree* (GreenMST) è necessaria una premessa. I controller OpenFlow prevedono la possibilità di usare due moduli diversi per il forwarding dei pacchetti, chiamati *Forwarding* e *LearningSwitch*.
Il primo, più completo e allo stesso tempo più complesso da programmare, prevede l'uso di grafi per astrarre la topologia di rete sul controller centralizzato e supporta l'uso di topologie magliate. Infatti, Forwarding crea un albero di copertura minima per ogni nodo della rete e fa sì che le azioni di *floodling* avvengano solo attraverso le porte appartenenti all'albero, evitando il cosiddetto fenomeno del *broadcast storm*. *Forwarding* è il modulo maggiormente usato e per questo motivo non è generalmente necessario includere nel controller ulteriori tecnologie cosiddette *loop-free*.

Al contrario di Forwarding, LearningSwitch è un modulo semplificato, di facile comprensione, semplice da utilizzare e debuggare e facilmente integrabile con altri moduli, che emula il comportamento dei tradizionali switch di livello 2. Tale modulo, generalmente preferito dagli utenti per la sua praticità, permette di sperimentare una buona parte delle potenzialità offerte da OpenFlow, ma non supporta in modo nativo l'uso di topologie magliate. Quindi, anche se il protocollo Openflow prevede la possibilità di gestire un insieme di switch che utilizzano fra loro il protocollo di *Spanning Tree*, non esiste alcun modulo sui controller in grado di interpretare i messaggi *Spanning Tree*, e i produttori di hardware hanno scelto di non implementarlo, considerando che la maggior parte degli utenti usi il modulo di Forwarding.

Per lo sviluppo del testbed virtuale è stato scelto Learning Switch. Esso mantiene centralmente sul controller le tabelle MAC, tradizionalmente distribuite in locale sugli switch. Analogamente a un comune switch, quando un pacchetto in ingresso arriva al controller, esso memorizza l'associazione tra lo switch di provenienza, la porta d'ingresso e l'indirizzo MAC, poi fa il forwarding del pacchetto al destinatario. Per gestire le topologie magliate è stato sviluppato un nuovo modulo per controller OpenFlow, che fosse in grado di gestire reti con loop a livello 2.

GreenMST è in grado di ricevere le notifiche di alterazione della topologia, dette di *LinkUpdate* (figura a sinistra) emanate dal modulo *Topology* sottostante. A sua volta, Topology intercetta i messaggi di *Link Layer Discovery Protocol* (LLDP) [9] provenienti dagli switch per ricostruire la topologia di rete. A ogni nuovo evento, GreenMST si occupa di calcolare il *minimum spanning tree* del nuovo grafo di rete tramite l'algoritmo di Kruskal [10]. L'albero dei cammini minimi ottenuto, unico per tutta la rete, è usato per chiudere le porte coinvolte nei collegamenti non appartenenti ad esso. Le interfacce escluse sono disattivate attraverso comandi di OpenFlow (PortMod) inviati dal controller agli switch. I cammini attivi disponibili tra tutti i nodi non sono necessariamente ottimali. Tale garanzia si ha solo tra il nodo scelto dall'algoritmo di Kruskal come radice dell'albero e tutti gli altri nodi della rete. L'algoritmo sceglie infatti tra tutti gli alberi possibili quello che minimizza la somma di tutti i cammini per questo albero di copertura minimo.

Per ottimizzare la segnalazione è stata creata un'ulteriore struttura dati che contiene lo stato attuale di tutte le porte degli switch. Al termine del calcolo dello spanning tree, la struttura dati è utilizzata per aprire o chiudere le porte solo quando necessario, minimizzando i messaggi in-

viati serialmente agli apparati.

A ogni link della rete è stato associato un peso statico pari ad uno: in questo modo l'algoritmo privilegia un cammino rispetto a un altro in base al numero di nodi che intercorre tra la radice dell'albero e un nodo di destinazione.

Al momento, il modulo Beacon è disponibile in modalità open source con licenza GPL2 e si sta lavorando per migliorarne i tempi di convergenza e la scalabilità per reti di grandi dimensioni. Presto sarà disponibile anche il modulo per Floodligth con licenza Apache2.

4. Interesse per Cloud

Il paradigma SDN descritto nell'articolo è un elemento abilitante per la realizzazione di soluzioni cloud più mature e articolate. SDN, infatti, introduce due benefici di grande impatto e potenzialmente dirompenti nell'ambito della gestione delle infrastrutture e dei servizi informatici:

1. Da un lato, l'obiettivo specifico di SDN, come dice il nome stesso, è quello di permettere la programmabilità via software di tutti gli apparati di networking. Questa caratteristica permette di sostituire o aggiornare i protocolli standard d'instradamento, adottando quelli più adatti ai centri calcolo.
2. Dall'altro, un controller centralizzato permette di avere un unico punto in cui concentrare la logica di programmazione dell'intera rete, basata sull'analisi per pacchetto dei flussi.

Il primo aspetto permette di superare la classica divisione a strati dei protocolli di rete e ottimizzare l'utilizzo della capacità disponibile, creando nuove architetture di funzionamento dei data center. Tra questi, i data center per le cloud possono beneficiarne, ad esempio ottimizzando il collegamento definito dinamicamente fra i nodi di calcolo e i servizi di storage (ad esempio SAN, iSCSI, AoE).

Grazie alla realizzazione di controller centralizzati è inoltre possibile introdurre un cambio di prospettiva radicale nell'approccio alle applicazioni che fanno un uso intensivo della rete. Ora, infatti, le applicazioni devono adattarsi alle caratteristiche della rete. Con i paradigmi SDN con controller centralizzato come OpenFlow, la rete può essere invece riconfigurata totalmente via software in funzione di quale applicazione genera i flussi di traffico. Questo permette di avere un adattamento programmabile istantaneo che sia in grado di soddisfare le richieste del servizio attivo. Il protocollo OpenFlow si sta evolvendo velocemente, sia al suo interno, sia con compatibilità verso MPLS e altri protocolli. Ai data center il protocollo OpenFlow può offrire una flessibilità di utilizzo delle risorse di rete a circuiti e un'integrazione dinamica con i servizi che non è possibile ottenere con i protocolli precedenti.

Il principale elemento innovativo di SDN risiede nell'aver proposto un paradigma in grado di spostare gli aspetti di configurazione e amministrazione delle reti nel dominio dell'ingegneria software.

Il protocollo OpenFlow e le implementazioni di controller sperimentate hanno comunque ancora limiti rilevanti. Va ad esempio sottolineato che, a oggi, OpenFlow non dispone di estensioni che permettano una comunicazione fra domini diversi. Grazie alle basi poste dal nuovo paradigma SDN è tuttavia possibile immaginare che diverse soluzioni già sperimentate in altri campi informatici possano trovare rapida applicazione anche nel dominio delle reti. Questo è reso possibile dal fatto di aver definitivamente spostato il focus all'interno di architetture software che, come tali, possono essere fatte evolvere in modo più flessibile e rapido.

5. Conclusioni e sviluppi futuri

L'interesse del paradigma SDN e del protocollo OpenFlow consiste nell'apertura delle apparecchiature di rete a un completo controllo dell'utente e alle conseguenti infinite possibilità di composizione di regole e azioni sul traffico. Tali caratteristiche lo rendono particolarmente interessante per i centri di calcolo e i servizi di tipo cloud, che si basano su un'alta densità di apparecchiature e rete trasmissiva, permettendo notevoli ottimizzazioni dell'infrastruttura e del suo

funzionamento.

L'esperienza fatta nel testbed virtuale ha confermato le potenzialità di SDN e del protocollo OpenFlow. Ha permesso di sviluppare in tempi relativamente brevi nuovi algoritmi d'instradamento con funzionalità aggiuntive a quelle classiche. L'innovazione nel campo dei protocolli delle reti locali di tipo Ethernet è un campo attivo della ricerca nelle reti, a cui SDN può dare strumenti nuovi.

Il modulo creato permette di sfruttare le potenzialità offerte da LearningSwitch anche in topologie magliate. Allo stesso tempo, spegnendo le interfacce fisiche degli apparati si può realizzare, in linea di principio, un risparmio energetico per i datacenter che ne faranno uso. Nella progettazione del modulo si è compreso come si possa pensare di usare concretamente parametri aggiuntivi per l'instradamento dei pacchetti. Si potrebbe assegnare dinamicamente un peso a ogni circuito (quindi a ogni arco del grafo) in base alle caratteristiche delle interfacce e della congestione della rete, come carico del circuito e ritardo di trasmissione banda.

Il passo successivo sarà la progettazione di moduli più complessi e l'analisi dell'efficacia di OpenFlow in un dominio di rete di produzione. Si studieranno le capacità di gestione di un gran numero di apparecchiature di livello 2 e livello 3, definendo policy multi-livello, quali ad esempio *routing*, *firewall* e *traffic engineering*.
Il protocollo OpenFlow è in continua evoluzione: ben presto sarà disponibile una nuova versione capace di supportare IPv6 e livello ottico (con controllo delle lunghezze d'onda), oltre al *tagging* dei pacchetti per il supporto a MPLS e QinQ.

Lo sviluppo e la gestione di tale ambiente restano per ora ancora troppo complessi per un uso diffuso. Il familiarizzarsi con l'architettura OpenFlow e la realizzazione dei moduli per i controller sono infatti operazioni onerose che richiedono per ora la collaborazione di personale qualificato ed esperto sia nella programmazione che nel networking.

Riferimenti bibliografici

[1] Nick McKeown, "Software-defined Networking", Infocom Keynote Talk, April 21st 2009, Rio de Janeiro, Brazil - disponibile ad http://tiny-tera.stanford.edu/~nickm/talks.html

[2] OpenFlow white paper: www.OpenFlow.org/documents/OpenFlow-wp-latest.pdf

[3] Wiki OpenFlow: http://www.OpenFlow.org/wk/index.php

[4] http://tools.ietf.org/html/rfc4318

[5] http://www.vmware.com/products/vsphere-hypervisor/overview.html

[6] Beacon controller: https://OpenFlow.stanford.edu/display/Beacon/Home

[7] FloodLight controller: http://floodlight.OpenFlowhub.org

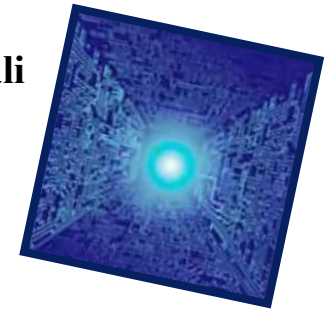
[8] http://www.apache.org/licenses/LICENSE-2.0.html

[9] http://standards.ieee.org/getieee802/download/ 802.1AB-2009.pdf

[10] Sedgewick, R. and Wayne, K, "Algorithms (4th Edition)", Ed. Addison-Wesley Professional, 624-627, 2011

Mauro Campanella
mauro.campanella@garr.it
Laureato in Fisica a Milano nel 1985. Svolge l'attività per la rete della ricerca italiana (Consortium GARR) in cui ricopre il ruolo di coordinatore dell'attività di ricerca e sviluppo.
Ha sempre collaborato a progetti internazionali. Oltre alla partecipazione alla creazione delle varie generazione della dorsale europea della ricerca (GÉANT) ed ai suoi servizi (premiumIP, AutoBAHN, Perfsonar) è stato coordinatore del progetto FEDERICA di supporto a Future Internet.

Octopus: una cloud self-service di machine virtuali



Antonio Cisternino, Maurizio Davini, Marco Mura

IT Center Università di Pisa

Abstract. Alla base delle infrastrutture Cloud si trovano i sistemi di virtualizzazione: l'allocazione dinamica delle risorse di un cloud richiede una flessibilità che le architetture tradizionali, con un sistema operativo in esecuzione direttamente su una macchina fisica, non sono in grado di fornire. In questo articolo presentiamo Octopus, un gestore di una rete di hypervisor, caratterizzato da un modello basato sull'idea di self-service via Web. Per gestire e coordinare un numero sempre crescente di macchine virtuali, secondo una politica di allocazione delle risorse, Octopus usa il sistema esperto CLIPS per definire in modo generale le regole che ne governano il comportamento.

1. Introduzione

Uno dei cardini del Cloud computing [1] è indubbiamente la capacità di virtualizzare le risorse in modo da favorire un disaccoppiamento tra l'erogazione di servizi e l'infrastruttura IT che li offre; le tecnologie di virtualizzazione svolgono quindi un ruolo importante nella realizzazione di Cloud pubbliche e private, e le macchine virtuali sono rapidamente divenute l'unità di allocazione di risorse virtuali alla base del cosiddetto modello IaaS (*Infrastructure as a Service*), a sua volta considerato alla base di modelli più articolati come PaaS (*Platform as a Service*) e SaaS (*Software as a Service*).

Le macchine virtuali sembrano offrire un'interfaccia ideale tra chi vuole offrire un'infrastruttura IT e i suoi utenti, poiché possono essere confinate garantendo la flessibilità necessaria tipica di un PC basato sull'ormai consolidata architettura x86. Tra le public cloud più di successo sicuramente si può menzionare quello di Amazon [2] che offre macchine virtuali piuttosto che una piattaforma per lo sviluppo di applicazioni, come invece accade per Google Apps Engine [3]. Anche Azure [4], la public Cloud di Microsoft, era partita con un modello analogo a quello del Cloud di Google ma ha poi introdotto il cosiddetto VMRole, che consente di allocare macchine virtuali ad uso generale.

Il progetto Octopus nasce nel 2010 con l'idea di realizzare un sistema capace di orchestrare una rete di *hypervisor* con un duplice obiettivo: consentire un modello self-service di provisioning di macchine virtuali attraverso il Web e orchestrare le macchine virtuali, cercando di ottimizzare il consumo energetico con lo spostamento delle macchine virtuali in modo da liberare nodi fisici che possano essere spenti. Il sistema si è poi evoluto per investigare come generalizzare il sistema di regole necessarie a gestire i vincoli nell'allocazione delle risorse e la loro gestione ottimale attraverso l'introduzione di un sistema esperto.

In questo articolo illustreremo l'architettura di Octopus originale e come l'introduzione del sistema esperto CLIPS [5] ne abbia condizionato la struttura e la scalabilità.

2. Octopus

Octopus ha l'architettura mostrata in Figura 1: il sistema si occupa di organizzare e amministrare macchine virtuali in esecuzione su uno o più hypervisor Hyper-V di Microsoft; gli utenti possono creare VM ed amministrare quelle già create attraverso un'interfaccia Web (vedi Figura 2). L'accesso a una macchina virtuale avviene attraverso il portale Web che consente di collegarsi a macchine Unix mediante SSH e a macchine Windows utilizzando il protocollo RDP.

Il sistema è programmato in F# e controlla gli

Workshop GARR Calcolo e Storage Distribuito - Selected Papers. Roma 29-30 novembre 2012
ISBN: 978-88-905077-4-8 © 2013 Associazione Consortium GARR

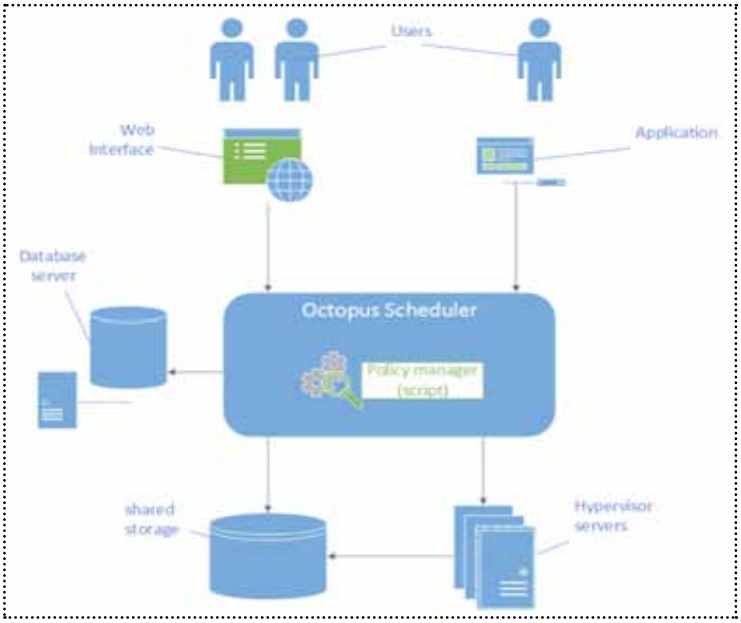


Fig 1 - Architettura di Octopus

Hyper-V utilizzando l'interfaccia WMI. La scelta di utilizzare lo stack Microsoft per la virtualizzazione è dovuta alla semplicità con cui Hyper-V si può controllare da programma e ha permesso di concentrare gli sforzi sulla realizzazione del sistema. Il linguaggio F# è Open Source e può essere eseguito su qualunque piattaforma su cui sia disponibile mono o .NET: è quindi possibile generalizzare il sistema ad altri hypervisor, mediante la sostituzione delle chiamate WMI con script che offrano funzionalità equivalenti.

Il cuore del sistema è il policy manager, ovvero un modulo responsabile di decidere dove allocare una nuova macchina virtuale, la quantità di risorse che possono essere dedicate ad un particolare utente, e in quali condizioni sospendere oppure spostare macchine virtuali da un hypervisor ad un altro della rete. Come già detto questo elemento era stato concepito con particolare attenzione agli aspetti di *green computing*, cercando quindi di distribuire il carico in modo da poter mettere in *standby* i nodi non necessari in un certo momento.

Il gestore del sistema Octopus esercita quindi il proprio controllo sulle politiche per l'allocazione delle risorse attraverso la definizione di due meccanismi essenziali:

- Un insieme di regole
- Un *repository* di immagini delle VM che gli utenti possono usare per creare nuove macchine virtuali.

Un'altra funzione esercitata da Octopus è l'organizzazione dei dischi delle VM utilizzando la tecnica dei *differencing disks*, cioè la creazione di nuovi dischi virtuali da associare a una VM utilizzando un'immagine già pronta come base. Il database di Octopus mantiene tutte le informazioni necessarie alla configurazione e al processo di orchestrazione effettuato dal gestore delle politiche del sistema.

3. Orchestrazione mediante un sistema esperto

La sperimentazione con una prima versione del sistema Octopus ha fatto emergere quali fossero le esigenze di un sistema di amministrazione di Cloud di virtual machine. Presto ci si è resi conto che per poter esprimere regole complesse occorreva riprogettare il modello per la definizione del gesto-

Fig 2 - L'interfaccia Web con le macchine virtuali create da un utente

re di politiche.

L'esplosione del numero di elementi da gestire in un'infrastruttura IT, anche a causa della virtualizzazione delle risorse, rende necessario l'impiego di automatismi sempre più sofisticati, ed è necessario riuscire a catturare la conoscenza di un IT manager per incorporarla in un sistema automatico di cui si possa controllare il funzionamento. Per questo motivo abbiamo deciso di integrare un vero e proprio sistema esperto in Octopus piuttosto che continuare lo sviluppo di un sistema a regole ad hoc la cui struttura era destinata a complicarsi, sfruttando soluzioni sviluppate nel campo dei sistemi esperti. Anche sistemi che gestiscono hypervisor, come ad esempio System Center 2012 di Microsoft, includono soluzioni basate su orchestrazione di processo, simili a quelle dei sistemi esperti che però non definiscono il comportamento in caso di conflitto tra più regole. La possibilità di gestire conflitti con opportune politiche, ad esempio decidendo, quali regole abbiano priorità, consente di esprimere in modo semplice politiche articolate.

Al crescere dei moduli di un'infrastruttura IT, e di conseguenza delle regole per la loro gestione, diviene sempre più difficile assicurarsi che un insieme di regole sia consistente; inoltre questi moduli risultano spesso difficili da estendere, perché sviluppati come interpreti che crescono progressivamente per estendere l'insieme di funzionalità supportate.

I sistemi esperti consentono agli specialisti del dominio di codificare la propria conoscenza specifica, che poi un sistema automatico provvederà ad impiegare per operare. I sistemi esperti sono stati studiati nell'ambito dei sistemi intelligenti ed è presente molta letteratura sul tema, prevalentemente sviluppata durante gli anni '80.

4. CLIPS

Un famoso sistema esperto, sviluppato alla NASA ed utilizzato in molti progetti, è CLIPS, un sistema a regole basato su un'implementazione efficiente dell'algoritmo RETE, la cui sintassi è un dialetto del linguaggio di programmazione LISP.

CLIPS, scritto in C, è progettato per essere incluso in applicazioni, ed esistono numerosi wrapper che consentono di includerlo in ambienti di programmazione moderni come ad esempio .NET. Il sistema consente di definire regole che vengono attivate da fatti espressi con asserzioni che rappresentano una data realtà. Ad esempio si può asserire che una certa VM è in stato di idle:

```
(assert (octopus-vm-idle "MyVM"))
```

La seguente regola definisce che quando una macchina virtuale sia in stato idle (qualunque cosa questo significhi) va sospesa la sua esecuzione:

```
(defrule suspend-when-idle (octopus-vm-idle ?x) =>
    (if (octopus-suspend-vm ?x) then
        (printout t ?x " suspended") else
        (printout t ?x " failed to suspend")))
```

Con l'introduzione di CLIPS in Octopus è il motore del sistema ad asserire automaticamente lo stato sotto forma di fatti in CLIPS. Le regole, predefinite o definite dall'amministratore del sistema, sono quindi attivate automaticamente da CLIPS e possono invocare azioni che Octopus espone come funzioni CLIPS. L'ambiente consente anche l'interazione con CLIPS attraverso un editor guidato dalla sintassi, che mostra le funzioni disponibili con l'ormai familiare approccio del completamento delle possibili voci mentre si scrive.

Mediante regole CLIPS, Octopus controlla i seguenti aspetti:

- Politiche di allocazione delle risorse agli utenti
- Monitoring dello stato delle macchine gestite
- Spostamento delle VM e spegnimento nodi per ottimizzare l'assorbimento energetico

5. Usare il sistema

Una volta installato Octopus su un nodo di un dominio Windows, è necessario allocare uno o più nodi dotati di hyper-V (disponibile anche nella versione desktop del sistema operativo a

partire da Windows 8). Vanno poi preparate le immagini di dischi virtuali, che saranno utilizzati come base per istanziare le macchine virtuali: il sistema supporta sia Windows che Linux.

Una volta configurato, il sistema accetta richieste attraverso il Web applicando le politiche definite nel file di configurazione del sistema e che contiene le regole scritte nel linguaggio CLIPS. Il manager IT può osservare il funzionamento interagendo in modo interattivo con la console CLIPS; può inoltre aggiungere regole anche temporaneamente per modificare il comportamento del sistema.

6. Conclusioni e sviluppi futuri

Octopus è un progetto open source (http://octopus.codeplex.com) ed è stato usato come campo di prova per studiare l'organizzazione di un meta hypervisor che offra meccanismi di self-service provisioning via Web vincolato da opportune politiche di gestione delle risorse. L'utilizzo di un sistema esperto ha consentito i realizzare un'architettura software del sistema più organica che permette di modellare esplicitamente la conoscenza del dominio di un esperto IT manager.

Il limite principale del sistema, oltre al fatto di essere ancora in uno stato prototipale, è quello di funzionare solo sulla piattaforma Microsoft. Stiamo lavorando per rendere gli hypervisor gestiti un parametro del sistema; l'adozione del sistema esperto in questo caso può favorire la gestione di ambienti eterogenei, in cui alcune operazioni possono essere disponibili solo tra hypervisor omologhi.

Riferimenti bibliografici

[1] Armbrust M., Fox A., Griffith R., Joseph A.D., Katz R., Konwinski A., Lee G., Patterson D., Rabkin A., Stoica I., and Zaharia M.; A view of Cloud Computing, Communications of the ACM, April 2010, 53:04.

[2] Amazon AWS, http://aws.amazon.com

[3] Google App Engine, http://appengine.google.com

[4] Microsoft Azure, http://windowsazure.com

[5] Giarratano J.C., Riley G.; Expert Systems, PWS Publishing Co., Boston, 1998, ISBN: 0534950531

Antonio Cisternino
cisterni@di.unipi.it
E' ricercatore presso il Dipartimento di Informatica dell'Università di Pisa e membro del centro interdipartimentale IT Center. Si occupa di programmazione di sistemi complessi, ambienti di virtualizzazione e user-interaction. È attivo nella comunità .NET a partire dal 2001 e più recentemente è uno dei leader della comunità del linguaggio di programmazione F#.

Maurizio Davini
mau@df.unipi.it
E' il Coordinatore Tecnico del Centro Interdipartimentale IT Center dell'Università di Pisa. Fino al 2012 è stato il responsabile del Centro di Calcolo del Dipartimento di Fisica. Ha fatto parte di numerosi advisory board in ternazionali in ambito HPC, collaborando con aziende quali Ferrari, AMD, Intel, Dell, e HP.

Marco Mura
mura@di.unipi.it
Ha una Laurea Specialistica in Tecnologie informatiche, Octopus è stato l'oggetto del lavoro di tesi. Ha partecipato alle attività dell'IT Center, partecipando con Acer a International Super Computing 2010. Nel 2010 ha svolto una internship presso Microsoft Research (Redmond) lavorando su tematiche relative ai fabric.

Grandi infrastrutture di storage per calcolo ad elevato throughput e Cloud

Michele Di Benedetto, Alessandro Cavalli, Luca dell'Agnello, Matteo Favaro, Daniele Gregori, Michele Pezzi, Andrea Prosperini, Pier Paolo Ricci, Elisabetta Ronchieri, Vladimir Sapunenko, Vincenzo Vagnoni, Valerio Venturi, Giovanni Zizzi

INFN-CNAF

Abstract. Gli esperimenti di fisica delle alte energie che lavorano al Large Hadron Collider hanno accumulato in pochi anni un'enorme quantità di dati, sviluppando soluzioni molto avanzate nella gestione di grandi spazi di storage disco e nastro. Il Tier-1 dell'INFN, presso il CNAF di Bologna, è uno dei maggiori centri di calcolo della collaborazione WLCG, con una capacità di oltre 12 PB di disco e 16 PB di nastro. Grazie ad una combinazione di soluzioni industriali e sviluppi specifici, è attualmente uno dei Tier-1 con le migliori prestazioni. L'esempio del Tier-1 INFN può essere di grande importanza per comunità che hanno necessità simili, e che vogliono utilizzare risorse di storage con paradigmi esistenti, come il Grid Computing o emergenti, come il Cloud Computing. In questo articolo presentiamo l'architettura e le scelte tecnologiche adottate, discutendo anche le evoluzioni più recenti verso un sistema di cloud storage per le comunità scientifiche.

1. Introduzione

Negli ultimi anni, le grandi collaborazioni scientifiche in vari campi di ricerca hanno accumulato una quantità di dati mai raggiunta in precedenza, in alcuni casi fino alla scala di svariate decine di PetaBytes (PB) all'anno. È questo per esempio il caso degli esperimenti di fisica delle alte energie che lavorano al *Large Hadron Collider* (LHC) del CERN. L'esperienza fatta in questo settore può essere di grande importanza per altre comunità che hanno necessità simili, specialmente nell'ottica di sfruttare i data center esistenti per offrire risorse di storage (sia con il paradigma del Grid Computing che del Cloud Computing) a numerosi gruppi di ricerca o comunità scientifiche.

Un *Mass Storage System* (MSS) che offra una soluzione *Hierarchical Storage Manager* (HSM), ovvero che comprenda sia risorse online (immediatamente disponibili, come il disco), sia nearline (disponibili con una latenza maggiore, ma anche con un maggiore grado di resilienza dei dati, come il nastro), e che raggiunga la scala delle decine di PB di spazio disponibile, è un sistema complesso composto da molti *layer* hardware e software, dei quali il livello visibile all'utente (ad es. l'interfaccia di cloud storage) è soltanto la punta dell'iceberg. I componenti di un sistema del genere sono sia hardware - dischi e controller, reti *fibre-channel* per *Storage Area Network* (SAN) e *Tape Area Network* (TAN), interfacce di rete a 10 Gbps, *disk server* che le supportino, tape server, ecc. - sia software - *file system*, software di management delle risorse *tape*, *middleware* di trasferimento file e di *storage management*, interfacce utente.

Il Tier-1 dell'INFN ha sviluppato una soluzione generale per MSS altamente scalabile e resistente. È implementato con un sistema modulare composto da standard industriali: *General Parallel File System* (GPFS[1]) e *Tivoli Storage Manager* (TSM[2]), entrambi prodotti IBM, interfacciati tra loro da un middleware sviluppato dall'INFN, GEMSS[3]. Tra le altre cose, il sistema implementa un modello intelligente per portare online i file da nastro, che riduce al mini-

Workshop GARR Calcolo e Storage Distribuito - Selected Papers. Roma 29-30 novembre 2012
ISBN: 978-88-905077-4-8 © 2013 Associazione Consortium GARR

mo le operazioni meccaniche della robotica, come il montaggio, lo smontaggio e la ricerca. L'esperienza, maturata in diversi anni di produzione ha dimostrato l'efficienza e la completezza di questa soluzione.

L'accesso allo storage avviene mediante protocolli standard, in accordo con la specifica *Storage Resource Manager* (SRM[4]), adottata dalle comunità WLCG[5], e più in generale dalle comunità che utilizzano il Grid Computing. SRM è un livello di astrazione che permette agli utenti di accedere allo storage attraverso un'interfaccia comune. Dietro questo tipo d'interfaccia, ogni data center può fare le proprie scelte per ciò che riguarda le componenti hardware e le soluzioni software per implementare il proprio MSS. In questo contributo ci si propone di dare una panoramica completa di come un'infrastruttura di storage di svariati PB funziona ed è gestita in produzione, dagli strati più bassi del livello hardware alle interfacce software di livello superiore. Saranno presentati anche i principali risultati e dati relativi alle prestazioni ottenute nel corso di questi ultimi anni di attività dalle principali collaborazioni scientifiche che lavorano presso il Tier-1 dell'INFN. La progettazione di un'efficiente, robusta e affidabile istallazione Cloud storage di grandi dimensioni, si basa sulla corretta scelta delle tante componenti coinvolte nel sistema, e su come queste lavorano in cooperazione. Esprimiamo, anche con il presente articolo, il desiderio di condividere la nostra esperienza con altre comunità.

2. Il Tier-1 dell'INFN

Il Tier-1 dell'INFN ospitato presso il CNAF - il centro nazionale dell'INFN per la ricerca e lo sviluppo nel campo delle tecnologie informatiche applicate agli esperimenti di fisica nucleare e delle alte energie - è il principale centro di calcolo dell'INFN, e uno dei più grandi in Europa.
Con una superficie di circa 1000 m2 e un impianto ridondato per la distribuzione elettrica (potenza utile di ~5 MVA), può ospitare più di 120 rack e due librerie di nastri, garantendo operatività agli utenti 24 ore su 24 per tutto l'anno. Attualmente ospita una farm di calcolo con una potenza complessiva di ~135 kHS06 ed una capacità di circa 12 PB di storage disco 16 PB di storage nastro.

Operativo dal 2003, il Tier-1 è parte della collaborazione WLCG (World-wide LHC Computing Grid), che fornisce le infrastrutture di calcolo e di storage per i quattro grandi esperimenti LHC (ALICE[6], ATLAS[7], CMS[8] and LHCb[9]). La frazione delle risorse ospitate al CNAF è circa il 13% del totale disponibile presso tutti i Tier-1 WLCG. Il centro è progressivamente diventato il punto di riferimento per il calcolo di molte altre collaborazioni scientifiche, sia esperimenti presso acceleratori (BABAR[10], CDF[11], AGATA, KLOE, LHCf), sia di fisica delle astroparticelle e raggi cosmici (AMS, ARGO, Auger, Borexino, FERMI/GLAST, Gerda, ICARUS, MAGIC, PAMELA, Xenon100, VIRGO).

All'interno della struttura del Tier1 sono anche ospitati il Tier-2 italiano di LHCb (le risorse in questo caso sono completamente condivise con quelle del Tier-1), ed una farm Tier-3 per gli utenti della locale Sezione dell'INFN. La capacità dello storage, sia su disco che su nastro, verrà ulteriormente incrementata durante il 2013 e negli anni successivi, mentre per le risorse di calcolo è previsto un sostanziale consolidamento sui valori attuali con la sostituzione progressiva, nell'arco di due anni, di un numero di server corrispondente a circa metà della potenza attualmente installata. Questo permetterà, oltre ad un ricambio fisiologico delle risorse, anche l'ottimizzazione dei consumi elettrici.

Nelle sale del Tier-1 è ospitato uno dei nodi più importanti della rete GARR[12]: è stato uno dei primi, nel corso del 2012, a migrare alla nuova infrastruttura basata su fibre spente (GARR-X).

Oltre al normale accesso alla rete della ricerca italiana, che assicura il collegamento alle reti della ricerca europee e mondiali attraverso la rete GÉANT, la connettività con il Tier-0 al CERN e con gli altri centri Tier-1 di WLCG è assicurata dalla rete dedicata LHCOPN, alla quale il CNAF accede con un collegamento ridondato a

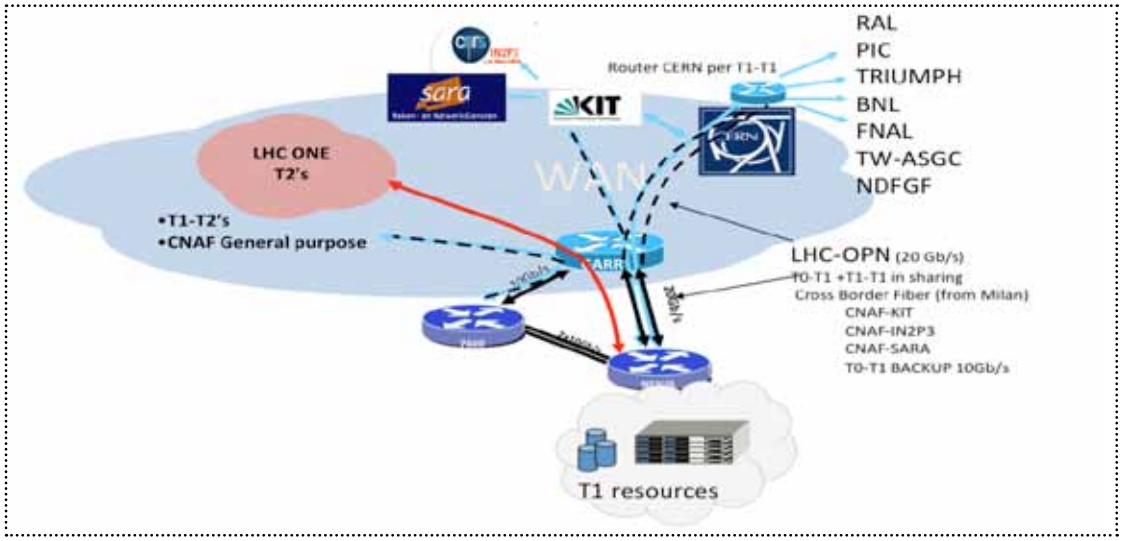


Fig 1 - Collegamenti di rete geografica del Tier-1 dell'INFN

20 Gbps. È inoltre in fase di realizzazione un'ulteriore rete, LHCONE, per l'interconnessione con i principali Tier-2 di WLCG (Fig. 1).

3. La soluzione hardware

Attualmente il CNAF ospita circa 1300 server di calcolo. Le risorse di calcolo vengono allocate dinamicamente ai singoli esperimenti tramite il meccanismo del *fair-share*. La gestione centralizzata permette il pieno utilizzo della farm, che risulta completamente utilizzata per più del 95% del tempo. In Fig. 2 è rappresentata la potenza di calcolo usata negli ultimi 12 mesi: il rapporto tra l'area rossa (tempo effettivo di uso di CPU) e verde (tempo complessivo di impegno delle risorse) è un'indicazione dell'efficienza di uso dell'infrastruttura di storage del centro, in media è piuttosto alta (84% in questo caso)

È bene sottolineare che, essendo la tipologia delle applicazioni assai varia - da programmi di simulazione che sostanzialmente non effettuano accesso allo storage, e quindi sono per definizione ad alta efficienza, a programmi di analisi dati "caotici", per i quali il tempo di accesso allo storage è spesso il fattore dominante - una misura precisa dell'efficienza non può prescindere dalla suddivisione per tipologia di job.

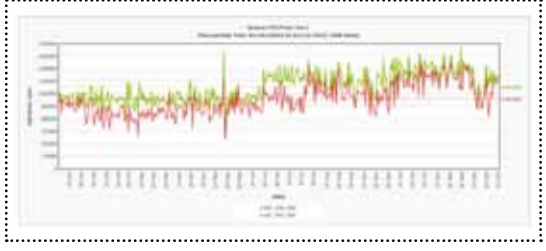


Fig 2 - Uso della farm al Tier-1 dell'INFN

Lo storage al CNAF è organizzato in *file system* GPFS serviti alla farm di calcolo tramite server dedicati, interconnessi alla LAN a 1 Gbps o 10 Gbps, e con 2 link *Fibre Channel* (8 Gbps ciascuno) alla *Storage Area Network* (SAN), cui sono connessi anche i sistemi di disco e (tramite una TAN) i drive della libreria a nastri. L'accesso dai nodi di calcolo ai file system avviene tramite il protocollo file (i file system sono montati sui nodi di calcolo). Le risorse di storage sono anche accessibili da WAN attraverso server GridFTP[13]

I sistemi storage che compongono attualmente la SAN appartengono a più generazioni successive, sia per i collegamenti di rete che per il disco:

- 7 *Data Direct Networks* (DDN) S2A 9950 (con dischi SATA da 2 TB) ed 1 DDN SFA10000 (con dischi SATA da 3 TB), per un totale di ~9 PB serviti da circa 40 disk server con collegamento alla LAN a 10 Gbps;
- 7 EMC2 CX3-80 e 1 EMC2 CX4-960 (con dischi SATA da 1 TB) per un totale di ~2 PB serviti da circa 90 disk server con collegamento alla LAN a 1 Gbps.

Le cassette a nastro sono ospitate su una libreria

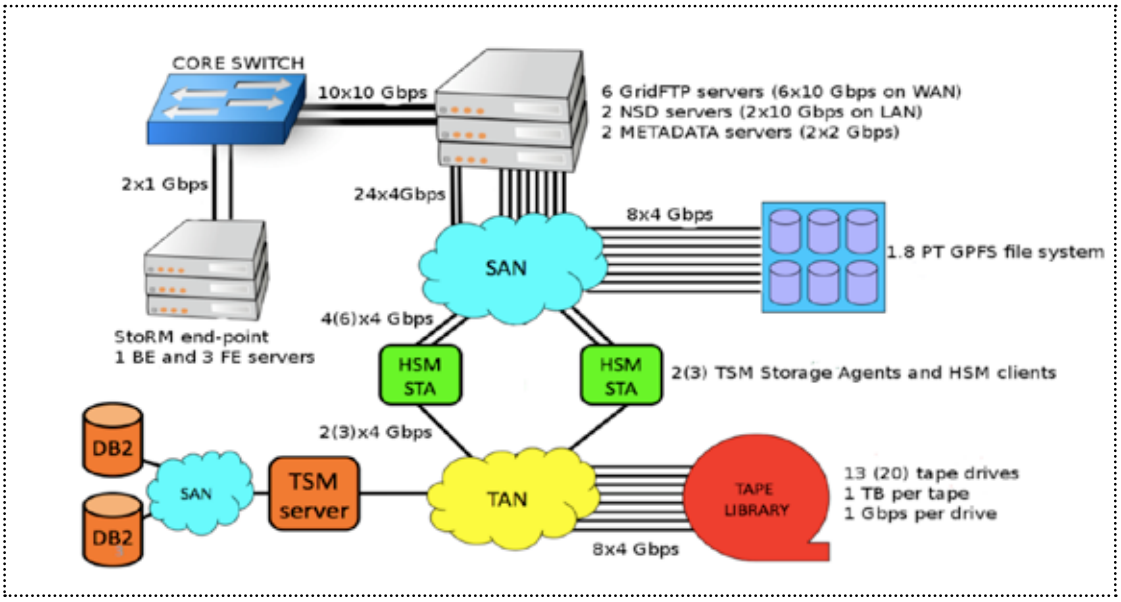


Fig 3 - Schema del sistema di storage per un tipico esperimento del Tier-1 INFN

Oracle SUN SL8500 con 20 drive T10KB (100MB/s di banda passante ciascuno, per ~8500 nastri da 1 TB) e 10 drive T10KC (200MB/s di banda passante ciascuno, per ~1500 nastri da 5 TB). L'interfacciamento fra file system e storage su nastro è realizzato da GEMSS, che implementa un vero e proprio sistema HSM.
In Fig. 3 è illustrato lo schema dello storage per un tipico esperimento al Tier-1.

4. La soluzione software

L'ottimo rendimento del Tier-1 del CNAF è frutto di un'attenta miscela di soluzioni industriali e sviluppi *ad hoc*, che ha consentito di utilizzare la robustezza e l'affidabilità di soluzioni commerciali, nel nostro caso prodotte da IBM, con le interfacce e i paradigmi specifici definiti nella comunità della fisica delle alte energie.

Il file system parallelo GPFS è la soluzione standard adottata dal CNAF per memorizzare i dati su disco. L'utilizzo di file system paralleli consente di avere un namespace unico (a livello di singolo sito, se non utilizzato su WAN) visto da tutti i nodi client come un file system locale. Le varie risorse disco sono aggregate attraverso molti disk server, sui quali i file sono spezzettati in cosiddette stripe di dimensione predefinita. I client possono avere accesso ai dati mediante molti percorsi, garantendo in questo modo la ridondanza e un bilanciamento ottimale del sistema, riducendo la probabilità di avere un collo di bottiglia e aumentando la banda totale di accesso allo storage. I client possono anche accedere simultaneamente in scrittura agli stessi file in modo concorrente, dato che la coerenza globale è garantita dal file system stesso. GPFS è in uso al CNAF da svariati anni. Circa una decina di file system è montata su tutti i nodi di calcolo della farm, per un totale di oltre 10 PB di disco. Un traffico aggregato di circa 10-20 GB/s è raggiunto su base quotidiana.

Il sistema di gestione dei nastri impiegato dal CNAF, il *Tivoli Storage Manager* (TSM), consente un agevole accesso da parte di nodi client a librerie robotizzate, nascondendo tutta la complessità dell'hardware sottostante. TSM è composto da un lato server, che implementa le funzionalità di backup, archiviazione e gestione di uno spazio gerarchico, e da un lato client, dove viene eseguito un piccolo insieme di comandi che consente l'interazione col server e il trasferimento dei dati. Il server memorizza tutte le informazioni su un database DB2, la cui gestione è comunque demandata ai processi server di TSM, ed è completamente nascosta agli amministratori del sistema. Nel nostro caso facciamo u-

so in particolare delle funzionalità HSM del prodotto. I file vengono inizialmente scritti sul file system GPFS, e quindi copiati in background verso i nastri in modo automatico. Il contenuto dei file copiati su nastro viene rimosso da disco dal sistema quando il file system è prossimo al riempimento totale. Nel momento in cui un'applicazione deve accedere ad un file che si trova su nastro ma non più su disco, esistono due diverse modalità di accesso. Una prima modalità consiste nell'accedere direttamente al file come se questo fosse effettivamente disponibile su disco. In questo caso il sistema intercetta la chiamata *read()* e richiama automaticamente su disco il file, mantenendo il client in attesa fino al completamento dell'operazione. La seconda modalità invece passa dall'interfaccia SRM. In questo caso il client richiede con uno specifico comando SRM che il file venga reso disponibile su disco, per procedere all'accesso solo dopo che la procedura di richiamo da nastro è terminata. In ogni caso, tutte le interazioni tra GPFS, TSM e StoRM[14], sono mediate da GEMSS, uno strato software sviluppato dall'INFN. GEMSS consente di aggregare e riordinare in modo dinamico e intelligente le varie richieste concorrenti di accesso ai nastri, in modo tale da avere un ordine di accesso ai file ottimale (i nastri sono dispositivi puramente sequenziali) e così ridurre al minimo le operazioni meccaniche di montaggio, smontaggio e ricerca sui nastri.

Come già anticipato, l'accesso allo storage avviene in accordo con la specifica SRM, adottata dalle comunità WLCG. SRM è un livello di astrazione che permette agli utenti di accedere allo storage attraverso un'interfaccia comune. L'interfaccia web service descritta nelle specifiche SRM fornisce un modo per spostare in modo trasparente i file da e verso la Grid, con libera scelta del protocollo di trasferimento e con un livello ben definito di servizio. SRM fornisce supporto per le più comuni operazioni sui file system, aggiungendo comandi più specifici per il controllo della gestione dello storage. L'interfaccia SRM è nata proprio per garantire l'interoperabilità, in modo che ogni data center possa fare le proprie scelte sul setup del proprio sistema di storage.

Ci sono diverse implementazioni SRM che supportano una varietà di configurazioni di storage. Per sfruttare al meglio la configurazione del MSS scelta per i Tier-1, l'INFN ha sviluppato una propria implementazione dell'interfaccia SRM, StoRM, progettata intorno al principio guida di sfruttare i vantaggi dei cluster file system come GPFS e Lustre[15]. StoRM si integra con GEMSS per garantire la gestione dello storage gerarchico, con la possibilità di trasferire file a e da storage basato su nastri.

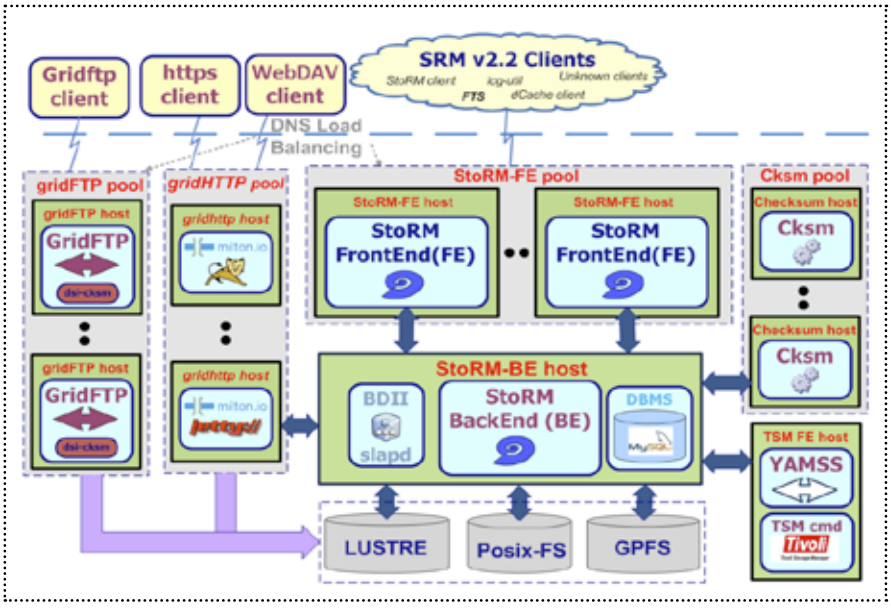


Fig 4 - Schema dell'architettura della SRM StoRM

Nell'ultima versione disponibile, StoRM fornisce una nuova interfaccia che riunisce operazione di storage management e di trasferimento file, in accordo con lo standard WebDAV[16]. Questa interfaccia nasconde i dettagli del protocollo SRM, e permette di

montare storage remoto come una partizione locale, o semplicemente di sfogliare i dati in uno spazio di storage tramite un *browser web*, con o senza autenticazione X509. Uno schema architetturale di StoRM è riportato in Fig. 4.

5. Evoluzioni future

L'interfaccia SRM, che ha servito egregiamente gli esperimenti LHC in questi anni, consentendo l'interoperabilità di centinaia di centri di calcolo con soluzione storage eterogenee, ha però evidenziato alcuni limiti. Il principale è che per centri che non hanno soluzioni di storage HSM, molte operazioni sono superflue e la complessità dell'interfaccia, specialmente per gli utilizzatori, non bilancia i vantaggi.

Una richiesta molto forte della comunità WLCG è quella di federare le risorse di storage, dando la possibilità di accedere alle risorse dei vari centri come se fosse un'unica entità. L'infrastruttura basata su redirector (come quello di xrootd[17] o http) implementa il *failover* nell'accesso a file, redirigendo il client ad una replica disponibile del file cercato.

Negli ultimi anni sono diventate molto diffuse soluzioni di Cloud Storage, come Amazon Web Services S3[18], Google Cloud Storage[19] oppure, con un target diverso, Dropbox[20]. Il superamento dell'interfaccia SRM verso interfacce che abbiano la semplicità delle interfacce di Cloud Storage è una questione di estrema attualità ed importanza. L'offerta di un servizio sullo stile di Dropbox alle comunità scientifiche, che integri la possibilità di utilizzare lo storage per task computazionali, è un'enorme possibilità. A questo scopo, nell'ambito dello sviluppo del prodotto INFN StoRM, è stata introdotta un'interfaccia WebDav, attualmente in fase di deployment. In questo modo, StoRM può integrare in un unico punto di accesso le funzionalità SRM standard, necessarie ad esempio per utilizzare in modo efficiente sistemi gerarchici dotati di risorse nastro, e storage utente o di gruppo, coadiuvato da un'interfaccia client WebDaV. Mediante questa interfaccia è possibile esportare un file system GPFS gestito da StoRM su un qualunque portatile o desktop, utilizzando un client grafico per la gestione, l'upload e il download dei file (si veda Fig. 5).

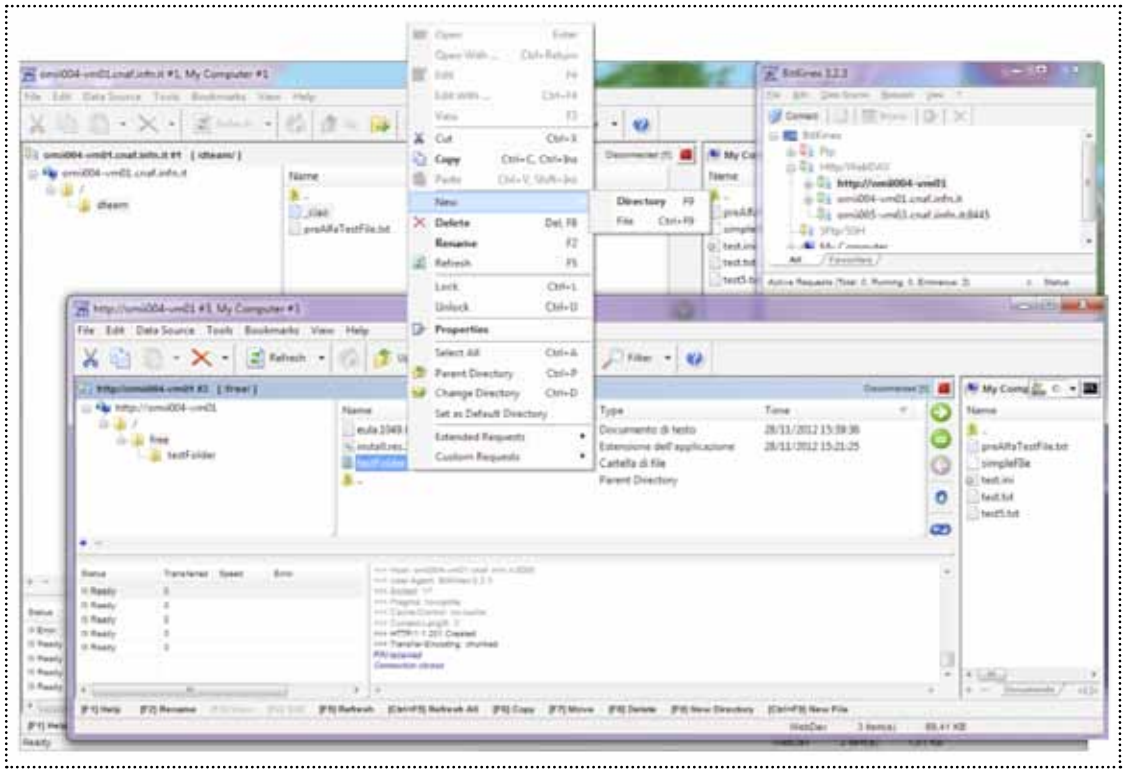


Fig 5 - Esempio di utilizzo dell'interfaccia WebDav in StoRM

6. Conclusioni

In questo lavoro abbiamo presentato l'architettura, le scelte implementative e i risultati del sistema di storage in uso al Tier-1 INFN del CNAF, che offre svariati PB di spazio disco e nastro alla comunità WLCG ed ad altre comunità della fisica delle alte energie. È stata descritta la dotazione hardware, e la scelta di un'integrazione di soluzione industriali e implementazioni ad hoc che hanno reso negli anni il Tier-1 come uno dei centri più affidabili di WLCG, caratterizzato da eccellenti prestazioni nell'ambito della Computing Grid di LHC. Sono state presentate anche le recenti evoluzioni del prodotto StoRM sviluppato dall'INFN, verso un sistema di cloud storage per le comunità scientifiche.

Riferimenti bibliografici

[1] http://www-03.ibm.com/systems/software/gpfs

[2] http://en.wikipedia.org/wiki/IBM_Tivoli_Storage_Manager

[3] https://github.com/italiangrid/gemms

[4] https://sdm.lbl.gov/srm-wg/doc/SRM.v2.2.html

[5] http://wlcg.web.cern.ch

[6] http://aliceinfo.cern.ch

[7] http://atlas.web.cern.ch/Atlas/Collaboration

[8] http://cms.web.cern.ch

[9] http://lhcb.web.cern.ch/lhcb

[10] http://www.slac.stanford.edu/BF

[11] http://www-cdf.fnal.gov/collaboration

[12] http://www.garr.it

[13] http://www.globus.org/toolkit/docs/latest-stable/gridftp

[14] http://storm.forge.cnaf.infn.it

[15] http://wiki.lustre.org/index.php/Main_Page

[16] http://www.webdav.org

[17] http://xrootd.slac.stanford.edu

[18] http://aws.amazon.com/s3

[19] https://cloud.google.com/products/cloud-storage

[20] https://www.dropbox.com

Michele Di Benedetto
michele.dibenedetto@cnaf.infn.it

si è laureato in informatica all'Università di Bologna nel 2008.

Ha lavorato nel privato per un anno prima di essere assunto all'Istituto Nazionale di Fisica Nucleare, dove ha lavorato per 4 anni allo sviluppo di middleware di Grid.

Cloud Computing in ENEA-GRID: Macchine Virtuali, Roaming Profile e Online Storage

Giovanni Ponti, Alessio Rocchi, Antonio Colavincenzo, Gianfilippo Giannini, Alessandro Secco, Giovanni Bracco, Silvio Migliori

ENEA – UTICT (Unità Tecnica per Information and Communication Technology)

Abstract. In questo lavoro, verrà descritta l'infrastruttura di cloud computing in ENEA-GRID che ha come obiettivo quello di fornire macchine virtuali per gli utenti di griglia. Tali macchine sfruttano e si poggiano sull'infrastruttura ENEA-GRID, la quale permette di attuare alcune soluzioni interessanti, quali la gestione dei profile degli utenti, l'integrazione delle macchine virtuali nella cella AFS ENEA e la possibilità di scambiare dati in modo semplice e sicuro utilizzando l'applicativo web OKBox sviluppato in ENEA. Sarà descritta anche l'apposita interfaccia per la gestione delle macchine virtuali integrata nel portale web FARO di accesso in ENEA-GRID e saranno discussi i primi risultati di stress test dell'infrastruttura di cloud.

1. Introduzione

L'ENEA, Agenzia Nazionale per le nuove tecnologie, l'energia e lo sviluppo economico e sostenibile, svolge attività di ricerca in diversi settori, quali efficienza energetica, fonti rinnovabili, fissione e la fusione nucleare, clima e ambiente e nuove tecnologie. Ha 14 sedi e centri di ricerca in territorio italiano, diversi uffici territoriali, un presidio di ricerca in Antartide e una sede di rappresentanza a Bruxelles, per un totale di oltre 2.600 dipendenti.

Per supportare le esigenze dei diversi gruppi di ricerca, è stata progettata e definita un'infrastruttura distribuita che prende il nome di ENEA-GRID [1]. Quest'ultima è nata nel 1998 e ha subito continue migliorie ed evoluzioni, che hanno portato oggi ad ottenere l'integrazione completa delle risorse computazionali, dei sistemi di storage, e dei sistemi di monitoring delle risorse presenti nei 6 centri di calcolo ENEA. Il tutto permette di offrire una capacità di storage integrata di circa 500 TB e una potenza di calcolo integrata di oltre 50 Tflops, la maggior parte della quale è erogata dagli oltre 6.000 core dei sistemi HPC CRESCO [2], che hanno nel Centro Ricerche ENEA di Portici il sito di maggiore importanza.

ENEA-GRID offre una serie di risorse per supportare le attività scientifiche svolte dai ricercatori ENEA e dai loro collaboratori, sia per quanto riguarda applicazioni ad alto grado di parallelismo sia applicazioni seriali che richiedono un massivo utilizzo delle risorse. ENEA-GRID utilizza componenti software maturi, quali LSF per la gestione delle risorse, i due *file system* OpenAFS (geograficamente distribuito) a GPFS (per il calcolo parallelo), il sistema di autenticazione Kerberos 5, e il sistema di monitoring Zabbix. Inoltre, ENEA-GRID offre all'utente diversi modi di accesso, che vanno dalla console remota SSH a interfacce grafiche *user friendly*, quali il desktop remoto attraverso NX e il portale web FARO [3].

In questo lavoro vogliamo fornire una panoramica dell'esperienza di Cloud Computing in ENEA-GRID, descrivendo come tali funzionalità si sono evolute nel corso degli anni fino ad arrivare al contributo principale qui presentato, vale a dire l'offerta di macchine virtuali per gli utenti di ENEA-GRID attraverso la piattaforma di cloud OpenNebula [4]. Queste macchine virtuali sono on-demand, istanziabili su richiesta degli utenti da un set predefinito di template. La ge-

Workshop GARR Calcolo e Storage Distribuito - Selected Papers. Roma 29-30 novembre 2012
ISBN: 978-88-905077-4-8

stione di tali macchine è fatta in maniera efficiente mediante il salvataggio delle personalizzazioni utente (*roaming profile*), e questo meccanismo permette di gestire le risorse in maniera efficiente in quanto, una volta terminato l'utilizzo, le macchine virtuali possono essere distrutte per liberare le risorse. Inoltre, lo scambio dei dati tra la macchina virtuale e la macchina fisica dell'utente avviene tramite un sistema di storage online implementato in ENEA che prende il nome di OKBox.

L'articolo è organizzato nel seguente modo: verrà fatta una panoramica delle funzionalità di cloud computing disponibili in modalità nativa in ENEA-GRID, per poi entrare più nel dettaglio su due diverse tipologie di sistemi di cloud presenti nella griglia. Saranno quindi descritti i sistemi di cloud per servizi virtualizzati, per poi proseguire con l'obiettivo principale di questo lavoro, ossia i servizi di cloud per erogare macchine virtuali per gli utenti. Per queste ultime, sarà descritto il modo in cui salvare le personalizzazioni utente (*roaming profile*) e come utilizzare i servizio di storage online per lo scambio dei dati (ENEA OKBox). Infine, saranno discussi i risultati di un primo stress test effettuato sull'infrastruttura.

2. Cloud computing in ENEA-GRID

ENEA-GRID esporta intrinsecamente e fin dalla sua nascita funzionalità oggi peculiari dei sistemi di cloud computing. In particolare, si fa riferimento allo storage distribuito messo a disposizione da OpenAFS, il quale permette di accedere ai propri dati in griglia in maniera indipendente dalla postazione dalla quale ci si trova. Dal punto di vista delle applicazioni, ENEA-GRID è nata per l'utilizzo di software remoti e, in questa direzione, ha subito delle evoluzioni che hanno portato alla recente definizione dei Laboratori Virtuali [5]: aree accessibili via web che offrono un ambiente di servizi di utilità per una particolare area tematica e supportano utenti che vogliono collaborare in alcuni scenari specifici, permettendo la condivisione di documenti, opinioni, e l'accesso a software dedicati.

Vediamo ora quali sono stati i passi successivi che hanno permesso di estendere le funzionalità di ENEA-GRID in senso *cloud-oriented*. In particolare, faremo riferimento alla virtualizzazione di servizi e alle macchine virtuali per gli utenti.

2.1 Cloud per servizi virtualizzati

I servizi ICT virtualizzati di ENEA hanno permesso di ampliare l'offerta di funzionalità per gli utenti. Tali servizi possono consistere nell'accesso ad applicativi specifici o ad interi sistemi customizzati secondo le richieste specifiche che vengono forniti da centri di calcolo di taglia importante, capaci di rilevanti economie di scala, attraverso in generale l'utilizzo di risorse di calcolo virtualizzate. Per erogare questi servizi sono utilizzati prodotti stabili e consolidati, che si basano sulla piattaforma VMware e garantiscono soluzioni efficienti e affidabili per servizi critici.

Questo sistema permette di far fronte a richieste specifiche di gruppi di lavoro e di esigenze di progetti di ricerca senza tuttavia stravolgere l'infrastruttura fisica di ENEA-GRID, sia dal punto di vista hardware che software. In questo modo, i due concetti di Grid Computing e di Cloud Computing possono integrarsi all'interno di un'infrastruttura come ENEA-GRID che, pur essendo distribuita geograficamente, presenta all'utente un ambiente unificato ed integrato.

2.2 Cloud per macchine virtuali utente

In questa sezione, sarà illustrato l'obiettivo principale di questo articolo, che consiste appunto nel descrivere tutte le attività che hanno portato ad erogare macchine virtuali per gli utenti di ENEA-GRID. In particolare, tutto è nato da una sperimentazione iniziata negli ultimi mesi del 2011, all'interno della quale è stata testata una prima installazione di OpenNebula all'interno di ENEA-GRID per erogare macchine virtuali [6].

Verificata la sostenibilità di questa sperimentazione sia dal punto di vista architetturale che software, il passo successivo ha portato alla definizione di una serie di template di macchine virtuali che gli utenti posso istanziare in base alle proprie necessità. Si noti che queste macchine virtuali sono "*on demand*" e, tipicamente,

hanno un ciclo di vita limitato alla singola sessione di utilizzo da parte dell'utente, il quale, una volta terminato il proprio task, può distruggerle (disposable VM) e rilasciare in questo modo le risorse occupate.

Tuttavia, sebbene tali macchine siano "non persistenti" (i.e., nessun cambiamento viene salvato sul disco virtuale), abbiamo pensato di adottare la gestione e il salvataggio delle personalizzazioni utente salvando il suo profilo (meglio noto in ambiente Windows come "*roaming profile*"). In questo modo, quando l'utente chiederà nuovamente di istanziare un particolare tipo di macchina virtuale, saranno contestualmente caricati il suo ambiente desktop e le sue precedenti personalizzazioni.

La gestione del profilo utente permette di definire dei template comuni per più utenti, e di avere una gestione efficiente e ottimizzata delle risorse fisiche dell'infrastruttura di cloud. Inoltre, questi profili vengono gestiti sia per macchine Windows che per macchine Linux. È opportuno sottolineare che tale gestione è state resa possibile grazie all'integrazione delle macchine virtuali nella cella AFS di ENEA, che permette di montare la home utente AFS nelle macchine virtuali e di accedere ai propri dati in ENEA-GRID. Per di più, l'integrazione in AFS ha evitato di creare utenze locali sulle macchine virtuali poiché l'autenticazione avviene in maniera integrata, basandosi sull'accesso ad una *active directory* per le macchine virtuali Windows e su Kerberos per quelle Linux, rendendo quindi i template ancora più generali e riutilizzabili per tutti gli utenti di griglia.

Per una migliore gestione dei template e delle risorse virtuali, sono state definite all'interno di OpenNebula delle apposite ACL, in modo tale che gli utenti possano utilizzare e gestire solo i template e le macchine virtuali che sono abilitati a coordinare. Inoltre, le ACL sono state definite anche sulle risorse fisiche, in modo tale da limitare, per ogni utente o gruppo di utenti, ad esempio il numero di macchine virtuali istanziate, la quantità di RAM occupata, il numero di CPU richieste, etc.

Per quanto riguarda l'accesso alle macchine virtuali, è stata implementata un'interfaccia grafica integrata all'interno del portale web FARO, illustrata nella seguente figura.

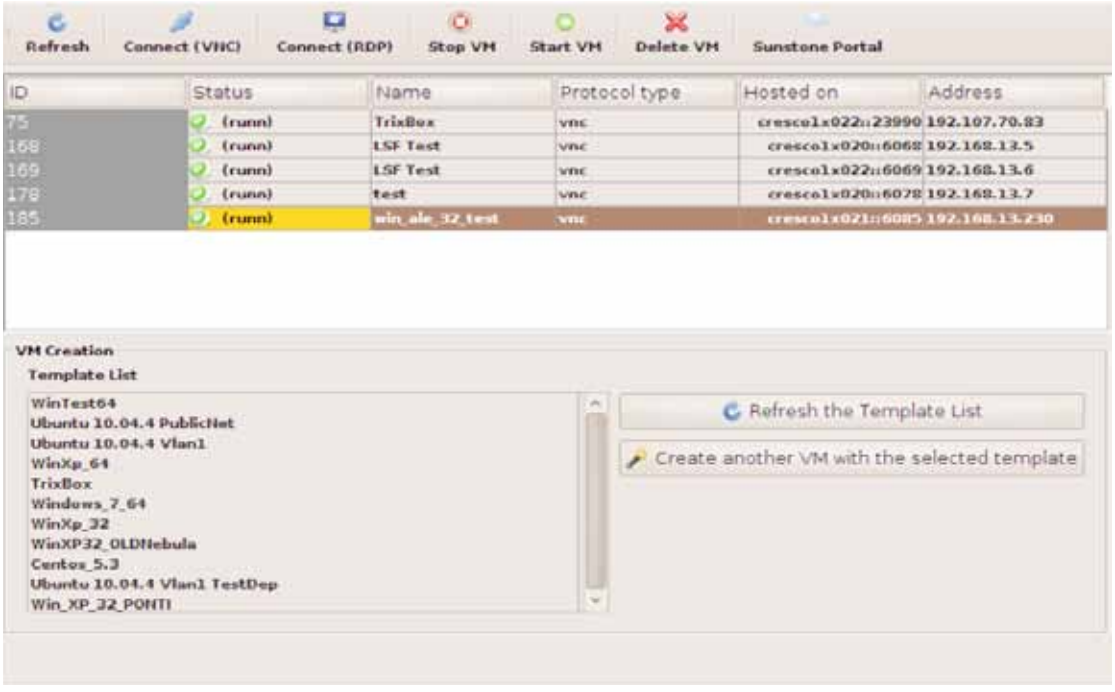


Fig 1 - FARO e OpenNebula – interfaccia di management in ENEA-GRID.
Si noti, in basso, la possibilità di selezionare il template da istanziare e, in alto, l'elenco delle macchine virtuali e i comandi per accedere alla macchina per controllarne l'esecuzione (start, stop, delete).

Si può notare come, anche in questo caso, ENEA-GRID offra una perfetta integrazione con le API di OpenNebula, permettendo la definizione di un'interfaccia grafica semplice, *user friendly*, e completamente funzionale. Questa interfaccia permette infatti di gestire le macchine virtuali e i template, fornendo un accesso immediato alle più comuni operazioni legate al ciclo di vita di una macchina virtuale, quali la creazione partendo da un template, lo *start*, lo *stop*, il *resuming*, e la *destroy*. Come si evince dalla figura, la lista dei template istanziabili può variare da utente a utente, in funzione delle ACL settate dagli amministratori; tuttavia, per ognuno dei template accessibili, l'utente è libero di scegliere quali risorse allocare e quante macchine creare e gestire (ovviamente, sempre con il vincolo delle ACL), senza doverne fare espressa richiesta agli amministratori. Inoltre, l'interfaccia è in grado di capire il tipo di macchina virtuale (ad esempio Windows o Linux) e, nel caso di macchina Windows, offre la possibilità di accedere, oltre che via VNC, anche via RDP, per migliorare l'efficienza.

Questa interfaccia esporta soltanto un sottoinsieme ristretto delle operazioni che possono essere eseguite sulla propria cloud. Per questo motivo, è stato predisposto anche l'accesso al portale web Sunstone di OpenNebula attraverso un apposito pulsante nell'interfaccia; in questo modo, gli utenti che vogliono gestire in maniera più fine le proprie risorse virtuali, avranno accesso ad una set completo di azioni.

3. Storage online e scambio dati con le macchine virtuali

Il contesto delle macchine virtuali presenta interessanti prospettive e si dimostra essere particolarmente versatile e interessante per soddisfare le richieste degli utenti senza stravolgere l'infrastruttura fisica di una rete di calcolo. Tuttavia, restano alcuni problemi pratici da gestire per rendere l'utilizzo di queste soluzioni semplice e immediato. Uno tra tutti riguarda il modo in cui i dati vengono trasferiti da e verso le macchine virtuali. In ENEA-GRID, la presenza del file system distribuito OpenAFS offre di per sé già un'ottima soluzione, permettendo di utilizzare la propria area utente per condividere file tra griglia, macchina client utente e macchine virtuali. Tuttavia, questo sistema prevede comunque l'installazione di un client e di meccanismi di autenticazione, e questa cosa può trovare alcuni limiti, specie se si pensa a dispositivi ubiqui quali smartphone e tablet.

Per coprire un *range* sempre più ampio di possibilità e fornire quindi un servizio sempre più completo, è stato implementato in ENEA un sistema di storage online accessibile via web e che fornisce una valida soluzione a questo problema. Nel seguente paragrafo descriveremo OKBox, un'applicazione web sviluppata in ENEA per supportare lo storage online.

3.1 ENEA OKBox

OKBox è un'applicazione web sviluppata in ENEA per lo storage online che implementa la politica "*Always and Anywhere on*" fornendo un portale web dal quale l'utente può caricare e scaricare i propri file, settare ACL e gestire la condivisione delle risorse avendo a disposizione soltanto la connettività in rete. L'archiviazione dei dati si basa su OpenAFS, mentre per l'autenticazione utilizza Kerberos.

Questo prodotto presenta notevoli vantaggi per l'utilizzatore, in quanto esporta una serie di funzionalità tutte integrate in un unico prodotto, e garantisce una gestione trasparente e sicura dei dati utente, poiché non si poggia su server di terze parti per archiviare i dati e non vìola la *privacy* degli utenti. Inoltre, OKBox offre strumenti per favorire il lavoro collaborativo, permettendo di condividere dati con altri utenti, non necessariamente utenti di ENEA-GRID, semplicemente delegando e abilitando appositi parametri di accesso e condivisione.

Per tutta questa serie di motivi, OKBox è particolarmente utile se impiegato nel contesto delle macchine virtuali, in quanto gli utenti possono trasferire dati tra l'ambiente remoto virtuale e la propria macchina senza installare e configurare altri tool. Un esempio molto comune è la necessità di accedere, da una macchina vir-

tuale, alle stampanti locali installate sulla macchina fisica dell'utente; tramite OKBox è possibile quindi fare l'upload del file da stampare utilizzando il browser web della macchina virtuale e, contestualmente, accedere dalla macchina fisica (sempre tramite il browser web), recuperare il documento e stamparlo in locale.

4. Test dell'infrastruttura

In questo paragrafo, sarà discussa la fase di test dell'infrastruttura di cloud appena descritta per erogare macchine virtuali utente in ENEA-GRID. Nello specifico, si fa riferimento alla configurazione che prevede una cloud farm composta da 3 nodi IBM x3850/x3950-M2, ognuno equipaggiato con 4 CPU Xeon Quad-Core Tigerton E7330 (2.4GHz/1066MHz/6MB L2) e 32 Gb di RAM, per un totale di 48 core e 96 Gb di RAM come risorse fisiche. Inoltre, su uno di questi nodi girano anche i processi di OpenNebula, come da installazione standard consigliata dagli stessi sviluppatori.

L'obiettivo dello *stress test* è stato quello di verificare la responsività e l'usabilità del sistema in sotto diverse condizioni di carico. In particolare, si fa riferimento a due diverse fasi della sperimentazione, e cioè a una prima configurazione che non impiega tutte le risorse fisiche dell'infrastruttura e ad una seconda che sovraccarica l'infrastruttura allocando più macchine virtuali di quante le risorse fisiche sono capaci di sostenere. Entrambi i test sono stati fatti utilizzando un template basato su Windows XP 32bit SP3, che utilizza 1 CPU fisica e 1 Gb di RAM, e ogni macchina virtuale mantiene occupata la sua CPU eseguendo un test *benchmark* di fattorizzazione di grandi numeri interi.

I risultati di utilizzo dell'infrastruttura si sono rilevati ottimi nel primo caso (a sistema non completamente sovraccarico) non mostrando alcun rallentamento durante l'utilizzo delle macchine virtuali. Anche nel secondo caso (sistema completamente sovraccarico, come illustrato in Figura 2) il sistema ha mostrato una buona reat-

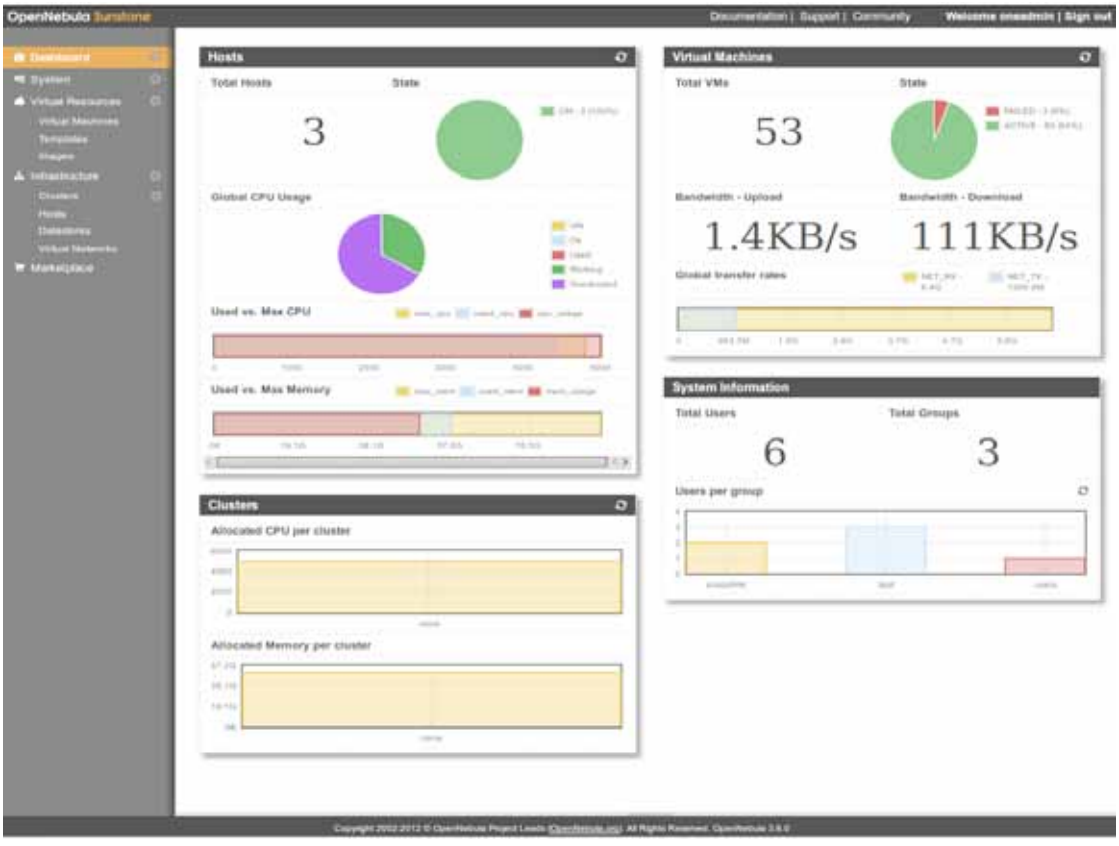


Fig 2 -Stress test dell'infrastruttura di cloud – Dashboard della Sunstone di OpenNebula.

tività, soprattutto nel caso in cui sulla macchina virtuale di test non si effettuano task impegnativi ma si simulano comportamenti standard e comuni, quali, e.g., la navigazione in internet e la video scrittura.

In Figura 2 è illustrata la *dashboard* della Sunstone di OpenNebula: In alto a sinistra, sono presentate le risorse fisiche occupate (numero di host, numero di CPU, e RAM); in alto a destra, le macchine virtuali istanziate e il loro stato; in basso, i cluster di macchine virtuali (a sinistra) e le informazioni su utenti e gruppi (a destra).

5. Conclusioni

In questo lavoro, è stata presentata l'infrastruttura di cloud computing sviluppata in ENEA-GRID. In particolare, è stata descritta la cloud basata su OpenNebula per fornire macchine virtuali per gli utenti ENEA. Queste macchine sono usa e getta, ma la gestione dei profili utente e la loro integrazione in AFS permette di conservare le personalizzazioni e di garantire l'acceso alle risorse della griglia. La flessibilità di OpenNebula ha permesso anche di sviluppare un'interfaccia grafica integrata nel portale web FARO di ENEA-GRID, attraverso la quale gli utenti possono accedere, gestire e amministrare le proprie risorse virtuali. Inoltre, è stato presentato OKBox come applicazione web capace supportare in modo semplice e sicuro la condivisione e lo scambio dei dati tra macchine client utente, griglia e macchine virtuali.

Da primi test effettuati sul sistema in diverse condizioni di carico, abbiamo verificato che l'infrastruttura permette di gestire le risorse in modo efficiente ed efficace, garantendo una responsività più che accettabile anche in condizioni di piena occupazione delle risorse.

Riferimenti bibliografici

[1] ENEA-GRID – http://www.eneagrid.enea.it/

[2] CRESCO – http://www.cresco.enea.it/

[3] Rocchi A., Pierattini S., Bracco G., Migliori S., Beone F., Sciò C., Petricca A.: FARO: accesso WEB a risorse remote per l'industria e la ricerca. Conferenza GARR, 2010

[4] OpenNebula – http://www.opennebula.org

[5] Laboratori Virtuali – http://www.cresco.enea.it/virtulabs.html

[6] Ponti G., Secco A., Ambrosino F., Bracco G., Ciavarella R., Colavincenzo A., D'Angelo P., De Rosa M., Funel A., Guarnieri G., Giammattei D., Migliori S., Pecoraro S., Petricca A., Pierattini S., Podda S., Rocchi A., Sciò C.: Esperimenti di Cloud Computing in ENEA-GRID. Conferenza GARR, 2011

Coautori e collaboratori

Hanno collaborato alla realizzazione di questo articolo: D. Abate, F. Ambrosino, G. Aprea, T. Bastianelli, F. Beone, M. Caporicci, M. Chinnici, A. Cucurullo, P. D'Angelo, A. Della Casa, M. De Rosa, A. Funel, G. Furini, D. Giammattei, S. Giusepponi, R. Guadagni, G. Guarnieri, A. Italiano, A. Mariano, G. Mencuccini, C. Mercuri, P. Ornelli, S. Pecoraro, A. Perozziello, A. Petricca, S. Pierattini, S. Podda, F. Poggi, A. Quintiliani, C. Sciò, F. Simoni dell'ENEA – UTICT (Unità Tecnica per Information and Communication Technology).

Giovanni Ponti
giovanni.ponti@enea.it

É ricercatore ENEA dal 2010. Nel 2005 si è laureato con lode in Ingegneria Informatica presso l'Università della Calabria e, nel 2010, ha conseguito il dottorato di ricerca in Ingegneria dei Sistemi e dell'Informazione. Le sue attività di ricerca riguardano i sistemi HPC, il Cloud Computing e il Data Mining. È coautore di articoli su riviste scientifiche internazionali, paper su atti di conferenza e capitoli di libro.

Distributed open cloud computing, storage e network con WNoDeS: Esperienza ed Evoluzione

Daniele Andreotti[1], Marco Caberletti[1], Vincenzo Ciaschini[1], Gianni Dalla Torre[1], Alessandro Italiano[2], Elisabetta Ronchieri[1], Davide Salomoni[1]

[1]INFN-CNAF, [2]INFN – Sezione di Bari

Abstract. WNoDeS è un *framework* per la virtualizzazione di risorse di calcolo che integra meccanismi di *scheduling* standard, come quelli messi a disposizione dai *batch system* utilizzati nei maggiori centri di calcolo del mondo. L'adozione di *batch system* ampiamente collaudati garantisce la scalabilità e flessibilità di WNoDeS: essa è stata verificata all'interno del centro di calcolo nazionale dell'INFN, dove è utilizzato fin dal 2009. In tale centro WNoDeS gestisce diverse migliaia di macchine virtuali create dinamicamente e messe a disposizione di comunità scientifiche internazionali. WNoDeS è inoltre integrato con soluzioni di accesso alle risorse di tipo Grid e Cloud. Questo lavoro presenta nuove funzionalità di WNoDeS, riportando alcune esperienze di utilizzo negli ultimi 4 anni e descrivendo nuove integrazioni e collaborazioni.

1. Introduzione

WNoDeS (*Worker Nodes on Demand Service*) è un framework progettato e sviluppato dall'INFN per la gestione di macchine reali e virtuali per il calcolo locale e distribuito, sia di tipo Grid che Cloud.

Le richieste architetturali che hanno portato alla realizzazione di WNoDeS includono la necessità di garantire la scalabilità, il riutilizzo di software e il bisogno di minimizzare i cambiamenti di configurazione per centri di calcolo di dimensione medio-grande. WNoDeS utilizza pertanto alcune soluzioni di virtualizzazione e *scheduling* di provata affidabilità e disponibili sul mercato: in particolare, supporta Linux KVM come virtualizzatore e IBM/Platform LSF e Torque/MAUI [1] come schedulatori. L'approccio seguito permette a WNoDeS di essere installato e configurato in un centro di calcolo per gestire risorse reali e virtuali di tipo Grid e Cloud senza richiedere agli amministratori un impegno eccessivamente oneroso. WNoDeS gestisce in particolare le risorse senza la necessità che queste siano suddivise in sottoinsiemi statici dedicati a specifiche applicazioni o interfacce, supportando trasparentemente diversi casi d'uso.

Questo articolo è strutturato come segue: la Sezione 2 descrive la struttura logica di WNoDeS; la Sezione 3 descrive una funzionalità importante di WNoDeS chiamata Mixed Mode; la Sezione IV fornisce lo stato dell'arte; la Sezione V conclude e descrive alcuni nuovi sviluppi.

2. Struttura logica

Dal punto di vista logico l'architettura di WnoDeS si può immaginare caratterizzata da un certo numero di componenti fondamentali (Figura 1), distribuite su cinque livelli.

L'*Access Layer* rappresenta il punto di accesso dell'utente al framework. Questo prevede una *Cloud Command-Line Interface* (CLI) per la sottomissione di richieste di istanziazione di macchine virtuali (VM) secondo la metodologia IaaS, e una *batch* CLI per la sottomissione di *job* di tipo *batch* o *Grid*. La Cloud CLI supporta le operazioni di creazione o cancellazione di un'istanza, di recupero di informazioni della singola istanza e di recupero di tutte le istanze associate ad un utente [5]. La *batch* CLI permette la gestione di richieste di esecuzione di *job* su macchine reali o virtuali, anche customizzate secondo le esigenze degli utenti. Per quanto riguarda

Workshop GARR Calcolo e Storage Distribuito - Selected Papers. Roma 29-30 novembre 2012
ISBN: 978-88-905077-4-8 © 2013 Associazione Consortium GARR

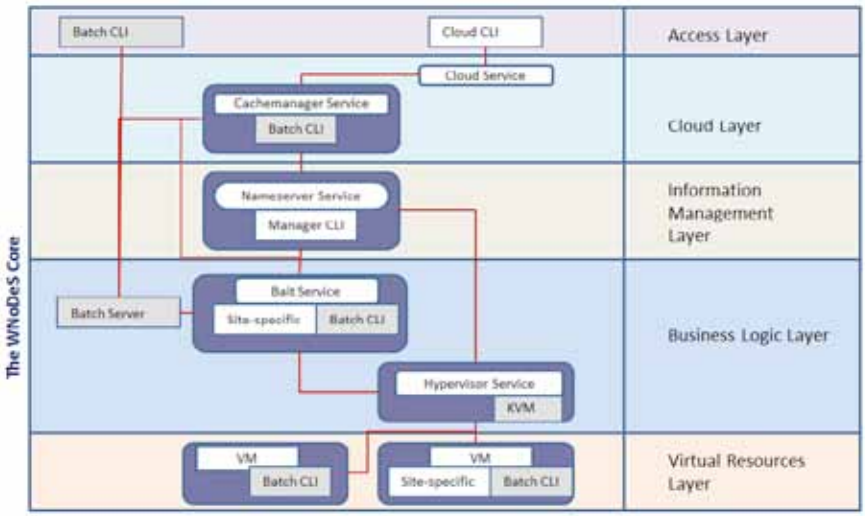


Fig. 1 - Architettura di WNoDeS

autenticazione e autorizzazione, nel caso di accessi di tipo Cloud o Grid l'utente deve appartenere a una VO e possedere un valido certificato X.509 [2]. Nel caso *batch,* l'utente deve appartenere al *pool* di utenti autorizzati dal *batch system* per sottomettere job di calcolo.

Il *Cloud Layer* [5,7] rappresenta la gestione Cloud del *framework* ed è caratterizzato da due componenti:

1. Il *Cloud Service*, che riceve le richieste di instanziazione dell'utente e le invia al *Cachemanager Service* più sotto descritto. La richiesta è marcata con un identificatore, tramite il quale il proprietario può esaminare lo stato della macchina e in particolare scoprire quando essa diventa attiva e accessibile all'utente.
2. Il *Cachemanager Service*, che gestisce la fornitura delle VM, mantenendone alcune già pronte in una cache per renderle disponibili all'utente più velocemente. La dimensione di questa cache è configurabile dall'amministratore.

L'*Information Management Layer* e il *Business Logic Layer* rappresentano la parte centrale di WNoDeS [1].

L'*Information Management Layer* è caratterizzato da due componenti, il *Nameserver Service e la Manager* CLI. Il primo può essere considerato come un catalogo che tiene traccia di tutte le VM in esecuzione su ogni *Hypervisor*, di tutte le immagini memorizzate in un opportuno repository, e delle configurazioni dei possibili *Bait* e *Hypervisor*. Il *Manager* CLI è responsabile della configurazione del *repository* delle immagini: fornisce all'amministratore una serie di opzioni per la gestione del *repository* delle immagini, delle VLAN, degli *hostname* delle immagini, e dei file di configurazione dei vari *Bait* e *Hypervisor*; fornisce inoltre una serie di opzioni per recuperare lo stato di *Bait* e *Hypervisor*.

Il *Business Logic Layer* è costituito da tre componenti:

1. L'*Hypervisor Service* è l'interfaccia al sistema di virtualizzazione responsabile dell'instanziazione delle VM (KVM). Se la richiesta utente è relativa all'esecuzione di un job (locale o Grid), tale job verrà automaticamente eseguito su una VM. Se la funzionalità *Mixed Mode* descritta nel seguito è abilitata, lo stesso *Hypervisor Service* è in grado di eseguire *batch jobs*.
2. Il *Site-specific* è il componente che permette di collegare le richieste delle risorse, internamente sempre viste come job gestiti dal batch server, al core di WnoDeS, lasciando all'amministratore la possibilità di personalizzare la richiesta dell'utente in base alle caratteristiche dell'immagine da utilizzare per l'istanziazione di una data VM. Questo componente invia inoltre la richiesta dell'utente al Bait e ne controlla lo stato.
3. Il *Bait Service* è il gestore delle risorse, responsabile di verificarne la disponibilità, di richiedere la istanziazione di VM quando necessario, e di eseguire la richiesta sulla risorsa più idonea.

Il Virtual Resources Layer rappresenta le VM istanziate sia per l'esecuzione di job tipo grid e

batch, sia per un utilizzo di tipo Cloud.

Le componenti indicate in grigio, come Batch CLI (identico nei vari *Layer* in cui è specificato come da Figura 1), *Batch Server* e Linux KVM, non sono specifici di WNoDeS.

3. Il Mixed Mode

Una delle funzionalità principali di WNoDeS permette di gestire risorse fisiche contemporaneamente sia come nodi tradizionali di un sistema batch sia come *hypervisor* per la istanziazione di VM. Tale funzionalità è detta *Mixed Mode*, è abilitabile opzionalmente e permette un'importante ottimizzazione dell'utilizzo delle risorse di un centro di calcolo, consentendo un'integrazione tra risorse reali e risorse virtuali. Come in installazioni senza *Mixed Mode*, le VM create da WNoDeS possono essere utilizzate per eseguire *job* di tipo *batch* o per fornire risorse di tipo Cloud.

Questa funzionalità permette di soddisfare alcuni requisiti che non sono comunemente gestiti da altri *framework* di virtualizzazione: ad esempio, è spesso preferibile che *job* che richiedono GPGPU o che presentano elevate richieste di I/O locale siano eseguiti senza l'*overhead* introdotto dalla virtualizzazione, e dunque su macchine fisiche. Tramite l'utilizzo di *Mixed Mode* è allo stesso tempo possibile utilizzare le medesime macchine fisiche anche per l'istanziazione di VM (purchè naturalmente risorse come memoria, disco e numero di core necessari siano sufficienti).

Il funzionamento del *Mixed Mode* è basato sul fatto che tutte le richieste di allocazione delle risorse in WNoDeS sono mediate da un *batch system*. Nel caso in cui tali richieste siano *batch job* di tipo tradizionale (sottomesse localmente o ad esempio via Grid) possiamo distinguere tra:

- *batch job* che devono essere eseguiti su un sistema fisico, senza virtualizzazione;
- *batch job* che devono essere eseguiti su VM di un certo tipo.

La distinzione tra l'uno o l'altro tipo avviene in WNoDeS attraverso un file di configurazione. Nel caso in cui le richieste di allocazione delle risorse siano di tipo Cloud, esse sono normalmente sottomesse tramite la Cloud CLI e si riferiscono alla creazione di VM. WNoDeS traduce trasparentemente queste richieste di creazione di VM in job che, come sopra, vengono passati a un batch system.

Sia che si tratti di job tradizionali che di allocazioni Cloud, dunque, per WNoDeS una richiesta di risorse è gestita da un *batch job*. Questo raggiunge un sistema fisico e in particolare il processo *Bait* di WNoDeS. Come descritto sopra, il processo *Bait* ha la responsabilità di associare da una parte le richieste di risorse associate al *job* ricevuto e dall'altra le risorse disponibili sul sistema locale. Con il *Mixed Mode* ogni sistema fisico, sul quale è in esecuzione il processo *Hypervisor* di WNoDeS, fa parte di un cluster di risorse gestito da un *batch system* e può dunque eseguire *job*. Se quindi la richiesta pervenuta al *Bait* è di esecuzione di un *job* non virtualizzato, il *Bait* manderà in esecuzione tale job direttamente sul *"bare metal"* (la macchina fisica). Se la richiesta è di istanziazione di una VM, il *Bait* richiederà all'*Hypervisor* l'istanziazione di tale VM. Se d'altra parte il *Mixed Mode* è disabilitato i sistemi fisici non faranno parte di un *batch system* e non potranno quindi eseguire job. In questo caso tutte le richieste saranno soddisfatte esclusivamente attraverso VM.

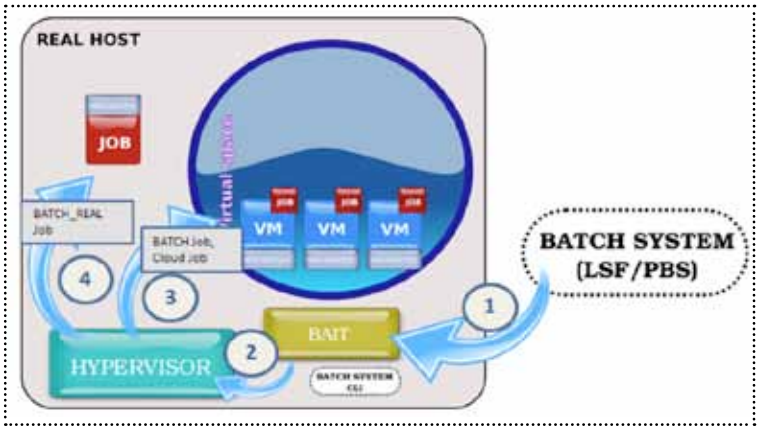


Fig 2 - Architettura di WNoDeS

Naturalmente il *Mixed Mode* richiede, per poter funzionare correttamente, che tutte le richieste di allocazione delle risorse su un certo nodo passino dal batch system e in particolare attraverso il processo *Bait*.

Il *Mixed Mode* è stato rilasciato a partire da WNoDeS 2.0.0-2 nella distribuzione EMI-2 Matterhorn [9]. Il *Mixed Mode* è inoltre integrato con il supporto fornito da WNoDeS per l'istanziazione di risorse Cloud attraverso la Cloud CLI, componente di WNoDeS rilasciata a partire a partire da WNoDeS 3.0.0-1 nella distribuzione EMI-3 Monte Bianco [9].

La figura 2 mostra il funzionamento del *Mixed Mode*, a partire da quando il *job* in coda nel sistema *batch* viene consegnato al *Bait*, che verifica lo stato delle macchine interrogando l'*Hypervisor*. In base alla tipologia di *job* e alla configurazione *site-specific*, un *job* di tipo *batch* sarà eseguito su VM o su macchina fisica. Richieste di tipo Cloud saranno invece sempre eseguite su VM.

Il *Mixed Mode* presenta vantaggi e svantaggi come descritto nella tabella seguente.

4. Stato dell'arte

In letteratura tra i *framework* di virtualizzazione e Cloud Computing di tipo *open source* sono presenti, tra gli altri, *OpenNebula* e *OpenStack*. Entrambi, come WNoDeS, creano un'infrastruttura distribuita come service (IaaS) sulla quale costruire servizi Cloud. Quello che differenzia WNoDeS da queste due tecnologie è la disponibilità della modalità *Mixed mode* e soprattutto lo sfruttamento del concetto di coda del *batch system* per la gestione delle richieste di risorse sia per job tradizionali che per allocazioni Cloud. Normalmente, infatti, le soluzioni di Cloud computing presumono una disponibilità "infinita" di risorse e implementano un sistema di *scheduling* relativamente semplice, che spesso non consente grande flessibilità nella policy di allocazione, specialmente dinamica, delle risorse (si pensi per esempio all'accodamento di richieste di allocazione, alla definizione di "*share*" di risorse, alla necessità di evitare il partizionamento delle risorse tra tenant multipli, alla scalabilità necessaria per supportare migliaia di richieste concorrenti). WNoDeS invece forni-

Vantaggi	Svantaggi
Facile installazione: rende graduale l'installazione di WNoDeS in un centro di calcolo rendendo non necessaria la preassegnazione di macchine dedicate alla virtualizzazione o al Cloud.	Il nodo reale (che contiene l'*Hypervisor*) deve essere accessibile da parte di programmi in *user-space* (job). Questa può essere una limitazione dal punto di vista della sicurezza. D'altra parte, questo è quello che accade normalmente nei casi in cui su sistemi fisici sia presente un *batch system*.
Supporto a diversi casi d'uso, in particolare ove si richieda l'allocazione flessibile di risorse reali e virtuali senza allocazione statica delle risorse.	L'utilizzo delle licenze d'uso di un *batch system* spesso aumenta proporzionalmente al numero di core visti dal *batch system* stesso. Questo può essere un problema se è disponibile solo un numero limitato di licenze (ad esempio con *batch system* di tipo commerciale). Se una macchina fisica è utilizzata per gestire *job* "reali" e per creare VM che a loro volte debbano eseguire *batch job*, il numero totale delle licenze è dell'ordine di O(2 x numero di core). Questo problema evidentemente non si pone nel caso in cui venga utilizzato un *batch system open source*.
Facile customizzazione dei *job* in base alle richieste degli utenti, che avviene tramite la modifica di un file di configurazione precompilato presente nel componente *site-specific*.	

Tab 1 - Vantaggi e svantaggi del Mixed Mode

sce, grazie alla stretta integrazione con un *batch system*, complesse politiche di allocazione e gestione risorse, comprese le componenti di autorizzazione e metering d'uso.

Un'integrazione delle architetture Cloud e Grid può anche avvenire seguendo un approccio di tipo *Grid-over-Cloud* e uno di tipo Cloud-over-Grid. I prodotti CLEVER [10] e MetaCentrum [11] implementano l'approccio *Cloud-over-Grid*. CLEVER è utilizzato per fornire un sistema IaaS realizzato a partire da una infrastruttura Grid. MetaCentrum coordina e fornisce servizi Grid nella Repubblica Ceca per conto della NGI Ceca. L'approccio *Grid-over-Cloud* è utilizzato da StratusLab [12], che fornisce servizi Grid utilizzando le risorse messe a disposizione da un'infrastruttura IaaS: la infrastruttura Grid risultante sfrutta la natura dinamica del Cloud fornendo le risorse quando necessario ed eseguendovi i servizi degli utenti oppirtunamente installati e configurati sulle risorse selezionando l'immagine della risorsa opportuna dal servizio Marketplace. Entrambi gli approcci *Grid-over-Cloud* e *Cloud-over-Grid* prevedono l'incapsulamento di una tecnologia in un'altra. D'altra parte, WNoDeS non privilegia l'interfaccia Grid oppure l'interfaccia Cloud e fornisce risorse reali o virtuali all'utente utilizzando in modo polimorfico le interfacce richieste (locali, Grid o Cloud) richieste dall'utente stesso attraverso una integrazione dinamica delle risorse.

5. Conclusioni

Da diversi anni il framework WNoDeS è utilizzato in produzione per la parte *batch* e Grid al centro di calcolo Tier-1 dell'INFN. La parte Cloud di WNoDeS, recentemente rilasciata, è attualmente messa a disposizione anche in un *testbed* fornito dalla *Italian Grid Infrastructure* (IGI) per le comunità scientifiche che ne richiedono l'accesso. Attualmente le parti specifiche di virtualizzazione Grid e Cloud di WNoDeS sono utilizzate da diversi esperimenti e collaborazioni, tra i quali si citano in particolare l'esperimento astro-particellare Auger , la infrastruttura europea WeNMR per NMR e biologia strutturale [13] e il Federated *Cloud Working Grou*p della European Grid Infrastructure (EGI) [14].

Tra le attività di prossima realizzazione sono presenti l'integrazione con il portale scientifico IGI per la parte Cloud e l'estensione della Cloud CLI per gestire l'istanziazioni di VM da parte di utenti locali per sviluppo software, test e analisi. Più a lungo termine è prevista una gestione più granulare delle risorse dei nodi attraverso meccanismi di *Control Groups* (cgroups) di Linux e l'integrazione di WNoDeS all'interno di OpenStack, utile soprattutto per permettere allo stesso OpenStack di supportare *workload* di calcolo scientifico in modo scalabile. All'interno di quest'ultima attività sono preliminarmente previste, in particolare, l'integrazione dello *scheduling* gestito da WNoDeS con lo scheduling di OpenStack, l'estensione della *dashboard* di Openstack per l'amministrazione ed il monitoraggio dei servizi di WNoDeS e l'integrazione del servizio prototipale di *Dynamic Virtual Networks* di WNoDeS [8] all'interno della gestione rete di OpenStack.

Riferimenti Bibliografici

[1] Davide Salomoni, Alessandro Italiano, Elisabetta Ronchieri, "WNoDeS, a Tool for Integrated Grid and Cloud Access and Computing Farm Virtualization," 2011 Journal of Physics: Conference Series Volume 331 Part 5: Computing Fabrics and Networking Technologies.

[2] Vincenzo Ciaschini, Davide Salomoni, "An Authentication Gateway for Integrated Grid and Cloud Access," 2011 Journal of Physics: Conference Series Volume 331 Part 6: Grid and Cloud Middleware.

[3] Claudio Grandi, Alessandro Italiano, Davide Salomoni, Anna Karen Calabrese Melcarne, "Virtual Pools for Interactive Analysis and Software Development through an Integrated Cloud Environment," 2011 Journal of Physics: Conference Series Volume 331 Part 7: Distributed Processing and Analysis.

[4] Davide Salomoni, Anna Karen Calabrese Melcarne, Andrea Chierici, Luca Cestari, Gui-

do Potena, Peter Solagna "Performance improvements in a large scale virtualization system," PoS(ISGC 2011 & OGF 31)049.

[5] Davide Salomoni, Daniele Andreotti, Luca Cestari, Guido Potena, Peter Solagna, "A Web-based Portal to Access and Manage WNoDeS Virtualized Cloud Resources," PoS(ISGC 2011 & OGF 31)054.

[6] Davide Salomoni, Elisabetta Ronchieri, "WORKER NODES ON DEMANDS SERVICE – Requirements for Virtualized Services, " http://web2.infn.it/wnodes/index.php/documentation/files-download/28_d67514b33dae20f979d866990b583b74

[7] Elisabetta Ronchieri, Giacinto Donvito, Paolo Veronesi, Davide Salomoni, Alessandro Italiano, Gianni Dalla Torre, Daniele Andreotti, Alessandro Paolini, "Resource Provisioning through Cloud and Grid Interfaces by means of the Standard CREAM CE and the WNoDeS Cloud Solution," PoS(EGICF12-EMITC2)124.

[8] Marco Caberletti, Davide Salomoni, "A Dynamic Virtual Networks Solution for Cloud Computing," Proceedings of the 2nd International Workshop on Netware-aware Data Management, November 11, 2012, Salt Lake City, Utah, USA.

[9] Cristina Aiftimiei, Andrea Ceccanti, Danilo Dongiovanni, Andrea Di Meglio, Francesco Giacomini, "Improving the quality of EMI Releases by leveraging the EMI Testing Infrastructure," 2012 Journal of Physics: Conference Series Volume 396 Part 5.

[10] Francesco Tusa, Maurizio Paone, Massimo Villari, Antonio Puliafito, "CLEVER: a Cloud Cross-Computing Platform leveraging GRID resources," UCC pp. 390 – 396, IEEE Computer Society (2011).

[11] Ruda Miroslav, Šustr Zdenek, Sitera Jirí, Antoš David, Hejtmánek Lukáš, Holub Petr, "Virtual Clusters as a New Service of MetaCentrum, the Czech NGI," In Cracow Grid Workshop '09. Krakow : Academic Computer Centre CYFRONET AGH, 2010. ISBN 978-83-61433-01-9, pp. 64-71. 12.10.2009, Krakow.

[12] Charles Loomis, Mohammed Airaj, Marc-Elian Bégin, Evangelos Floros, Stuart Kenny, David O'Callaghan, "StratusLab Cloud Distribution," In Dana Petcu and José Luis Vázquez-Poletti Eds., European Research Activities in Cloud Computing, pp. 260–282, Cambridge Scholars Publishing (2012).

[13] Tsjerk A. Wassenaar, Marc van Dijk, Nuno Loureiro-Ferreira, Gijs van der Schot, Sjoerd J. de Vries, Christophe Schmitz, Johan van der Zwan, Rolf Boelens, Andrea Giachetti, Lucio Ferella, Antonio Rosato, Ivano Bertini, Torsten Herrmann, Hendrik R. A. Jonker, Anurag Bagaria, Victor Jaravine, Peter Güntert, Harald Schwalbe, Wim F. Vranken, Jurgen F. Doreleijers, Gert Vriend, Geerten W. Vuister, Daniel Franke, Alexey Kikhney, Dmitri I. Svergun, Rasmus H. Fogh, John M. C. Ionides, Ernest D. Laue, Chris A. E. M. Spronk, Simonas Jurksa, Marco Verlato, Simone Badoer, Stefano Dal Pra, Mirco Mazzucato, Eric Frizziero, Alexandre M. J. J. Bonvin, "WeNMR: Structural Biology on the Grid," J. Grid. Computing. V. 10, pp. 743-767.

[14] Matteo Turilli, Michel Drescher, Steven Newhouse, David Wallom "Federeting clouds to aid researchers", ISGTW – International Science grid this week, 17 October 2012.

Elisabetta Ronchieri
elisabetta.ronchieri@cnaf.infn.it

Lavora presso il CNAF dell'INFN dal 2001, dopo aver trascorso un paio di anni nell'industria sviluppando software di carte nautiche e simulazioni di sistemi dinamici. Ha partecipato a numerosi progetti Europei riguardanti software engineering e calcolo distribuito in ambito Grid e Cloud. Attualmente gestisce il progetto WNoDeS, fa parte del Fedcloud Working Group del progetto Europeo EGI Inspire e collabora allo sviluppo di modelli per la valutazione della qualità del software.

Sull'interoperabilità tra risorse locali, Grid e cloud per la realizzazione di un'infrastruttura di calcolo distribuito in Italia

Diego Scardaci[1], Giuseppe Andronico[1], Roberto Barbera[1,2], Riccardo Bruno[1], Marco Fargetta[1], Andrea Fornaia[1], Giuseppe La Rocca[1], Salvatore Monforte[1], Rita Ricceri[1], Riccardo Rotondo[1], Davide Saitta[1].

[1]INFN Sezione di Catania, [2]Dipartimento di Fisica e Astronomia dell'Università di Catania

Abstract. In base al report finale dello studio condotto da eResearch2020, un'iniziativa finanziata dalla Commissione Europea, i maggiori ostacoli che finora hanno impedito un'ampia adozione ed un uso capillare delle infrastrutture di calcolo distribuite (DCI) sono stati la mancanza di interoperabilità tra i diversi middleware e la loro difficoltà di utilizzo, soprattutto nella fase iniziale, in cui la gestione dei certificati digitali personali e la mancanza di standard consolidati rende la curva di apprendimento davvero molto ripida per i non esperti in informatica.
Questo lavoro presenta il Catania Science Gateway Framework, per la creazione di portali tematici, che affronta e risolve entrambe le problematiche, fornendo un accesso semplice e intuitivo alle DCI. Ciò avviene tramite lo standard JSR 286, un sistema di autenticazione e autorizzazione basato sulle federazioni di identità e lo standard SAML, senza la necessità per l'utente di avere un certificato digitale personale, e, infine, un'interfaccia verso vari middleware che usa lo standard SAGA e che consente di eseguire in maniera trasparente la stessa applicazione su DCI eterogenee (cluster locali, Grid, Cloud, HPC, ecc.).

1. Introduzione

Negli ultimi 30 anni, grazie alla continua riduzione dei costi dei processori e della rete, a parità di potenza di calcolo e di banda passante, il calcolo scientifico si è evoluto passando dai mainframe monolitici, ai cluster di calcolatori, alle infrastrutture Grid e, infine, alle cloud.

I cluster di calcolatori, le diverse infrastrutture Grid e le cloud presenti oggigiorno nel mondo, sebbene basati, su un medesimo modello "distribuito", utilizzano strumenti software differenti, che non sono quasi mai interoperabili tra loro. Esiste, infatti, una grande varietà di Local Resource Management System (Condor, LSF, PBS, Torque+MAUI, ecc.), di middleware Grid, di stack cloud. Si tratta in buona sostanza di implementazioni "verticali" che non si "parlano" l'una con l'altra. Questo fa sì che l'offerta di calcolo e storage delle cosiddette e-Infrastructure non si basi su una piattaforma globale standard, ma sia bensì vista come un insieme di diverse "isole" (le varie infrastrutture) che non comunicano tra loro.

Nel report finale dello studio condotto da e-Research2020 [1], un'iniziativa finanziata dalla Commissione Europea con l'obiettivo di analizzare il ruolo delle infrastrutture digitali nella creazione di comunità virtuali globali di ricercatori e scienziati, è sottolineato come l'assenza di questa interoperabilità sia una grande barriera nell'adozione del modello basato sulle infrastrutture virtuali da parte dei potenziali utenti. Infatti, cambiare l'infrastruttura usata vuol dire per il ricercatore dover studiare nuove interfacce e modificare la propria applicazione al fine di poterla eseguire sulla nuova piattaforma. Altre importanti barriere identificate dallo studio sono la difficoltà di accesso e il tempo necessario

Workshop GARR Calcolo e Storage Distribuito - Selected Papers. Roma 29-30 novembre 2012
ISBN: 978-88-905077-4-8 © 2013 Associazione Consortium GARR

ad adattare la propria applicazione per essere eseguita correttamente sulle varie infrastrutture.

La presenza di questi ostacoli sta alla base del fatto che il numero di utenti della European Grid Infrastructure (EGI) [2] è solo di poco superiore alle 22.000 unità, circa l'1% del numero di addetti ai lavori della ricerca pubblica in Europa. Eliminare queste barriere aprirebbe il mondo delle infrastrutture digitali a un enorme bacino di utenti potenziali e per far ciò è fondamentale rendere interoperabili le varie infrastrutture digitali presenti in Europa e nel resto del mondo, fornendo ai ricercatori strumenti di accesso semplici ed intuitivi che possano far loro vedere immediatamente il vantaggio di utilizzarle.

L'interoperabilità può essere definita come l'abilità di diversi sistemi e organizzazioni di lavorare insieme. Il termine è usato anche in contesto informatico e lo standard ISO/IEC 2382-01 (Information Technology Vocabulary, Fundamental Terms) lo definisce come la capacità di comunicare, eseguire programmi o trasferire dati tra varie unità funzionali in un modo che richiede all'utente di non avere alcuna (o al più una limitata) conoscenza delle caratteristiche di queste unità.

Obiettivo di questo lavoro è proporre un sistema per garantire l'interoperabilità, come sopra definita, tra tutte le infrastrutture di calcolo al momento esistenti (cluster locali, Grid e Cloud), eliminando le barriere suddette.

Lo strumento da noi scelto per raggiungere tale obiettivo è quello degli Science Gateway. Secondo TeraGrid/XSede (https://www.xsede.org/), uno Science Gateway è costituito da un set di tool, applicazioni e basi di dati sviluppati dalle comunità scientifiche e integrati in un portale web, solitamente con un'interfaccia utente semplice, che è adattato a soddisfare i bisogni di una specifica comunità.

Al fine di poter creare velocemente diversi Science Gateway che soddisfino queste caratteristiche, abbiamo progettato e sviluppato un framework che fornisce una serie di funzionalità comuni che possano essere facilmente usate e adattate, ogni qual volta occorre sviluppare un nuovo Science Gateway per una comunità specifica. Il Catania Science Gateway Framework [3, 4] è descritto nel prossimo paragrafo.

2. Il Catania Science Gateway Framework

I requisiti di base che hanno guidato la progettazione del Catania Science Gateway Framework (CSGF) sono:

- l'uso di standard riconosciuti a livello internazionale;
- la semplicità di sviluppo;
- la semplicità di uso;
- la riusabilità.

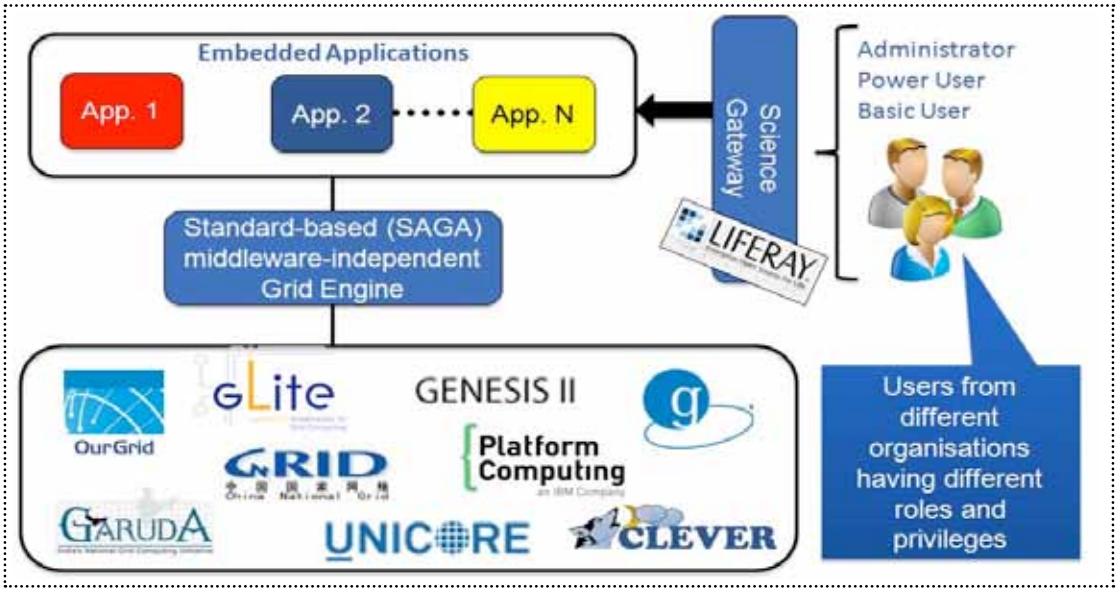


Fig. 1 - Schema del Catania Science Gateway Framework

Lo schema del framework è mostrato in figura1. Il CSGF è costituito da un set di librerie per gestire i meccanismi di autenticazione e autorizzazione verso le infrastrutture, che consentono di eseguire applicazioni con diversi middleware quali LRMS, Grid e Cloud. Il CSGF rispetta inoltre pienamente le politiche definite da EGI per i portali che offrono accesso alle DCI tramite web [5, 6].

Nelle implementazioni esistenti, il CSGF è stato integrato all'interno dell'enterprise portal framework Liferay [7], un software per costruire portali che permette di integrare portlet standard JSR 286 [8] in un application server (GlassFish [9], nel nostro caso). Ogni applicazione di un dato Science Gateway implementato con il nostro framework è integrata in una portlet specifica e ciò fornisce una serie di importanti vantaggi:

- la possibilità di comporre uno Science Gateway "on the fly", aggiungendo o rimuovendo portlet in funzione dei requisiti della comunità dei suoi utenti;
- la capacità di abilitare permessi specifici per ogni utente e ogni applicazione;
- il riutilizzo delle portlet che integrano un'applicazione in più di uno Science Gateway, grazie alla loro portabilità, garantita dallo standard;
- la possibilità di configurare in modo opportuno le portlet usando le portlet preferences (ad es. per stabilire il set di risorse su cui eseguire una data applicazione).

2.1 Autenticazione e Autorizzazione

L'autenticazione nel CSGF si basa sull'uso delle cosiddette identità federate [10,11]. Ciò è reso possibile dall'adozione dello standard SAML 2.0 [12] e della sua implementazione fatta da Shibboleth [13]. Al fine di massimizzare il numero di potenziali utenti, gli Science Gateway implementati con il CSGF possono essere configurati come Service Provider della federazione italiana IDEM [11] e, attraverso questa, dell'interfederazione internazionale eduGAIN [14]. È in questo contesto importante sottolineare che, allo scopo di far accedere ai Catania Science Gateway anche coloro che non appartengono a nessuna federazione, è stata creata la federazione "catch-all" GrIDP [15] e l'IDP Open [16] che sono co-gestiti dal GARR e dalla Sezione di Catania dell'INFN.

L'autorizzazione è invece gestita tramite un server LDAP, il quale registra per ciascun utente i ruoli ricoperti all'interno delle varie organizzazioni facenti parte della comunità, e le relative autorizzazioni ottenute all'interno dello Science Gateway. Per soddisfare un preciso requisito di sicurezza, la registrazione delle autorizzazioni è gestita manualmente.

2.2 Il Grid-Engine

La componente principale del CSGF è il Grid-Engine, un tool che può eseguire applicazioni su ogni DCI (Grid, Cloud, HPC o cluster locali) tramite l'adozione dello standard SAGA [17], e della sua implementazione JSAGA [18], che definisce un'interfaccia software di alto livello orientata alle applicazioni. Questa può essere usata per eseguire applicazioni su middleware differenti. Un'infrastruttura è supportata da JSAGA quando è disponibile il corrispondente "adaptor". Un JSAGA adaptor è una libreria che mappa le chiamate standard SAGA con l'interfaccia del middleware da supportare.

Sfruttando le caratteristiche di SAGA, tramite il CSGF è possibile creare un front-end unico, capace di sottoporre applicazioni su differenti DCI. Il CSGF fornisce agli sviluppatori di applicazioni una potente interfaccia che consente di definire set di risorse (siti Grid, macchine HPC, Cloud, ecc.) sui quali l'applicazione potrà essere eseguita. Il set di risorse sarà scelto dallo sviluppatore in funzione delle caratteristiche delle applicazioni e dell'accessibilità delle risorse. Come mostrato in figura 2, il set potrà essere composto da risorse appartenenti a tutti i tipi di DCI supportate o solo di un tipo.

Per esempio, il set potrà essere composto solo da risorse HPC, nel caso di applicazioni che possono essere eseguite solo su macchine parallele, oppure, nel caso di un workflow, ogni suo nodo potrà essere eseguito su un'infrastruttura diversa, in accordo alle caratteristiche del singolo nodo.

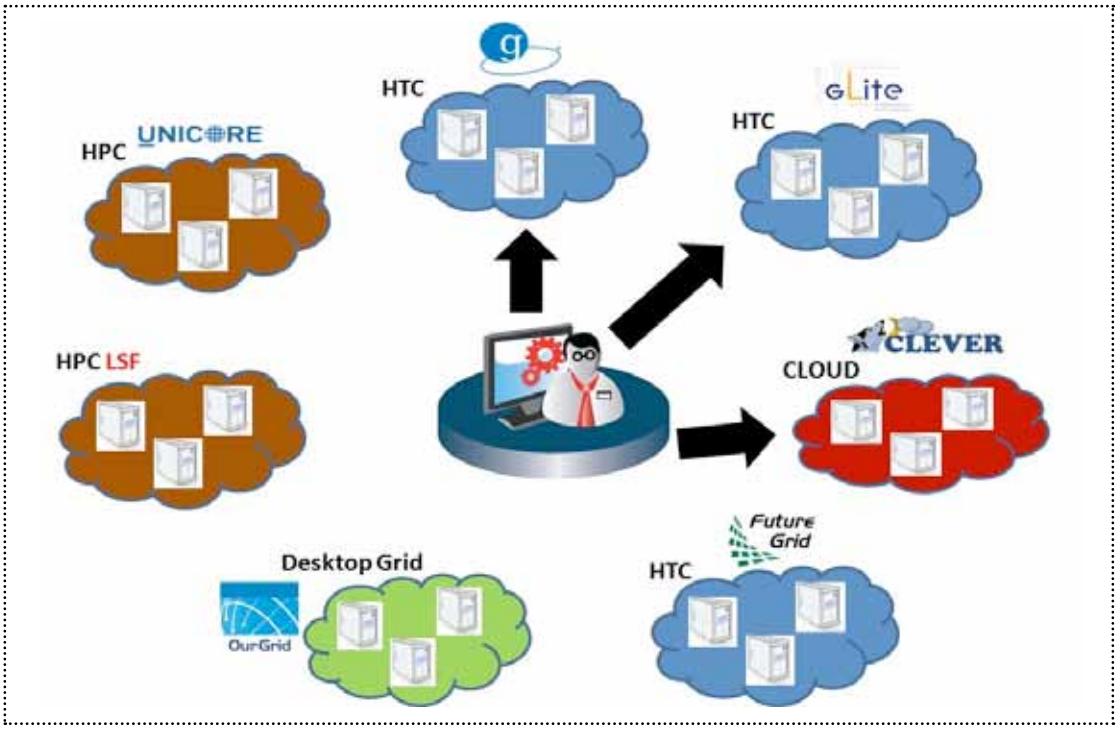


Fig 2 - Lo sviluppatore può scegliere su quali infrastrutture distribuite eseguire l'applicazione in base alle carattestiche dell'applicazioni e ai suoi permessi di esecuzione

2.2.1 Lo standard SAGA per il mondo cloud

Sebbene lo standard SAGA sia stato definito per le infrastrutture Grid, esso può essere adattato senza stravolgimenti anche per quelle Cloud. In pratica, il modello di servizio Software as a Service (SaaS), per cui i provider cloud installano e operano applicazioni nella cloud e gli utenti accedono a tali software tramite client, può essere interamente offerto agli utenti da uno Science Gateway che si basi sul CSGF e adotti lo standard SAGA. È importante osservare che, nel modello SaaS, l'utente cloud non gestisce l'infrastruttura dove l'applicazione viene eseguita. Nel CSGF, le chiamate SAGA, quando si riferiscono a un'infrastruttura cloud, devono essere mappate con tutte le operazioni necessarie per eseguire un'applicazione su cloud. Il nostro requisito è che con le stese chiamate SAGA si possa eseguire un'applicazione su una qualsiasi infrastruttura, comprese quelle di tipo cloud. Per soddisfarlo abbiamo definito un algoritmo che mappa la chiamata SAGA per eseguire un'applicazione in un set di azioni in un'infrastruttura cloud:

1. un utente invia un'applicazione verso una cloud usando le API SAGA standard;
2. il motore SAGA capisce che l'applicazione deve essere eseguita su cloud e contatta il Cloud Virtual Infrastructure Manager (VIM) per identificare e avviare una Virtual Machine (VM) dove l'applicazione può essere eseguita;
3. il Cloud VIM fornisce al motore SAGA le informazioni necessarie per accedere alla macchina;
4. il motore SAGA trasferisce l file di input sulla VM e avvia l'applicazione;
5. il motore SAGA controlla l'applicazione e recupera l'output quando pronto;
6. il motore SAGA contatta il Cloud VIM per spegnere la VM;
7. il motore SAGA restituisce l'output dell'applicazione all'utente tramite le API standard di SAGA.

Questo algoritmo consente al CSGF di sfruttare un'infrastruttura Cloud tramite un'interfaccia SAGA come una DCI classica. Quindi, gli u-

tenti potranno eseguire applicazioni su risorse Grid, HPC e Cloud in un modo trasparente.

3. La demo di CHAIN dell'interoperabilità tra middleware differenti

Obiettivo di questa demo, realizzata nell'ambito del progetto europeo CHAIN [19] utilizzando uno Science Gateway dedicato [20] implementato con il CSGF, è stato quello di dimostrare che:

- le infrastrutture possono essere rese interoperabili (secondo la definizione data precedentemente) l'una con l'altra a livello di applicazioni utente, usando standard.
- Applicazioni appartenenti a diverse comunità scientifiche possono essere sottoposte da qualunque posto, per essere eseguite ovunque.

Nella demo sono state coinvolte infrastrutture europee, nord e sudamericane, africane e asiatiche come si può evincere dalla figura 3.

I middleware coinvolti sono stati:

- EMI-gLite (in Europa);
- EMI-Unicore (in Europa);
- Globus (in India);
- Genesis II (in USA);
- GOS (in Cina);
- OurGrid (in Brasile);
- Platform Computing LSF (sul sito ENEA-CRESCO di Portici).

La fiigura 4 mostra la distribuzione geografica dei job in esecuzione, o ormai terminati, inviati durante la demo. Ogni middleware è identificato da un colore diverso, specificato nella legenda.

In figura 5 è mostrato uno zoom della mappa dei job in cui si vedono quelli eseguiti sul sito di Portici dell'ENEA tramite LSF.

4. Conclusioni

Le infrastrutture di calcolo distribuito (DCI) potranno incrementare notevolmente il loro bacino di utenti solo fornendo servizi realmente semplici da usare. Il Catania Science Gateway Framework, col supporto alle federazioni di identità, cambia radicalmente il modo in cui le DCI possono essere usate allargando enormemente il bacino degli utenti potenziali in diversi continenti e organizzazioni,.

Inoltre, l'adozione di standard (JSR 286, SAGA, SAML, ecc.) rappresenta un concreto investimento verso la sostenibilità del framework.

Il CHAIN worldwide interoperability program ha dimostrato che, tramite uno Science Gateway basato su standard, gli utenti possono accedere, in modo trasparente e da qualunque lo-

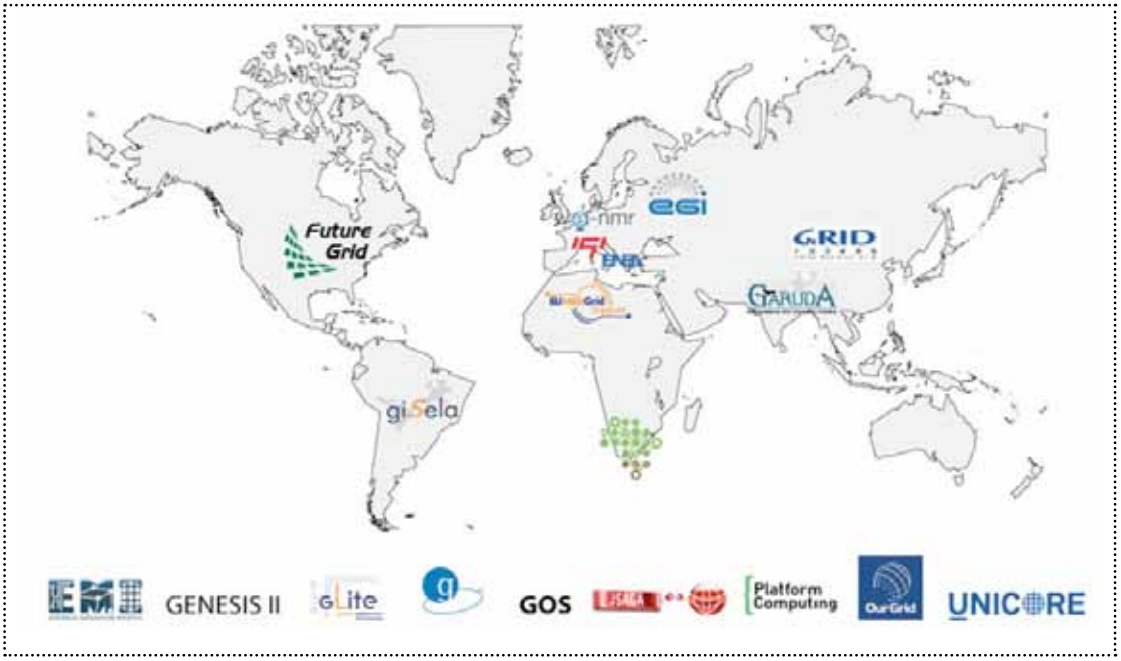


Fig 3 - e-Infrastructures coinvolte nella CHAIN worldwide Interoperability Demo

Fig 4 - Distribuzione geografica dei job in esecuzione. Ogni middleware è identificato con un colore diverso

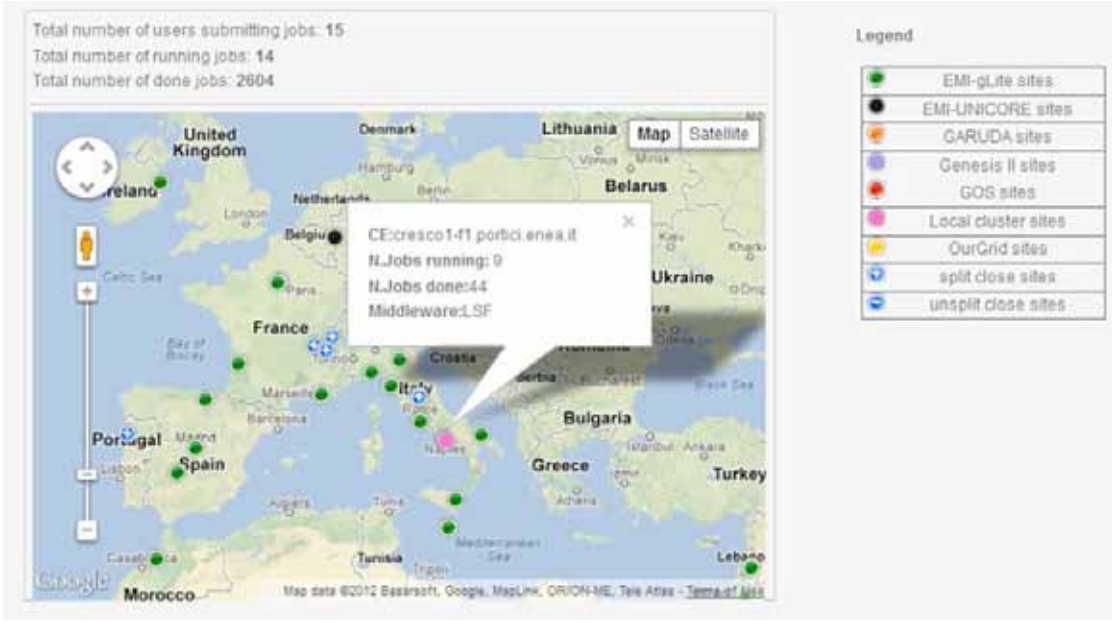


Fig 5 - Mappa dei job eseguiti sul sito ENEA di Portici con LSF

calità, a diverse DCI globali (cluster locali, HPC, Grid, Cloud).

Noi proponiamo lo stesso approccio per riunire le risorse distribuite di tutta l'Italia e costruire un'infrastruttura Italiana completamente interoperabile, nel rispetto delle specificità locali e sfruttando le competenze di tutte le organizzazioni interessate a partecipare.

Riferimenti Bibliografici

[1] www.eresearch2020.eu/eResearch2020%20Final%20Report.pdf

[2] www.egi.eu

[3] V. Ardizzone et al. J. Grid Computing (2012) 10:689-707, DOI 10.1007/s10723-012-9242-3

[4] www.catania-science-gateways.it

[5] https://documents.egi.eu/public/ShowDocument?docid=80

[6]https://documents.egi.eu/public/ShowDocument?docid=81

[7] www.liferay.com

[8] www.jcp.org/en/jsr/detail?id=286

[9] http://glassfish.java.net

[10] Per maggiori informazioni sulle Federazioni d'Identità esistenti nel mondo della ricerca scientifica, si visiti il sito web https://refeds.org

[11] Per maggiori informazioni sulla Federazione d'Identità italiana, si visiti il sito web http://www.idem.garr.it

[12] http://saml.xml.org

[13] http://shibboleth.net

[14] www.edugain.org

[15] http://gridp.garr.it

[16] http://idpopen.garr.it

[17] www.gridforum.org/documents/GFD.90.pdf

[18] http://grid.in2p3.fr/jsaga

[19] www.chain-project.eu

[20] http://science-gateway.chain-project.eu

Diego Scardaci
diego.scardaci@ct.infn.it

Laureato in Ingegneria Informatica, ha iniziato la carriera come ricercatore al TelecomItaliaLab. Dal 2006 lavora per l'INFN sez. di Catania occupandosi di e-Infrastructures, ricoprendo incarichi di management in progetti Europei. Attualmente è l'Activity Manager del WP7 del progetto EGI-InSPIRE.

Realizzazione di un'infrastruttura Cloud pilota basata su OpenStack

Livio Fanò Illic[1], Enrico Fattibene[2], Matteo Manzali[2,3], Hassen Riahi[1], Davide Salomoni[2], Andrea Valentini[1], Paolo Veronesi[2], Valerio Venturi[2]

[1]INFN-PERUGIA, [2] INFN CNAF, [3]Università degli Studi di Ferrara

Abstract. Il contributo si sviluppa all'interno del progetto Marche Cloud, che prevede lo sviluppo di un'infrastruttura cloud basata su software open source per la Regione Marche. In questo lavoro si presenta la realizzazione del prototipo di tipo IaaS (Infrastructure as a Service) di tale infrastruttura, inizialmente installato al CNAF e successivamente portato presso il data center della Regione Marche ad Ancona. L'infrastruttura è basata sul software OpenStack, installato e configurato nelle componenti di *identity service*, *image repository, compute node*, *object storage* e *dashboard*. Nel progetto sono supportati diversi sistemi operativi e formati di immagini per le VM. Il file system distribuito GlusterFS è stato utilizzato per abilitare la funzionalità di live migration e al fine di ottenere ridondanza, performance e alta affidabilità di alcune componenti dell'infrastruttura stessa. È stato sviluppato un sistema flessibile di monitoring e allarmistica sfruttando l'integrazione in OpenStack di *framework* esterni, specificatamente Ganglia e Nagios.

1. Introduzione

La Regione Marche intende dotarsi di un'infrastruttura di *Cloud computing* (MCloud, Figura 1) che eroghi innovativi servizi ad alto contenuto tecnologico ad aziende, istituzioni pubbliche e società civile favorendo:

- efficienza e innovazione, sviluppo di nuovi prodotti, crescita della produttività;
- opportunità di business per il territorio marchigiano;
- realizzazione di importanti economie di scala nell'uso di risorse pubbliche e private;
- attrazione e diffusione di competenze avanzate nel settore strategico ICT;
- progressi nell'interscambio di informazione e conoscenza, nell'aggregazione sociale e nella qualità della vita per i cittadini e le imprese.

A questo scopo la Regione Marche ha stipulato un Protocollo d'Intesa con l'Istituto Nazionale di Fisica Nucleare (INFN), definendo un progetto pilota *MCloud* con l'obiettivo di implementare un'infrastruttura Cloud e su di essa servizi per i cittadini. Un primo servizio, già reso disponibile, consente l'accesso a refertazione elettronica di laboratori di analisi presenti sul territorio marchigiano. Il progetto pilota è strutturato in quattro work packages (WP):

1. WP1 - Infrastruttura;
2. WP2 - Monitoring;
3. WP3 - Autenticazione;
4. WP4 - Interfacce utente;

Questo lavoro descrive i compiti dei WP più infrastrutturali, specificatamente WP1 e WP2.

L'adozione di software Open Source, elemento caratterizzante della proposta architetturale, è fortemente raccomandata per le Pubbliche Amministrazioni; cfr. ad esempio le recenti modifiche introdotte all'art. 68 del Codice dell'Amministrazione Digitale (D.Lgs. 82/2005) [1].

Come software Cloud di riferimento è stato adottato OpenStack [2]. Le motivazioni che hanno portato a questa scelta sono principalmente le seguenti:

- è un prodotto Open Source che può essere eseguito su piattaforme anch'esse interamente Open Source come, ad esempio, sistemi operativi Linux;

Workshop GARR Calcolo e Storage Distribuito - Selected Papers. Roma 29-30 novembre 2012
ISBN: 978-88-905077-4-8 © 2013 Associazione Consortium GARR

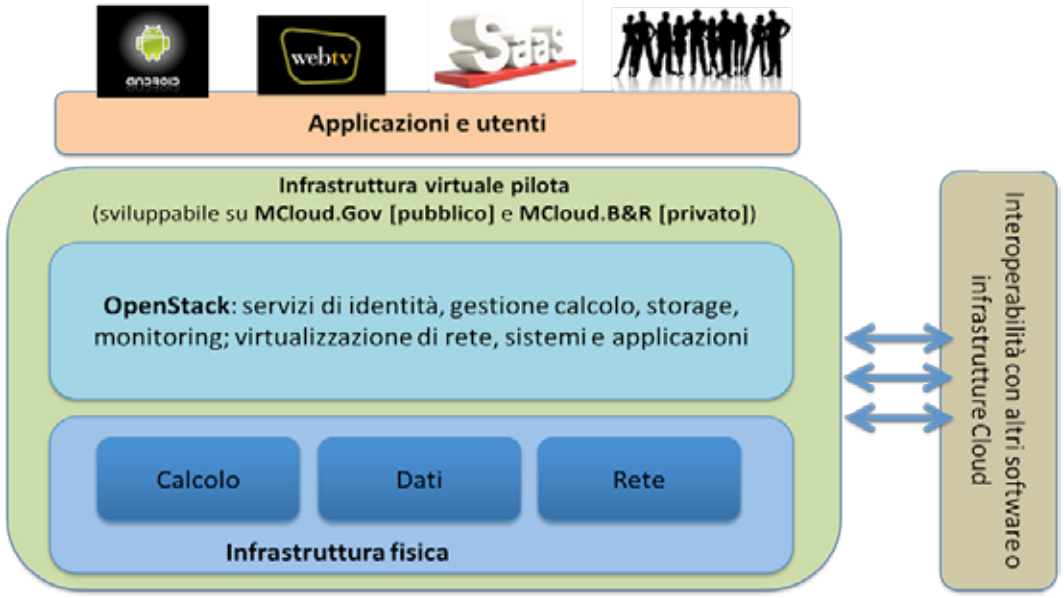


Fig 1 - Architettura del progetto MCloud

- ha un forte supporto da parte dell'industria: tra i partner del consorzio OpenStack [3] figurano molti tra i maggiori *player* nel campo dell'informatica mondiale;
- rispetto ad altre piattaforme di Cloud computing mostra una continua forte crescita sia in termini di funzionalità che di comunità di sviluppatori [4];
- esistono diffuse competenze sul prodotto per la realizzazione di piattaforme di Cloud computing da parte dell'INFN e di partner dell'INFN, come il CERN;
- ha un disegno architetturale aperto e modulare, principalmente sviluppato in Python. Questo comporta facile estendibilità e possibilità di un'adozione selettiva, composta solamente dalle parti che interessano per un dato caso d'uso;
- ha una governance interna ben definita e rilasci del software periodici e incrementali;
- è interoperabile con altri sistemi di tipo Cloud pubblici o privati. In particolare, attraverso OpenStack è possibile eseguire VM create in altri ambienti, anche proprietari, come *VMware*, ed è possibile la connessione con *VMware ESX Server* [5]. È anche possibile connettere OpenStack a Cloud pubbliche attraverso lo standard API "de facto" Amazon EC2.

2. I servizi forniti da OpenStack che caratterizzano l'infrastruttura pilota

Con riferimento alla figura architetturale generale di OpenStack nella versione attuale denominata "*Folsom*" [6] (Figura 2), nel prototipo MCloud si è decisa una focalizzazione sulle componenti di:

Dashboard: OpenStack fornisce un'interfaccia di gestione dell'infrastruttura attraverso un modulo di dashboard chiamato Horizon.

Storage: OpenStack supporta due tipi di Cloud storage:

- object storage (servizio Swift) per la gestione di file intesi come singoli oggetti (cioè non come volumi montabili);
- block storage (servizio nova-volume/cinder) per la definizione di un'area dati persistente che, in analogia a un disco USB, può essere collegata ad una VM per volta e che viene preservata quando la VM viene eliminata.

Poiché nel pilota MCloud non è richiesto un servizio di memorizzazione file di tipo object, nell'infrastruttura pilota è stata abilitata solo la funzionalità di block storage.

Network: la versione *Folsom* di OpenStack integra una componente, chiamata Quantum, dedicata alla definizione delle connessioni di rete con

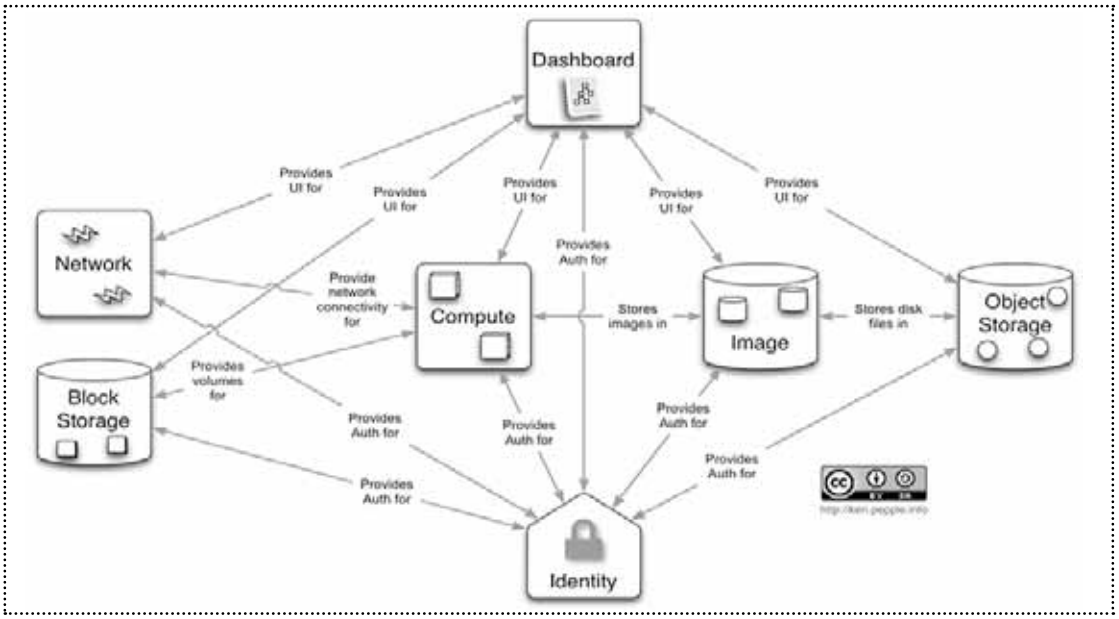


Fig 2 - Architettura di OpenStack "Folsom"

un servizio di "*Network as a Service*". Nella prima fase di MCloud, tuttavia, allo scopo di ottenere una configurazione il più possibile affidabile e semplice, è stato deciso di configurare la parte di rete attraverso una replica della componente nova-network su tutti i nodi coinvolti.

Image repository: OpenStack fornisce un servizio di catalogazione e gestione delle immagini virtuali chiamato Glance, che gestisce diversi formati (es. raw, qcow2, vmdk).

Autenticazione: un punto architetturalmente importante è dato dalla disponibilità in OpenStack di una componente dedicata all'autenticazione, chiamatz Keystone. Keystone non supporta ancora ufficialmente un'autenticazione federata basata sul Security Assertion Markup Language (SAML), ma è stato esteso in tal senso nell'ambito del WP3 al fine di facilitare l'integrazione con i sistemi SAML-based già in uso in Regione.

Le linee-guida per l'architettura e per l'integrazione dei servizi da realizzare sull'infrastruttura del Centro di Calcolo della Regione Marche possono essere sintetizzate come segue:

- Il file system distribuito utilizzato in MCloud è *GlusterFS*, configurato per essere utilizzabile sia per archiviare le immagini virtuali nel servizio di image repository in alta affidabilità, sia per definire una storage area comune tra i compute node. L'architettura è stata inoltre disegnata in modo da realizzare un fail-over automatico in caso di problemi a uno dei server *GlusterFS* e in modo da poter essere in seguito integrata in una SAN esterna.
- Visto il basso numero di sistemi fisici del progetto iniziale, l'alta disponibilità è stata affrontata in queste componenti:
  - deve esserci garanzia che problemi ad uno qualunque dei sistemi non compromettano il file system sottostante. Questo è garantito da *GlusterFS*, come descritto sopra.
  - La gestione della rete attraverso il servizio nova-network è stata replicata su tutti i sistemi per evitare problemi di connettività dovuti a malfunzionamenti di uno o più sistemi.
  - Eventuali problemi al Cloud controller di OpenStack non devono avere effetto sulle applicazioni in esecuzione nelle VM.

Questa architettura consente di evitare un'iniziale esplicita configurazione di alta disponibilità basata sulla ridondanza di tutti i sottosistemi di OpenStack (DB, autenticazione, messaggistica, image repository, dashboard, monitoring). Tale tipo di ridondanza porterebbe infatti ad una complessità sproporzionata rispetto alla dimensione del progetto pilota.

La figura 3 mostra come attraverso *GlusterFS*

il volume di *Glance* (image repository) sia replicato 3 volte per ridondanza e come il volume nova (lo spazio disco condiviso tra i compute node che ospita i file delle VM in esecuzione) sia configurato in "replica 2 distribuita", in modo da consentire la live migration sull'infrastruttura ed offrire un volume in alta disponibilità con buone performance di I/O.

In figura 4 sono riportate le configurazioni di rete definite nell'infrastruttura pilota. Seguendo linee guida e best practice definite in OpenStack, la rete è stata organizzata come segue:

- *Management network* (192.168.200.0/24): utilizzata per la comunicazione tra i servizi di OpenStack.
- *Data network* (192.168.122.0/24): utilizzata per assegnare indirizzi IP privati alle VM.
- *External network* (10.101.8.0/24): rete "pubblica", usata per permettere alle VM di comunicare con le reti esterne all'infrastruttura. Da questa sottorete vengono inoltre prelevati gli indirizzi di rete pubblici da usare come "*floating IP*" da assegnare dinamicamente alle VM.

Sull'infrastruttura pilota sono state abilitate le seguenti funzionalità:

- *Volumi persistenti*: attraverso nova-volume è possibile creare un volume persistente e connetterlo a una VM in esecuzione. Il volume è indipendente dalla VM e può essere associato a una sola VM per volta. Le modifiche sul volume vengono mantenute anche dopo la terminazione della VM a cui è collegato;
- *Live Migration*: è la possibilità di migrare una VM da un compute node a un altro senza perdita di connettività. Per poter effettuare questa operazione è necessario che la directory contenente i file delle VM sia condivisa tra tutti i compute node (in MCloud, attraverso *GlusterFS*). La live migration è fondamentale nella gestione di un'infrastruttura Cloud, ad esempio per effettuare attività di manutenzione su un nodo fisico evitando impatto sulle VM in esecuzione.
- *Floating IP*: è la possibilità di associare in maniera dinamica un certo indirizzo IP (chiamato "*floating*") a una VM. Si può scegliere se associare in automatico l'indirizzo alla VM durante la sua creazione oppure aggiungerlo o rimuoverlo manualmente mentre la VM è in esecuzione. Risulta in tal modo semplice sostituire la VM che risponde ad un certo indirizzo pubblico con un'altra VM, ad esempio a causa di un malfunzionamento della prima;
- *Snapshot*: crea una fotografia (*snapshot*) di una VM, inserendo questa nuova immagine nell'*image repository* in modo da poter istanziare nuove VM a partire dallo snapshot creato.

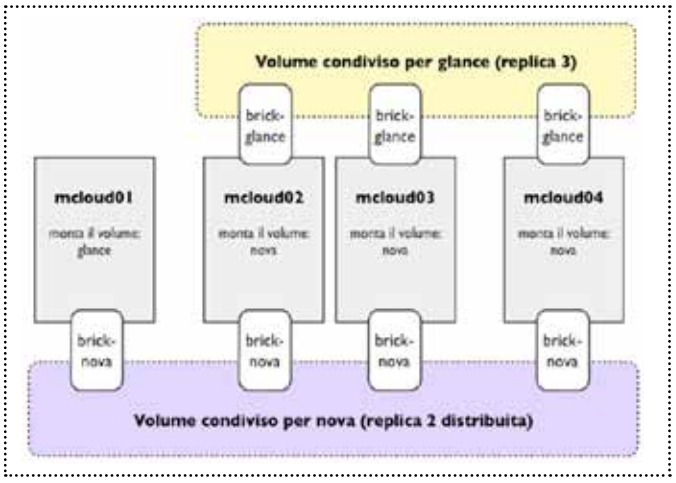


Fig 3 - Organizzazione spazio dati dell'infrastruttura pilota

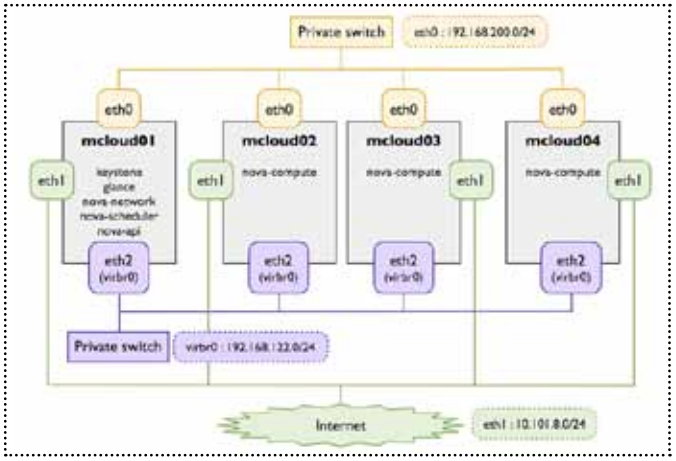


Fig 4 - La configurazione di rete dell'infrastruttura pilota

3. Valutazione delle esigenze e identificazione dei dati utilizzabili

per monitoring e accounting

Analizzando in dettaglio l'architettura di Openstack, sono state identificate due categorie di dati e le rispettive sorgenti di informazioni:

- Infrastrutturali: informazioni derivate da Nova e popolate dal sistema di messaggistica di OpenStack.
- Legati ai consumi delle VM: informazioni gestite direttamente dal sistema di virtualizzazione (KVM), cui è possibile accedere tramite API, plugin specifici e libvirt.

Sono state poi individuate le componenti principali del sistema in termini di:

1. *Data Producer*. Le sorgenti naturali d'informazione possono essere classificate secondo:
   a) Facilities e Services monitoring;
   b) Activities monitoring.
2. *Data Flow*. Nel data flow per monitoring o accounting si identificano i seguenti tre aspetti principali:
   a) Trasporto dei dati;
   b) Archiviazione dei dati;
   c) Monitoring e visualizzazione dei dati.
3. *Formato dei dati*. Esso deve essere sufficientemente flessibile da permettere a nuove applicazioni di agganciarsi al framework.
4. *Architettura*. È stato sviluppato un modulo di monitoring e accounting, integrato nella Dashboard, che può essere scomposto in tre aree (*fig. 5*):
   a) Resource level monitoring;
   b) Alarms;
   c) User level monitoring.

Sono quindi state studiate le applicazioni più diffuse per integrare queste tre aree, specificatamente Ganglia, Collectd e Zenoss come performance monitor e Nagios e Zenoss come notification systems. È stata inoltre studiata una soluzione, basata su Python, che si interfaccia direttamente con la messaggistica interna AMQP e via libvirt con KVM. Nella Tabella 1 sono riassunti i risultati dell'analisi delle soluzioni possibili in termini di funzionalità supportate.

La scelta finale prevede l'uso di Ganglia come misuratore e aggregatore delle metriche disponibili in nova e KVM; Nagios per il sistema di allarmistica; una serie di tool e plugin specifici per il livello applicativo. La scelta della combinazione Ganglia/Nagios, nonostante richieda una personalizzazione delle immagini virtuali, è motivata dal fatto che essi:

1. sono molto diffusi in combinazione con OpenStack;
2. minimizzano lo sforzo di integrazione grazie ai numerosi plugin, tra cui uno specifico di Ganglia [7] che permette il monitor dei servizi nova usando le librerie stesse del modulo;
3. sono facilmente espandibili.

Zenoss, una soluzione più completa, ha un'elevata richiesta di risorse in termini di funzionamento e una mi-

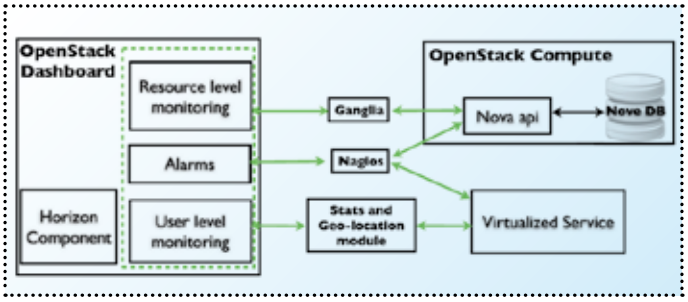


Fig 5 - Architettura delle componenti per il monitoring e accounting

	Performance monitoring	User-friendly Web App	Notifications	Log monitoring	Libvirt plugin	Support of Windows	Plugin for OpenStack	Plugin based metrics	Richness in existing metrics	Popularity
Collectd	X		X	X	X	X		X		
Ganglia	X	X				X	X	X	X	X
Nagios		X	X	X	X	X	X	X		X
Zenoss	X	X	X	X	X	X	X	X		
Own libvirt-based script					X			X		

Tab 1 - Risultati dell'analisi dei tool per il monitoring

nor diffusione in soluzioni basate su OpenStack.

4. Integrazione del monitoring e dell'accounting nella dashboard

Con riferimento ad una delle applicazioni individuate per MCloud (accesso ai referti di analisi mediche), sono stati sviluppati dei plugin adhoc che permettono il monitoring infrastrutturale e applicativo.

Sono state quindi integrate nella dashboard le componenti server di Ganglia e Nagios attraverso la definizione di 3 menu custom:

- "*Resource Level Monitor*" e "Alarms" che mostrano il Web frontend di Ganglia e Nagios;
- "*User Level Monitor*" che permette di visualizzare plot specifici e aggregati per utenza di OpenStack.

Diversi plugin per Ganglia e Nagios sono stati integrati sulle immagini virtuali dalle quali vengono istanziate le VM da monitorare. Sono stati poi configurati un plugin di Ganglia per recuperare e visualizzare informazioni (ricavate mediante interrogazioni al DB MySQL di OpenStack) relative alle VM monitorate e 2 plugin di Nagios:

- apache_usage per controllare le richieste per secondo al server Apache;
- check_log per verificare in modo incrementale la presenza di una certa stringa su un file

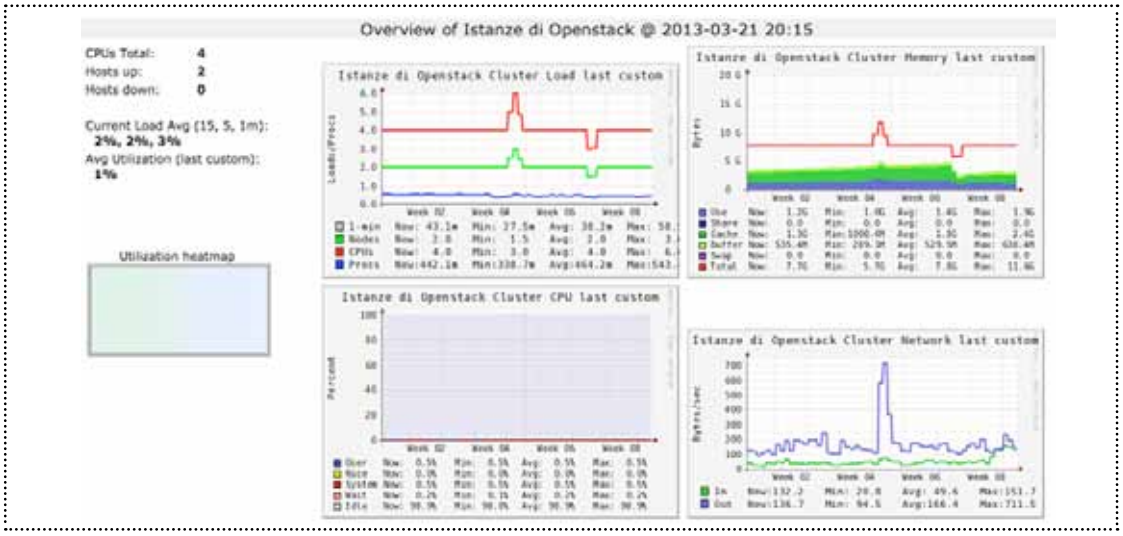


Fig 6 - Media dei consumi delle risorse virtuali.

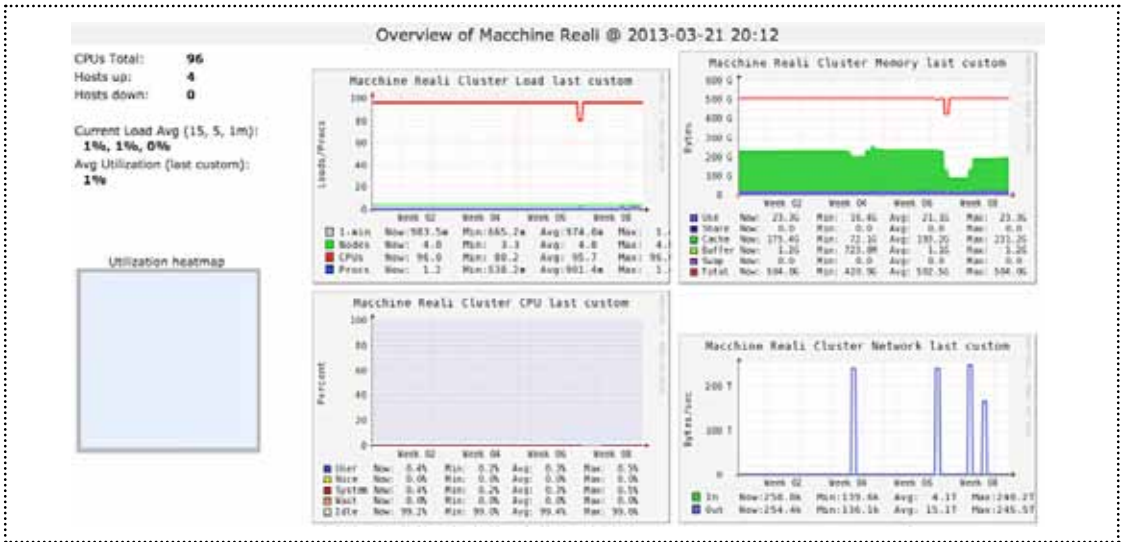


Fig 7 - Media dei consumi delle risorse fisiche

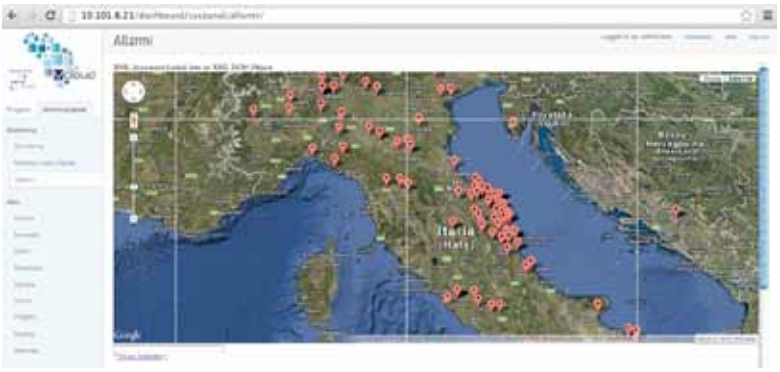


Fig 8 - Dettagli geografico degli accessi dell'utenza.

di log.

Attraverso il modulo di monitoring e accounting realizzato è possibile accedere ai dettagli legati al funzionamento dell'infrastruttura. Le figure 6 e 7 mostrano i dettagli di monitoring relativi al periodo Gennaio-Febbraio 2013 per le VM e per le macchine fisiche, rispettivamente.

Si osserva un'ottima stabilità infrastrutturale, con consumi molto bassi nelle metriche di carico osservate (CPU, Storage, Load e Networking).

Nagios è inoltre utilizzato per monitorare tramite una mappa le posizioni degli utenti che accedono all'applicazione (Fig, 8) attraverso la definizione dei seguenti plugin:

- check_webserver_log eredita dal plugin check_log e notifica quando avvengono connessioni all'applicazione Web.
- check_nagios_log controlla le notifiche mandate dal plugin precedente estraendo indirizzi IP e date di connessione. Questo permette all'applicazione di mappare gli indirizzi IP in un dato periodo temporale.

5. Conclusioni e sviluppi futuri

Nel progetto MCloud è stato realizzato un prototipo di infrastruttura Cloud e sono stati resi operativi alcuni servizi. Durante progettazione e realizzazione sono stati valutati aspetti di compatibilità con più ambienti di virtualizzazione ed analizzati i vantaggi derivanti dall'utilizzo di sistemi aperti con standard che consentono interoperabilità con applicazioni e altre Cloud. In una fase successiva, in cui è previsto che l'infrastruttura venga espansa in modo sostanziale e con l'introduzione di sedi distribuite geograficamente, sarà possibile integrare, anche progressivamente, nuove componenti (es. *object storage*) e funzionalità (es. *network as a service*). Una soluzione di monitoring e allarmistica integrata rende accessibili a livello amministrativo ed utente informazioni su utilizzo e disponibilità di servizi ed infrastruttura; la flessibilità e configurabilità degli strumenti adottati agevolerà infine la definizione, la raccolta e l'analisi delle metriche che nel tempo dovessero rendersi necessarie.

Riferimenti Bibliografici

[1]http://www.camera.it/parlam/leggi/deleghe/05082dl.htm

[2] http://www.openstack.org/

[3]http://www.openstack.org/foundation/companies/

[4] http://www.qyjohn.net/?p=2733

[5] http://docs.openstack.org/trunk/openstack-compute/admin/content/vmware.html

[6]http://ken.pepple.info/openstack/2012/09/25/openstack-folsom-architecture/

[7] https://github.com/ganglia/gmond_python_modules/tree/ master/openstack_monitor

Paolo Veronesi
paolo.veronesii@cnaf.infn.it
Tecnologo presso l'INFN-CNAF, ha partecipato a diversi progetti Europei in ambito Grid Computing, coordina le Operations dell'infrastruttura Grid Italiana nell'ambito del progetto Europeo EGI. Gli interessi principali vertono sulla gestione automatizzata di Data Center e l'alta affidabilità di servizi.

Prototipo per un servizio di cloud storage federato per il mondo accademico e della ricerca

Simon Vocella, Andrea Biancini, Cristiano Valli, Mario Reale, Fabio Farina

Consortium GARR

Abstract. Lo storage personale è una delle categorie di servizio di maggior successo del paradigma cloud. GARR sta svolgendo un progetto di sperimentazione in questa direzione a seguito delle richieste di una parte della propria comunità. In particolare si stanno effettuando studi in merito alle problematiche legate alla creazione di uno storage personale cloud con autenticazione e autorizzazione federata, che abbia elevati standard di resilienza e confidenzialità dei dati e sfrutti al massimo i benefici offerti dalla Federazione IDEM per la gestione delle identità. Il risultato di queste attività è riassunto nell'architettura del servizio GARRbox e nella produzione di un'istanza prototipale. Questo documento discute gli aspetti salienti previsti per il servizio, le lezioni imparate fino ad oggi e disegna le possibili evoluzioni che GARRbox dovrà considerare al fine di rispondere nel miglior modo possibile alle esigenze della Comunità GARR.

1. Introduzione

L'approccio Cloud è al momento la risposta di maggiore successo alla richiesta degli utenti di accedere a servizi complessi in modo semplice e trasparente tramite Internet. Nella definizione proposta dal NIST[1] i servizi Cloud sono fortemente caratterizzati dai seguenti benefici:

- Percezione di risorse illimitate: caratteristica principale del cloud storage è l'espandibilità dinamica delle risorse assegnabili a ciascun utente al variare delle necessità nel tempo.
- Elasticità: le risorse sono assegnate e rilasciate automaticamente in base al carico di lavoro istantaneo, garantendo la continuità del servizio.
- Accesso multi-modale e ubiquo: gli utenti interagiscono con la Cloud utilizzando diversi protocolli e dispositivi di accesso in mobilità, in modo trasparente rispetto ai tecnicismi della soluzione.
- Un modello di business definito: il modello economico cloud è proporzionale alla quantità di risorse consumate.
- Ottimizzazione d'uso delle risorse: il cloud storage model è indipendente dall'hardware utilizzato, permettendo pertanto il riutilizzo di ciò che già si ha (aumento dell'efficienza aziendale sul piano costi/benefici).

La prospettiva di accedere a insiemi di grandi risorse, controllandole direttamente con un costo di utilizzo contenuto, se non marginale, è ovviamente allettante per la comunità della Ricerca Italiana. Per venire incontro a tale richiesta, GARR ha progettato un prototipo di servizio cloud storage chiamato GARRbox, per la sincronizzazione e la condivisione dei dati personali di ricerca in modo semplice, rapido, sicuro, controllabile e indipendente dalla tipologia di risorse hardware a disposizione.

GARRbox rientra nella categoria delle cloud infrastrutturali IaaS e adotta un modello di servizio a cloud pubblica di tipo federato (Community). Il modello Community Cloud, come Helix Nebula [2], rappresenta un'alternativa valida per ridurre sia lo sforzo della messa in opera dei servizi, sia l'impatto finanziario che un unico partner dovrebbe affrontare nel creare un nuovo servizio, velocizzando l'adozione di soluzioni condivise dalla comunità e raggiungendo quindi un bacino di potenziali utenti più ampio.

Una Community Cloud necessita di accordi di federazione, di standard, e di strumenti per l'interoperabilità sia dei piani di controllo delle ri-

Workshop GARR Calcolo e Storage Distribuito - Selected Papers. Roma 29-30 novembre 2012
ISBN: 978-88-905077-4-8 © 2013 Associazione Consortium GARR

sorse sia dei dati. La scelta naturale in questo senso è adottare l'esperienza della federazione IDEM nella gestione delle identità, estendendola alle necessità dello storage cloud federato.

Il contributo descrive l'approccio e i principi che GARR ha seguito per creare un proto-servizio di GARRbox per la propria Direzione, al fine di validare i principi architetturali che saranno estesi per un servizio su scala nazionale.

1.1 Le caratteristiche di GARRbox

GARRbox offre funzionalità simili agli strumenti di sincronizzazione dei dati commerciali come Dropbox, Google Drive e SugarSync. In aggiunta presenta i seguenti benefici:

- È pensato per essere gestito da enti nazionali e sul territorio nazionale, garantendo quindi conformità alle leggi sulla gestione dei dati in materia di privacy, resilienza e copyright.
- È pensato come sforzo comune della comunità R&I Italiana, garantendo economie di scala e condivisione equa dei benefici tra tutti i membri.
- Si basa su un'infrastruttura ad hoc, ma potrà federare eventuali risorse esterne sottoutilizzate, aumentandone l'efficienza e riducendo i costi di messa in opera del servizio, supportando la sostenibilità del servizio sul lungo periodo.
- È sotto l'egida di GARR, a garanzia di equità, dell'apertura agli standard, della certezza che i dati siano preservati e di sostenibilità del servizio per parecchi anni a venire.
- Garantisce confidenzialità e riservatezza dei dati dell'utente finale, tramite l'utilizzo di meccanismi di cifratura.

Come già accennato, GARRbox offre le caratteristiche del paradigma cloud: tramite un piano di controllo unico è possibile accedere in modo semplice ad uno storage distribuito con resilienza e replica dei dati in modo trasparente. Gli utenti percepiscono il servizio come un disco virtuale aggiuntivo o un tool di sincronizzazione, e quando sarà necessario più spazio, potranno ottenerlo in modo elastico.

Da progetto, GARRbox supporta le seguenti tecnologie:

- Autenticazione e autorizzazione federate tramite IDEM.
- Accesso ai dati tramite canali multipli: da browser, cellulare, da riga di comando.

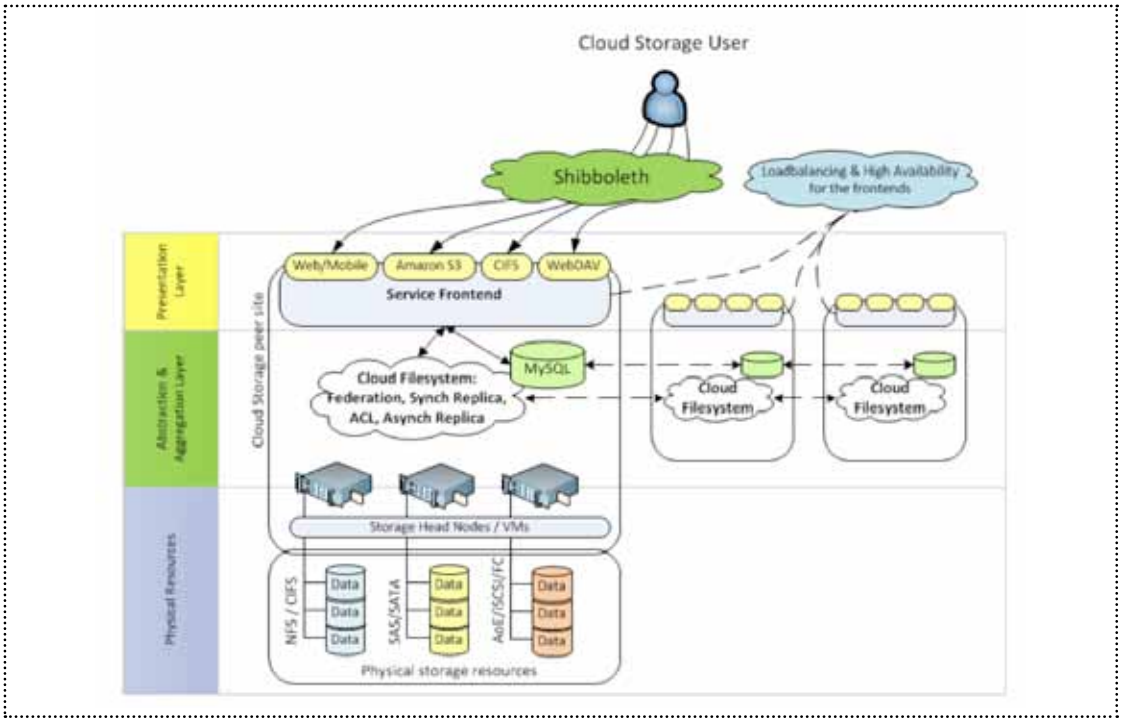


Fig 1 - L'architettura a strati del servizio di cloud storage

- Ridondanza e crittografia dei dati e dei metadati, se richiesto.
- *Logging* e *audit* capillare delle operazioni sui dati.

L'architettura di GARRbox segue il modello *multi-tier* per avere scalabilità, resilienza e supporto di protocolli aperti a ogni strato, semplificando l'adozione e il coinvolgimento di nuovi partner nella federazione. I tre livelli mostrati in Figura 1 sono:

- Le risorse fisiche di storage. GARRbox si basa su risorse di GARR e sulla cooperazione con i membri della Comunità, al fine di mettere a fattor comune risorse distribuite, per garantire replica e bilanciamento.
- Lo strato d'integrazione e federazione, basato su tecnologie di file system cloud distribuiti. Lo strato implementa gli aspetti cloud e di virtualizzazione per garantire l'elasticità delle risorse. Inoltre è responsabile della replicazione dei dati e della loro disponibilità in caso di fallimento.
- I *front-end* con interfaccia multi-canale. Le interfacce spazieranno da portali web a client di sincronizzazione, da applicazioni mobili a protocolli di basso livello con le relative API. Ognuno dei moduli di accesso è integrato con la Federazione IDEM.

Quest'approccio permette di integrare tecnologie eterogenee nei diversi strati, permettendo di restare sempre al passo con le evoluzioni ICT, senza dover mutare il quadro generare del servizio e assicurando in ogni momento ridondanza, alta disponibilità e affidabilità.

1.2 Le differenze tra GARRbox e le offerte commerciali

Molte istituzioni e aziende private sono ancora scettiche nel conservare dati sensibili sulle cloud commerciali. Le ragioni di questa cautela sono principalmente legate ai problemi oggettivi di sicurezza che le cloud pongono. In particolare, i provider commerciali sono in grado di assicurare la persistenza dei dati, ma non sono in grado di garantire il livello di privacy e, in generale, di fiducia necessario per conservare dati sensibili, ad esempio immagini mediche per scopi di ricerca. In aggiunta, i provider commerciali non possono garantire che i dati degli utenti restino dentro i domini imposti dalla giurisdizione Europea .

GARR assicura un alto livello di fiducia alla propria comunità: gestendo la connettività per la Comunità ha già un'ampia esperienza nell'affrontare le problematiche di sicurezza degli utenti, e l'adozione di GARRbox non introduce un nuovo soggetto nella gestione delle informazioni.

GARRbox supera queste limitazioni grazie al ruolo istituzionale di GARR, assicurando che i dati siano ospitati su data center della comunità, con tutti i criteri ottimali di ridondanza a failover. GARRbox assicura che la privacy e le policy di accesso siano regolate tramite IDEM, lasciando agli utenti pieno controllo su come e a chi sono concessi gli accessi ai dati. In aggiunta, i dati potranno essere protetti con i migliori metodi di crittografia, rendendoli inaccessibili anche ai gestori dello storage fisico. Grazie alla Comunità, il servizio non sarà mai soggetto a improvvisi cambiamenti unilaterali degli accordi tra gli utenti e il servizio stesso. Analogamente, GARRbox non sarà soggetto a *vendor lock* a nessun livello, favorendo sempre protocolli e soluzioni aperte.

2. Il prototipo di servizio per la Direzione

Per verificare le scelte architetturali descritte nel paragrafo precedente, abbiamo creato un prototipo di servizio aperto ai soli utenti della Direzione GARR. Il sistema è stato dimensionato per supportare circa trenta utenti, offrendo a ognuno 10 GB di spazio.

Le risorse fisiche sono costituite da dispositivi *off the shelf* accedute tramite POSIX. Per il livello di aggregazione che astrae le risorse fisiche, esponendo un file system resiliente omogeneo, è stato sperimentato il file system distribuito GlusterFS. I dati degli utenti sono replicati in triplice copia localmente in Direzione e ulteriormente mantenuti come backup su un'istanza secondaria ospitata dalla Sezione INFN di Milano-Bicocca. Nello strato di presentazione, ogni in-

terfaccia è realizzata come un modulo indipendente per scalabilità e personalizzazione, ed è implementata come un servizio web che accede ai dati del file system aggregato. Questo strato è anche responsabile delle funzioni di autenticazione e autorizzazione, implementato con Shibboleth SP.

Il prototipo fornisce funzioni di navigazione dei dati tramite un'interfaccia di *file-browsing* per caricare, scaricare e gestire i propri dati su Internet. I file memorizzati sul servizio sono quindi accessibili tramite:

- Un'interfaccia web basata su Ajaxplorer, per accedere ai *repository* e gestire le preferenze, la condivisione e le attività amministrative. Inoltre, il portale espone lo stato della federazione e il monitoraggio.
- Un gateway custom compatibile con il protocollo Amazon S3. L'interfaccia è realizzata per superare la limitazione che il web ha nella gestione delle cartelle annidate. L'interfaccia permette agli utenti di accedere ai propri dati ed eseguirne una migrazione verso provider cloud differenti, utilizzando gli strumenti client standard per l'accesso ai servizi di Amazon.
- Un'interfaccia WebDAV che permette di integrare facilmente le applicazioni preesistenti e i mount point nativi dei diversi sistemi operativi.
- Un'interfaccia di download BitTorrent, che sfruttando funzionalità *peer-to-peer* permette di migliorare la disponibilità di un file, ottimizzando l'utilizzo della banda. Questa interfaccia è stata pensata per aumentare la diffusione di materiale per la divulgazione e la didattica.

2.1 La gestione delle identità

La sicurezza è una questione fondamentale per i servizi Cloud. La certezza che i dati siano accessibili solo dagli utenti che ne hanno diritto è il presupposto fondamentale per le infrastrutture *multi-tenancy*, come i dispositivi fisici sui quali opera il servizio cloud. Il prototipo realizzato affronta sin d'ora queste problematiche adottando gli standard di sicurezza web e integrando i meccanismi di autenticazione e autorizzazione basati su SAML, secondo le disposizioni della Federazione IDEM. Gli attributi sono utilizzati per identificare non solo l'utente, ma anche per estrarre i gruppi cui l'utente appartiene e la sua istituzione. Il prototipo usa queste informazioni per determinare automaticamente la visibilità che l'utente ha sui dati e sui contenuti condivisi. Anche i limiti e le quote di spazio disponibili possono essere definite secondo regole basate su questi attributi. Le regole sono definite dagli amministratori di ogni sito. Le quote sono implementate con una doppia soglia: quando un utente supera la soft-quota riceve una notifica via e-mail. Quando la hard-quota è superata, l'utente non può caricare ulteriori contenuti.

2.2 Condivisione e versioning dei dati

Il prototipo supporta due modelli di condivisione:

- I file possono essere condivisi tramite URL dinamici. L'URL può essere protetto tramite password e può rimanere valido per un tempo limitato specificando una scadenza.
- Le cartelle possono essere condivise con altri utenti del servizio per creare spazi di collaborazione.

A differenza dei servizi di cloud storage pubblici, l'approccio federato consente agli utenti la condivisione delle cartelle anche secondo gli attributi previsti dalla federazione di identità. Nel dettaglio, gli utenti possono condividere le proprie cartelle con un'istituzione o con diversi criteri di raggruppamento, determinati dai valori dei diritti previsti da IDEM. Gli utenti possono specificare quali autorizzazioni hanno i partecipanti alla condivisione sui dati: sola lettura, sola scrittura, o pieno controllo.

Il prototipo prevede anche un sistema di controllo delle versioni per consentire agli utenti di tenere traccia delle modifiche sui file. Il sistema implementa un server GIT [3]: un sistema di controllo di revisione distribuito, che si distingue per efficienza e rapidità. Tramite l'interfaccia web, l'utente gestisce le versioni correnti e precedenti dei dati. Il numero totale di versioni per un file è limitato a un numero fisso e solo le ultime modifiche possono essere ripristina-

te. Questo vincolo è stato introdotto per impedire al numero di versioni di crescere troppo velocemente, penalizzando la reattività e l'efficienza del servizio.

3. Primi feedback degli utenti

Il prototipo è stato rilasciato agli utenti della Direzione il 25 settembre 2012. Ad oggi sono presenti 35 utenti registrati. L'analisi dei log indica che il sistema è acceduto principalmente dall'interfaccia web e che la maggior parte degli utenti usa il sistema con regolarità. Un numero esiguo di utenti (sei) lo utilizza saltuariamente. L'interfaccia compatibile con Amazon è utilizzata dagli utenti più esperti per spostare grandi quantitativi di file organizzati in cartelle annidate. La distribuzione dello spazio occupato mostra il classico andamento a coda lunga, con un utente che ha richiesto più di 10 GB, un ristretto gruppo di utenti che occupa buona parte della quota assegnata e la maggior parte delle utenze abbondantemente sotto la quota assegnata. Le aree di lavoro condiviso create dagli utenti sono 16: questo indica che i nostri utenti prediligono gli strumenti collaborativi offerti. L'architettura a strati ha permesso di minimizzare gli impatti dei malfunzionamenti, avendo un solo incidente risolto in meno di un'ora in sei mesi di sperimentazione (99,85% di uptime). I feedback utente guidano e guideranno le evoluzioni del servizio. In particolare gli utenti richiedono strumenti di sincronizzazione desktop e maggiori informazioni sulle modifiche ai documenti nelle aree condivise.

4. Conclusioni

Questo contributo presenta GARRbox, un'architettura di cloud storage federato per il mondo accademico e della ricerca, e il prototipo con cui GARR ne ha messo alla prova i principi. I risultati della sperimentazione indicano quali siano le tecnologie adatte per estendere il prototipo di servizio, creato per gli utenti della Direzione GARR, e quali aspetti debbano essere rivisti e rafforzati per offrire un servizio scalabile, sicuro ed efficiente alla Comunità.

Quando questi ultimi punti tecnologici saranno migliorati, si inizieranno a definire le politiche di gestione del funzionamento del servizio, in modo da trasformarlo in un servizio per la comunità e in una federazione di cloud storage. Gli sviluppi futuri faranno tesoro delle conoscenze acquisite con il prototipo e dei preziosi suggerimenti forniti dagli utenti. L'attività immediata che è in fase di sviluppo consiste nel miglioramento dell'interfaccia utente, comprensiva di client di sincronizzazione, con feedback più immediati e un migliore controllo delle versioni.

Riferimenti Bibliografici

[1] Mell, P., & Grance, T. (2009, August 8). National Institute of Standards and Technology - Cloud Computing. Retrieved September 4, 2009, from National Institute of Standards and Technology: http://csrc.nist.gov/groups/SNS/cloud-computing/index.html

[2] http://helix-nebula.eu

[3] http://git-scm.com

Simon Vocella
simon.vocella@garr.it
lavora da diversi anni come software developer. Interessato a nuove tecnologie emergenti, in particolare a sistemi distribuiti e tecnologie cloud. Ha collaborato con GARR lavorando nei progetti europei FEDERICA e NOVI e nel progetto cloud storage GARRbox.

Workshop GARR CSD - *Selected papers*

Progetto DECIDE: un esempio di infrastruttura al servizio della comunità biomedica
V. Ardizzone

IGI Portal: portale web di accesso a risorse Grid e Cloud per le comunità scientifiche
M. Bencivenni, D. Michelotto, A. Ceccanti, A. Cristofori, E. Fattibene, G. Misurelli, R. Brunetti, P. Veronesi

Authentication e authorization federate nelle Cloud: estensioni a Shibboleth per l'applicazione in contesti di Cloud Computing
A. Biancini, L. Prete, S. Vocella

GaaS: Grid personalizzate per il calcolo su Cloud
V. Boccia, G.B. Barone, R. Bifulco, D. Bottalico, L. Carracciuolo, R. Canonico

Lo Science Gateway del progetto agINFRA per l'accesso a una data infrastructure per le Scienze Agrarie
R. Bruno, G. Allegri, G. Andronico, R. Barbera, F. Bitelli, A. Budano, A. Calanducci, E. A. C. Costantini, M. Fargetta, A. Fornaia, G. L'Abate, S. Monforte, A. Puliafito, R. Ricceri, F. Ruggieri, D. Saitta, M. Villari

Software-Defined Networking: Esperienze OpenFlow e l'interesse per Cloud
M. Campanella, F. Farina, L. Prete, A. Biancini

Octopus: una Cloud self-service di macchine virtuali
A. Cisternino, M. Davini, M. Mura

Grandi infrastrutture di storage per calcolo ad elevato throughput e Cloud
M. Di Benedetto, A. Cavalli, L. dell'Agnello, M. Favaro, D. Gregori, M. Pezzi, A. Prosperini, P.P Ricci, E. Ronchieri, V. Sapunenko, V. Vagnoni, V. Venturi, G. Zizzi

Cloud Computing in ENEA-GRID: Macchine Virtuali, Roaming Profile e Online Storage
G. Ponti, A. Rocchi, A. Colavincenzo, G. Giannini, A. Secco, G. Bracco, S. Migliori

Distributed Open Cloud Computing, Storage e Network con WNoDeS: Esperienza ed Evoluzione
D. Andreotti, M. Caberletti, V. Ciaschini, G. Dalla Torre, A. Italiano, E. Ronchieri, D. Salomoni

Sull'interoperabilità tra risorse locali, Grid e Cloud per la realizzazione di un'infrastruttura di calcolo distribuito in Italia
D. Scardaci, G. Andronico, R. Barbera, R. Bruno, M. Fargetta, A. Fornaia, G. La Rocca, S. Monforte, R. Ricceri, R. Rotondo, D. Saitta

Realizzazione di un'infrastruttura Cloud pilota basata su OpenStack
L. Fanò Illic, E. Fattibene, M. Manzali, H. Riahi, D. Salomoni, A. Valentini, P. Veronesi, V. Venturi

Prototipo per un servizio di Cloud Storage federato per il mondo accademico e della ricerca
S. Vocella, A. Biancini, C. Valli, M. Reale, F. Farina

ISBN 978-88-905077-4-8