ISSN online: 2282-5665

Consiglio Nazionale delle Ricerche

CERIS ISTITUTO DI RICERCA SULL'IMPRESA E LO SVILUPPO

Novembre

2011

Rapporto tecnico N.39

REPOSITORY E FRONT-END OPEN-SOURCE PER LA CONSERVAZIONE DI OPERE DIGITALI

Giancarlo Birello, Ivano Fucile, Valter Giovanetti, Anna Perin

Consiglio Nazionale delle Ricerche
Istituto di Ricerche sull'Impresa e Lo Sviluppo

RAPPORTO TECNICO CNR-CERIS
Anno 6, N° 39; Novembre 2011

Direttore Responsabile
Secondo Rolfo

Direzione e Redazione
Ceris-Cnr
Istituto di Ricerca sull'Impresa e lo Sviluppo
Via Real Collegio, 30
10024 Moncalieri (Torino), Italy
Tel. +39 011 6824.911
Fax +39 011 6824.966
segreteria@ceris.cnr.it
http://www.ceris.cnr.it

Sede di Roma
Via dei Taurini, 19
00185 Roma, Italy
Tel. 06 49937810
Fax 06 49937884

Sede di Milano
Via Bassini, 15
20121 Milano, Italy
tel. 02 23699501
Fax 02 23699530

Segreteria di redazione
Maria Zittino
m.zittino@ceris.cnr.it

Enrico Viarisio
e.viarisio@ceris.cnr.it

REPOSITORY E FRONT-END OPEN-SOURCE PER LA CONSERVAZIONE DI OPERE DIGITALI[*]

Giancarlo Birello*, Ivano Fucile, Valter Giovanetti
(Cnr-Ceris, Ufficio IT)

Anna Perin
(Ceris-Cnr, Biblioteca)

Ceris-Cnr
Ufficio IT
Strada delle Cacce, 73
10135 Torino – Italy
Tel.: 011 3977533/534/535

Cnr-Ceris
Biblioteca Ceris
Via Real Collegio, 30
10024 Moncalieri (Torino), Italy
Tel.: 011 6824928

*Corresponding author: g.birello@ceris.cnr.it

ABSTRACT: Ceris-CNR IT Office and Ceris-CNR Library are involved in a digitization project promoted by Bess (Electronic Library of Economic and Social Sciences in Piedmont Area) and commissioned to handle all the post-scan of the digitization.
This technical report analyzes the strategies adopted and the main open-source software used.
Ceris-CNR had to provide for the management of large volumes of data with the availability of space storage for the digitized works with characteristics of stability, versatility and dynamism. Ceris-CNR has deployed the software and server platforms of the repository, in a virtualized and redundant infrastructure. Ceris–CNR also take care of the design, development and management of the web portal (front-end) for the presentation, research and consulting data of the digitalized items

KEY WORDS: storage, digital archive, islandora, fedora commons.

[*] Questo lavoro è stato presentato alla Conferenza Garr 2011, "Da 20 anni nel Futuro", Bologna, 8-10 novembre 2011.

INDICE

Indice delle figure

INTRODUZIONE

L'Ufficio IT del Ceris-CNR ha sede presso l'Area di Ricerca CNR di Torino e amministra l'Infrastruttura di rete CNR in Piemonte. Presso la propria sede è situata anche la sala macchine che costituisce il centro-stella delle connessioni e dei servizi di rete per gli organi CNR afferenti, in totale 15 strutture CNR per un'utenza complessiva indicativa di 420 unità di personale.

A livello sistemistico, la migrazione degli ultimi anni della maggior parte dei server e dei servizi di rete su infrastrutture virtualizzate, rende lo storage uno dei punti vitali dell'intera struttura. Infatti, nel caso di macchine virtuali sono file i dischi virtuali dei server e sono file i dischi virtuali sui quali sono memorizzate le informazioni quali email, backup utenti, cartelle di gruppo. Tutte le ultime scelte sono state nella direzione del software open-source, che richiede un lavoro di studio ed approfondimento per individuare ciò che è più adatto a soddisfare le esigenze e consente di contenere i costi.

La Biblioteca del Ceris-CNR si trova presso la sede dell'Istituto a Moncalieri. Suo compito è quello di acquisire, gestire e rendere fruibile il patrimonio bibliografico/statistico dell'Istituto, di offrire supporto alle attività di ricerca e servizio di reference per gli utenti interni. La biblioteca non è aperta al pubblico ma è riservata al personale Ceris (strutturato e non). L'accesso è comunque consentito agli utenti specialistici previo appuntamento.

La biblioteca fa parte di Essper, associazione di 137 biblioteche di istituti di studio e ricerca nell'ambito delle discipline economiche, delle scienze sociali, giuridiche e storiche. La biblioteca ha il proprio catalogo riviste riversato in ACNP (Catalogo italiano dei periodici) e utilizza NILDE (Network Interlibrary Document Exchange) quale metodo privilegiato per il document delivery.

La biblioteca fa anche parte di Bess - Biblioteca Elettronica di Scienze Economiche e Sociali del Piemonte, gruppo di 18 biblioteche e centri di documentazione di scienze economiche e sociali Piemontesi, che con il sostegno della Compagnia di San Paolo, ha avviato un progetto di digitalizzazione di materiale relativo all'economia e ai processi di trasformazione sociale ed economica della regione Piemonte presente nelle biblioteche del gruppo e di altri enti Piemontesi che hanno dimostrato interesse per l'iniziativa. La finalità di conservazione e preservazione di materiale di rilevante importanza, sia di difficile reperimento perché fuori stampa che relativo a letteratura grigia, è abbinato alla costruzione di una rete di soggetti depositari che desiderino collaborare allo sviluppo di tale repository.

Il Ceris-CNR nell'ambito di questo progetto è stato incaricato di occuparsi di gestire tutta la parte successiva alla digitalizzazione a cominciare con la disponibilità di spazio di memorizzazione per il deposito delle opere digitalizzate adatto alla gestione di grossi volumi di dati, con caratteristiche di stabilità, versatilità e dinamicità, realizzato tramite storage in cluster a 2-nodi attivo/passivo in HA (alta affidabilità). Per proseguire con lo sviluppo delle piattaforme server e dei software con i relativi adattamenti per il progetto specifico, inserite su infrastrutture virtualizzate e ridondate che includono la progettazione e implementazione del sistema di repository, dedicato a mantenere, indicizzare e collegare i dati prodotti dalla digitalizzazione. Infine la progettazione, sviluppo e gestione del portale web (front-end) per la presentazione, ricerca e consultazione dei dati, in grado inoltre di gestire le policy di accesso ai dati e le relative autorizzazioni. E' ovviamente anche compresa la manutenzione ordinaria e consulenza continuativa sulle problematiche sistemistiche e del software.

1 ARCHITETTURA

L'architettura complessiva presenta una certa complessità, come si può rilevare dalla figura, riportandoci alla mente l'analogia con un puzzle.

Primo aspetto la varietà e la quantità di tecnologie coinvolte che hanno richiesto approfondita analisi per essere applicate e adattate alle nostre esigenze. Partiamo dalle nozioni base sistemistiche di clustering per la realizzazione dello storage ridondato e di virtualizzazione fino ad arrivare agli aspetti specifici di gestione di applicativi java e php, il tutto farcito da standard quali xml e xslt unito ad aspetti più prettamente bibliografici quali i metadati Dublin Core, applicati a oggetti digitali, immagini e testo, inseriti in modelli legati da relazioni semantiche.

Altro aspetto interessante, che ci riporta all'analogia con un puzzle, è l'utilizzo quasi esclusivo di soluzioni open-source, che da un lato permettono un notevole risparmio economico e la possibilità di sfruttare quanto di più innovativo è disponibile nella comunità, ma per contro richiedono un grosso lavoro di ricerca e adattamento delle varie componenti per ottenere e soddisfare le nostre esigenze specifiche.

Procediamo ora alla descrizione dell'architettura suddividendola in tre strati: quello hardware, quello specifico dei file system e infine quello applicativo.

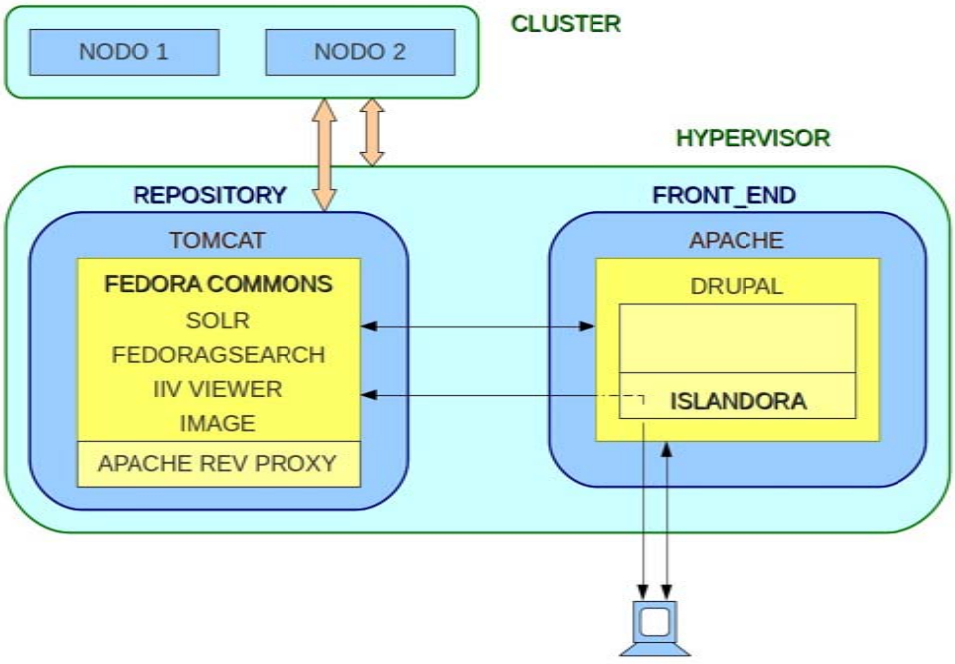

Figura 1: Architettura complessiva

1.1 Hardware

Le principali apparecchiature coinvolte nello sviluppo del progetto sono i server dedicati alla realizzazione del cluster e quelli impiegati come hypervisor per ospitare le macchine virtuali. Sono esclusi da questa trattazione i vari apparati di rete coinvolti nelle interconnessioni quali gli switch, i firewall ed ogni altro elemento relativo l'infrastruttura di rete.

1.1.1 Storage

Il cluster a due nodi attivo/passivo è stato completamente realizzato da noi (cfr. “Storage in HA: cluster attivo/passivo open-source ”, Rapporto Tecnico CNR-CERIS, Anno 6, N 37; Giugno 2011) sfruttando solo software open-source che ci ha permesso di concentrare la spesa sugli apparati e in particolare sull'espansione del numero di hard-disk per ottenere la capacità di memorizzazione richiesta.

Lo storage rende disponibili i vari volumi in cui è suddiviso tramite protocollo iSCSI. Nei test ha dato ottimi risultati di prestazioni ed affidabilità nella specifica architettura in cui inserito, offrendo spazio disco per il backup delle macchine virtuali e lo spazio di memorizzazione degli oggetti digitali, gestito da Fedora Commons, così come gli indici del sistema di ricerca Solr.

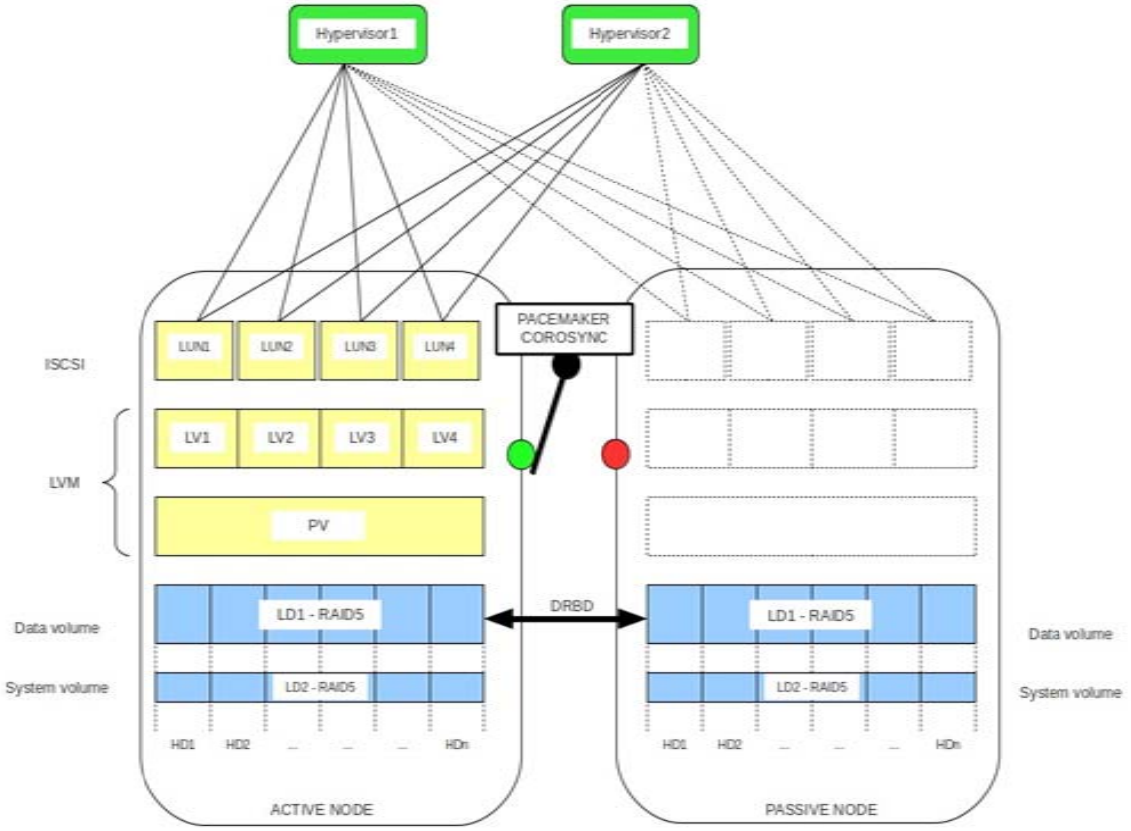

Figura 2: Cluster 2 nodi attivo/passivo

1.1.2 Hypervisor

Unica componente non open-source, ma comunque con licenza free, è l'hypervisor, di cui al momento non è prevista una migrazione verso un prodotto di diversa natura. Il sistema che ospita le macchine virtuali è ridondato, cioè esistono due apparati di cui uno è in produzione e l'altro, di caratteristiche simili, disponibile per migrare le macchine virtuali presenti sul primo nel caso di guasto.

Entrambi gli apparati sono connessi allo stesso volume di backup sul cluster e dal quale posso essere avviate le macchine virtuali in caso di necessità senza doverle spostare fisicamente sui dischi locali. Tramite script vengono fatti i backup delle macchine virtuali in produzione, consistente nella duplicazione dell'immagine del disco di sistema di ogni server, nella maggior parte dei casi si tratta di backup a caldo, ossia senza interruzioni del server e dei suoi servizi.

Il server che ospita il repository, oltre alla partizione di sistema, è connesso a partizioni direttamente sul cluster, in questo caso il backup consiste sempre nella sola immagine del disco di sistema, per ovvie ragioni di spazio e di inutilità, essendo le partizioni sul cluster già ridondate implicitamente.

1.2 Filesystem

La diversità dei vari filesystem chiamati in causa dalle molteplici applicazioni e dai sistemi coinvolti nella realizzazione del progetto così come il loro posizionamento merita una nota particolare. Si va dalle implementazioni in grado di supportare il clustering agli store annidati all'interno di altri file system per la gestione degli oggetti.

In particolare analizziamo la struttura di virtualizzazione e di memorizzazione dei dati del repository.

1.2.1 Macchine virtuali e backup

Gli hypervisor sono installati su server dotati di uno storage contenuto ma di elevata affidabilità, dischi SAS (Serial-attached SCSI) in mirror che garantiscono oltretutto un certo livello di prestazioni. Su questo storage locale sono di norma memorizzati i filesystem virtuali delle macchine ospitate sull'hypervisor, in particolare la partizione di sistema dei vari server.

Gli hypervisor sono connessi inoltre ad alcune partizioni sul cluster a loro dedicate, tramite protocollo iSCSI, potendo quindi usufruire, oltre dello storage locale, anche di quello reso disponibile dal cluster. In questo modo lo storage remoto può essere utilizzato sia direttamente dall'hypervisor stesso che dalle macchine virtuali ospitate, previa formattazione dello spazio con il filesystem, in questo caso proprietario, usato dal sistema di virtualizzazione, il cui supporto del clustering ne garantisce l'accesso contemporaneo da più sistemi.

Il backup delle macchine virtuali, cioè dei file che contengono le partizioni di sistema, avviene tramite uno script, “ghettoVCB” proveniente dal mondo open-source, programmato giornalmente e settimanalmente sull'hypervisor, che permette fare una copia a caldo sullo storage remoto in tempi ragionevolmente brevi, si parla di un paio di minuti per un disco virtuale da 8GB.

La copia ottenuta risiede sul cluster ed è a tutti gli effetti pronta per essere eventualmente eseguita, agganciandola ad una macchina virtuale, senza doverla spostare sullo storage locale. In questo modo abbiamo a disposizione una copia aggiornata del server che può essere utile per prove ma soprattutto sarà fondamentale nel caso di guasti seri al server originario permettendo il ripristino dei servizi in pochissimo tempo.

Nel caso di server virtuali che necessitano di spazio particolarmente ampio, come ad esempio il server di posta elettronica o quello di backup degli utenti, oltre al disco di sistema, gli possono essere assegnati ulteriori dischi virtuali di dimensioni opportune, memorizzati sullo storage remoto invece che su quello locale, ma sempre in forma di file, all'interno del filesystem dell'hypervisor, rappresentazione del disco virtuale.

1.2.2 Dati

Una soluzione diversa è invece stata adottata per lo spazio dati del repository oggetto del progetto, aggiuntivo a quello di sistema. Le ragioni principali sono le dimensioni richieste che superano quelle massime gestite dalla coppia filesystem e iSCSI dell'hypervisor e le prestazioni. In questo caso è direttamente il server virtuale a collegarsi tramite iSCSI ai volumi offerti dal cluster senza intermediazioni. Il filesystem con il quale viene formattato lo spazio remoto è a discrezione del sistema operativo del server e può quindi essere utilizzato uno qualsiasi o il più opportuno a seconda del caso, tenendo presente che lo spazio sarà acceduto solo e sempre da un un'unica macchina alla volta, non è richiesto un sistema con supporto del clustering.

Su questa base sono memorizzati gli oggetti del repository, assimilati a dei BLOB (Binary Large Objects) e in quanto tali è opportuna una forma efficiente di lettura e scrittura, ragione per cui, il repository utilizza un sistema di store specifico per questi oggetti. In particolare Fedora Commons utilizza Akubra data store, un progetto open-source di un'interfaccia per la memorizzazione di file adattabile ai principali sistemi di storage e tesa al più alto livello di interoperabilità tra i diversi sistemi. Per Akubra un Blob è un bitstream di lunghezza finita dotato di un identificativo e un Blob Store è concepito principalmente per fornire l'accesso in lettura e scrittura di un Blob.

Le opzioni di configurazione del repository permettono poter posizionare su una partizione qualsiasi del sistema lo store Akubra, nel nostro caso individuato su una partizione coincidente col volume iSCSI dello storage remoto. In questo modo abbiamo garantita l'affidabilità del sistema, la ridondanza dell'informazione e una veloce procedura di ripristino in caso di guasto del server, essendo sufficiente connettere al volume iSCSI un nuovo server che avrà a disposizione l'intero store del repository.

1.3 Applicazioni

Tutti gli applicativi utilizzati nel progetto richiederebbero una trattazione a parte per la loro importanza ed estensione, ci limiteremo quindi a segnalare delle principali applicazioni gli aspetti più rilevanti ai fini del progetto.

Innanzitutto abbiamo scelto di separare e distribuire su due server distinti le componenti relative al repository da quelle di presentazione del front-end. In questo modo si dovrebbe ottenere un miglioramento delle prestazioni generali del sistema ma soprattutto una semplificazione delle operazioni di mantenimento dei sistemi potendo agire in modo autonomo sui due server.

La suddivisione risponde anche alla logica di separare le applicazioni java da quelle php: le prime inserite in un unico contenitore Tomcat sul server del repository e le seconde dentro Apache sul secondo server previsto per la parte di front-end. Questa semplificazione permessa dalla natura propria delle applicazioni vede come unica eccezione l'installazione, sul server del repository, del web server Apache con la funzione di reverse-proxy per le applicazioni installate in Tomcat.

Riguardo il server Tomcat che ospita le numerose applicazioni java, si è scelto di non utilizzare il componente incluso nella distribuzione di Fedora Commons, ma di installarne uno esterno incluso nei pacchetti della distribuzione linux del server. Questa scelta ha come ragioni le maggiori prestazioni che si possono ottenere, come anche consigliato nei manuali d'installazione di Fedora Commons, oltre la semplicità di manutenzione del componente stesso essendo legato alla distribuzione del server e potendo così sfruttare le patch fornite con il sistema operativo.

All'interno di Tomcat sul server del repository troviamo varie applicazioni, che qui elenchiamo per completezza, delle quali solo alcune verranno trattate più nello specifico, in particolare abbiamo:

- adore-djatoka, trattamento immagini
- fedora, il repository server
- fedoragsearch, il motore di ricerca di fedora
- fop, formattazione e rendering
- iiv, il viewer on-line dei libri
- imagemanip, trattamento immagini
- saxon, trasformazioni xslt
- solr, motore di ricerca e indicizzazione

1.3.1 Repository: Fedora Commons

Il cuore dell'intero progetto è sicuramente il repository, cioè il sistema di memorizzazione e gestione degli oggetti digitali. Dopo aver preso in considerazione varie soluzioni, Fedora Commons è stato il prodotto prescelto, sicuramente di spicco tra tutte le soluzioni possibili, sia per la diffusione che ha in questo settore sia per le numerose potenzialità che offre.

Il nome è l'acronimo di “Flexible Extensible Digital Object Repository Architecture”, il software è stato sviluppato in Java ed è frutto della comunità open-source, la cui fervida attività dimostra l'interesse diffuso esistente su questo prodotto, dagli aspetti più interni del sistema a tutte le varie applicazioni sviluppate a contorno.

Senza scendere nei particolari dell'architettura, cerchiamo di dare una panoramica su quello che Fedora Commons offre, in particolare la prima nota riguarda il fatto che il prodotto in sé è una base astratta che definisce un contesto e delle regole per la gestione di oggetti digitali. Queste assunzioni iniziali sono fatte pensando a sistemi di conservazione di opere digitali a lungo termine, in grado di gestire grossi volumi di dati e in modo flessibile per poter trattare i più svariati tipi di oggetti.

Nel repository gli elementi base sono gli oggetti che possiedono uno o più componenti chiamati “datastream”, i quali sono a loro volta dei contenuti, ad esempio un'immagine, oppure metadati che descrivono l'oggetto. I datastream possono essere memorizzati localmente sul server oppure referenziati tramite un url esterno. Gli oggetti posso dichiarare una o più relazioni con altri oggetti all'interno del repository, tramite asserzioni semantiche del tipo soggetto-predicato-complemento, che costituisce sicuramente una delle caratteristiche più evolute e con grandi potenzialità che offre questa architettura.

Quanto descritto finora abbinato alla possibilità di generare datastream virtuali, ad esempio la generazione di una miniatura da un'immagine ad alta definizione, in aggiunta all'identificazione univoca tramite namespace e Persistent Identifier (PID) degli oggetti, conferiscono al prodotto il carattere di repository durevole nel tempo di contenuti fruibili via web.

L'architettura di questo sistema è stata pensata per poter offrire la definizione personalizzata da parte dell'utente di modelli di oggetti, restando ferme le caratteristiche sopra elencate, potendo così sfruttare le potenzialità dell'ambiente adattato alle esigenze specifiche della soluzione. Un esempio tipo è proprio la nostra implementazione dove vengono utilizzati modelli quali collezioni, libri e pagine legati da semplici relazioni come “è membro della collezione” o “è parte di”, oggetti che prevedono datastream funzione di quanto il sistema di scansione produce, ossia immagini ad alta risoluzione, testo prodotto dal sistema di OCR e file pdf ricercabili.

Possiamo quindi concludere, tralasciando l'approfondimento di parecchie altre caratteristiche di cui è dotato Fedora Commons, con un paio di ulteriori funzioni interessanti per la nostra applicazione. La prima è la possibilità di effettuare l'harvesting degli oggetti con interrogazioni OAI-PMH (Open Archives Initiative Protocol for Metadata Harvesting), importante nell'ottica di condivisione dei contenuti del repository con altre entità. La seconda è la disponibilità di funzioni richiamabili da script per la gestione dell'intero repository, che includono le operazioni di manutenzione e quelle di inserimento e rimozione di oggetti, semplificando così la fase di ingesting dei contenuti e rendendo l'operazione effettuabile in automatico, sicuramente molto utile se si pensa ad esempio al nostro caso dove un oggetto corrisponde ad una singola pagina di ogni singolo volume.

1.3.2 Front-end: Drupal

Come interfaccia web verso gli utenti per presentare gli oggetti digitali, nel nostro caso libri e riviste, è stato scelto Drupal, un CMS (Content Management System) open-source, largamente diffuso e già utilizzato in rete per l'accesso a repository.

La scelta è stata dettata fondamentalmente dalla disponibilità del framework Islandora, che tratteremo nel seguito, che prevede proprio Drupal come base di sviluppo ed integrazione tra il repository ed il front-end verso gli utenti.

Su questo prodotto non c'è molto da dire, ovvero è largamente conosciuto e diffuso come piattaforma di presentazione web e non sarà oggetto del nostro lavoro il suo approfondimento, vogliamo solamente evidenziare come, rispetto ad altre soluzioni, sia più versatile ed aperto, soprattutto per chi voglia integrarlo e personalizzarlo con moduli aggiuntivi come fatto da Islandora.

Dal punto di vista della sicurezza Drupal prevede moduli per l’autenticazione ed autorizzazione in un ottica di possibili integrazioni dell'applicazione nel caso siano richieste policy diverse di accesso ai vari contenuti. Queste funzionalità hanno le potenzialità di integrarsi con le policy del repository, che permettono un controllo granulare dell'accesso ai singoli datastream, così come con i modelli ed il framework Islandora, lasciando aperta anche questa possibilità per gli sviluppi futuri.

1.3.3 Framework: Islandora

Il repository Fedora Commons è un'ottima base per sviluppare applicazioni inerenti la conservazione e presentazione di oggetti digitali, infatti sono molte le applicazioni disponibili nella comunità che si poggiano su tale sistema, tra tutte ci è parsa più conveniente e vicina alle nostre esigenze Islandora, un framework open-source sviluppato dalla biblioteca Robertson della UPEI (University of Prince Edward Island, Canada).

Il framework, sviluppato dalla collaborazione di informatici e bibliotecari, nell'ottica di utilizzare al meglio le risorse disponibili nella comunità, costituisce un sistema completo perfettamente integrato di congiunzione e coordinamento tra il repository Fedora Commons e il CMS Drupal, rendendo quest'ultimo l'interfaccia tramite la quale amministrare e presentare i contenuti del repository.

Nel nostro caso non abbiamo utilizzato tutte le potenzialità che offre Islandora, ci siamo limitati a utilizzare solo alcune delle funzioni, in particolare quelle di presentazione, che per la loro impostazione “open”, ci hanno permesso modifiche e integrazioni con una certa agilità, ottenendo alcune variazioni per noi importanti, come ad esempio l'utilizzo dei soli matadatiDublin Core o l'aggiunta e disponibilità di un indice di ogni volume non inizialmente previsto dal framework.

Senza voler anche in questo caso scendere in troppi particolari, il framework è costituito principalmente da due componenti: dei moduli lato Drupal che mettono in comunicazione il front-end ed il repository e dei modelli lato repository costituiti da oggetti e relativi datastream in grado di caratterizzare gli oggetti inseriti secondo uno schema specifico che rende semplice la presentazione al pubblico. Avere modelli direttamente nel repository presenta un'ulteriore particolarità, quella che nessuna informazione relativa agli oggetti è memorizza nel database di Drupal o in nessun altra parte del front-end, ma tutto risiede sotto forma di datastream ed oggetti all'interno del repository stesso.

La vivace comunità di Islandora ha prodotto, tra i vari moduli e componenti, oltre a quelli direttamente dedicati alla comunicazione con il repository, un paio di elementi particolarmente interessanti per il nostro progetto: il viewer ed il sistema di ricerca. Mentre del secondo specificheremo meglio gli aspetti nel paragrafo seguente, il viewer è un componente sviluppato in java che permette la lettura on-line dei libri senza dover far ricorso a soluzioni esterne. Il viewer viene richiamato direttamente dal repository sotto forma di datastream virtuale e produce la visualizzazione, eventualmente integrata all'interno di una pagina del server web, delle singole pagine di un libro abbinate al relativo testo prodotto dall'OCR, potendo eventualmente effettuare lo zoom delle immagini e navigare tra le pagine del volume.

Figura 3: Viewer on-line

1.3.4 Piattaforma di ricerca: Solr

Merita sicuramente un cenno specifico la piattaforma di indicizzazione e ricerca, che costituisce per questo tipo di progetti sicuramente uno dei componenti più apprezzati ed utili per l'utente finale. La scelta è ricaduta su Solr, parte del progetto Apache Lucene, soprattutto per l'integrazione disponibile nei componenti prescelti come repository e front-end.

In Fedora Commons è disponibile un applicativo aggiuntivo “Fedora Generic Search Service”, integrato come servizio web all'interno di Tomcat, che offre preziose funzioni di indicizzazione e ricerca sfruttando come piattaforma proprio Solr ed integrandosi con il repository fino a rendere automatici gli aggiornamenti degli indici a seguito dell'ingesting di nuovi oggetti. Mentre tra i vari moduli sviluppati dalla comunità di Islandora, ne esiste uno specifico per integrare nel sito web di Drupal la ricerca tramite Solr all'interno del repository con tutte le principali funzioni richieste.

E' stato necessario personalizzare lo schema e la configurazione di Solr per adattarla alla nostra situazione, in particolare per definire quali datastream indicizzare e di quali oggetti, oltre al tipo di indicizzazione. Comunque è stato uno dei pochi interventi richiesti per ottenere l'indicizzazione full-text dei datastream relativi ai testi dei volumi e quella a parole dei metadati Dublin Core. Il risultato ottenuto permette dal sito web la ricerca per parole chiave nei dati descrittivi delle opere con possibilità di filtri suggeriti dallo stesso sistema di indicizzazione (facet) ed eventualmente abbinata o in alternativa la ricerca full-text nel contenuto dei libri. Per quest'ultima sono stati adattati i moduli Islandora originari e modificati per produrre a video il risultato della ricerca, con le parti di testo in cui sono state ritrovate le parole che vengono evidenziate rispetto il resto del documento, e il collegamento alla pagina specifica del libro.

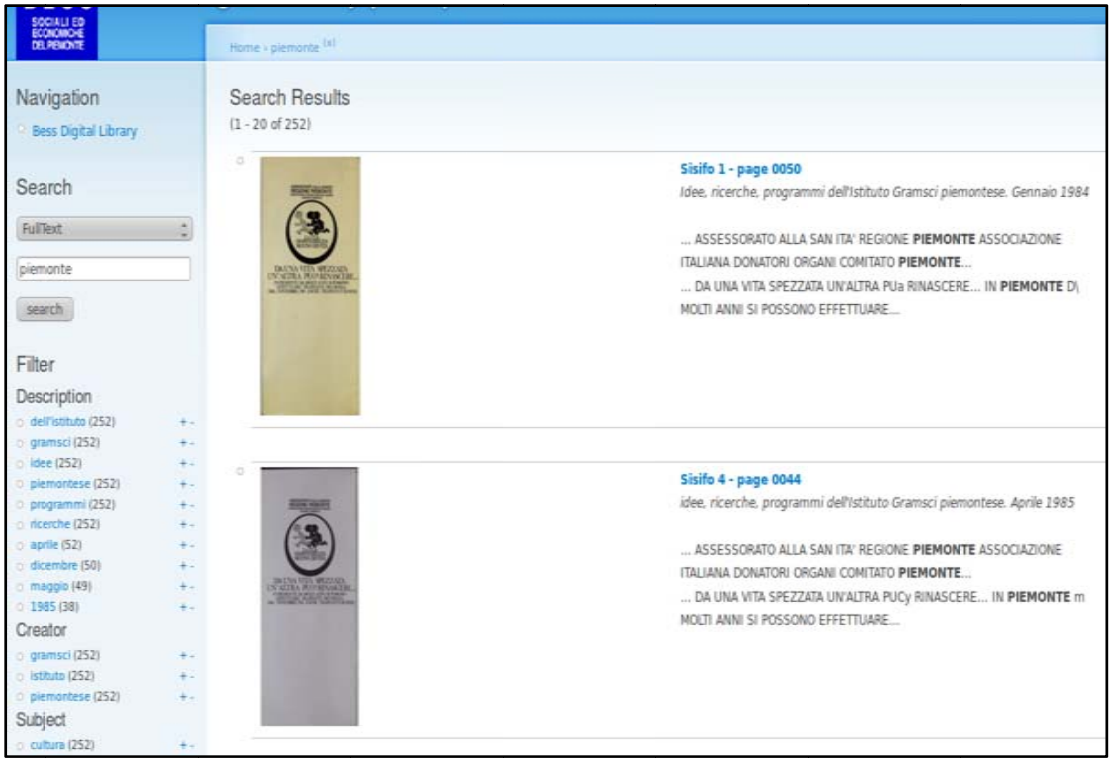

Figura 4: Ricerca full-text e filtri

2 INFRASTRUTTURA DI RETE

Infine concludiamo con quello che è il collante di tutta l'architettura e delle applicazioni chiamate in causa finora: l'infrastruttura di rete. In effetti qualsiasi oggetto del nostro puzzle non avrebbe ragione d'essere e non potrebbe funzionare se non possedesse una connessione in rete che costituisce il mezzo fondamentale di comunicazione a tutti i livelli dell'applicazione.

Si può osservare come il nocciolo del cluster sia proprio una replica in rete di volumi di dati, i servizi interni sono raggiungibili quasi esclusivamente via rete, ad esempio quelli ospitati in Tomcat, per non parlare dell'evidente necessità di connettività internet per il front-end per essere interrogato e acceduto da qualsiasi utente. Il tutto inserito all'interno di una rete locale suddivisa in VLAN tra le quali sono distribuiti i vari componenti a seconda della funzione che ricoprono, per esempio esiste una VLAN per la sola comunicazione tra gli hypervisor e i volumi iSCSI del cluster, così come una zona dedicata ad ospitare i server che offrono un qualche tipo di servizio per l'esterno.

Tra le caratteristiche che meritano un accenno abbiamo la raggiungibilità tramite protocollo IPv6 e non solo IPv4 delle applicazioni pubbliche, tra le quali includiamo il repository stesso oltre al sito web. Inoltre possiamo sottolineare il lavoro di sicurezza che inevitabilmente è richiesto per questo tipo di esposizione dei servizi, intervenendo sia a livello di singolo server, ad esempio chiudendo all'esterno applicazioni Tomcat e rendendo raggiungibili a mezzo reverse-proxy quelle richieste, sia a livello di infrastruttura operando sulle configurazioni e sui filtri disponibili sugli apparati di rete quali il firewall.

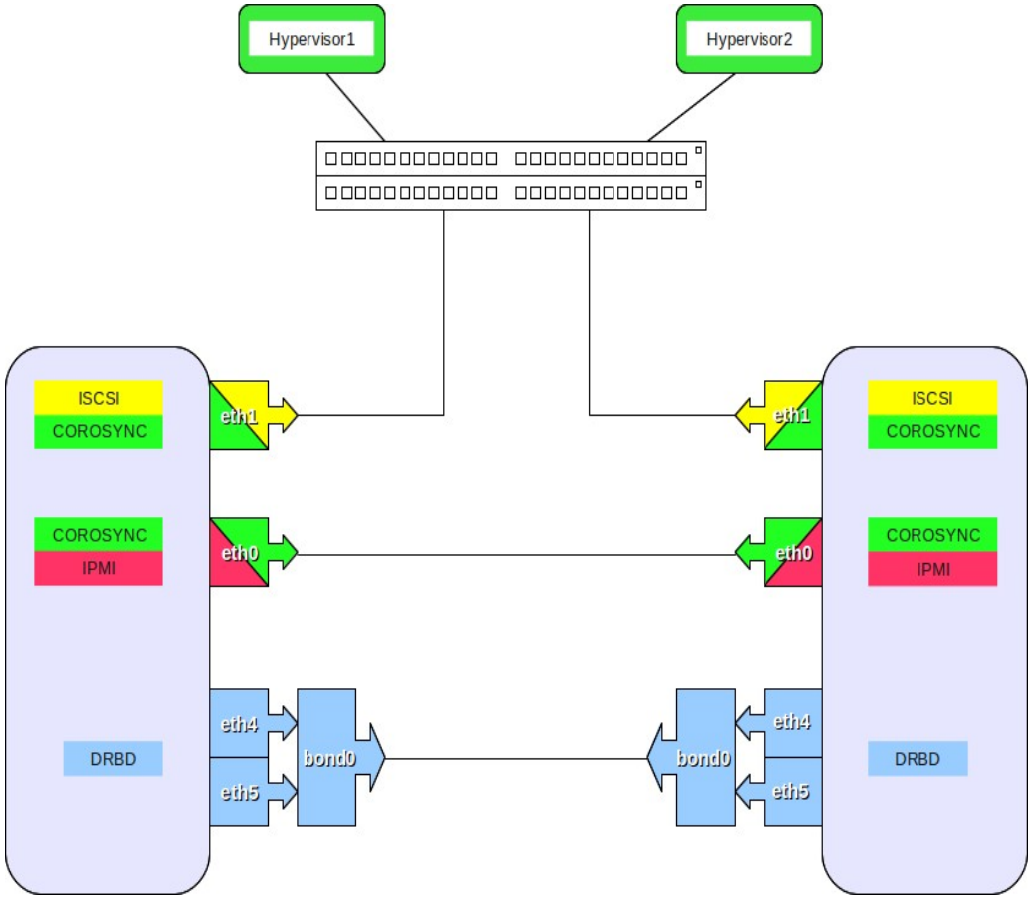

Figura 5: Collegamenti di rete del cluster

3 CONCLUSIONI

Il lavoro svolto è stato molto appassionante e costruttivo, ci ha permesso approfondire e conoscere tecnologie nuove, con la sensazione di aver sviluppato qualcosa che va oltre il solo settore informatico avvicinandosi ad altri ambienti come quello delle biblioteche e della condivisione di opere digitali.

Uno degli obiettivi che ci eravamo prefissati era quello di produrre qualcosa che, in sintonia con l'attingere dalla comunità open-source, fosse aperto come soluzione e di facile replica in altri contesti e riutilizzabile eventualmente da altri progetti. Direi che possiamo considerarci soddisfatti su questo punto dato che già è in corso una collaborazione con un altro organo del CNR dove stiamo replicando l'intera applicazione, nella soluzione a due server, con una certa facilità grazie alle competenze locali ed alla documentazione prodotta passo a passo nella realizzazione del sistema.

Siamo comunque solo all'inizio del progetto, il tempo che resta potrà essere utilizzato per affinare meglio alcune soluzioni adottate, ad esempio si potrebbe lavorare sul viewer o pensare addirittura allo sviluppo di uno nuovo se ci saranno le risorse necessarie, così come lavorare sul front-end per migliorare alcuni aspetti della presentazione delle opere.

Manca al momento la possibilità di restringere l'accesso in modo granulare ad alcune opere, eventualmente quelle coperte ancora da copyright, anche se tutte le soluzioni e tecnologie adottate possiedono i requisiti necessari e le funzionalità per realizzarla, ma pensiamo di introdurla solo nel caso fosse chiaramente richiesto, preferendo sicuramente l'impostazione open library.

Riassumendo, il progetto ha raggiunto un buon livello di sviluppo, soprattutto sono state definite in modo chiaro e preciso architettura e distribuzione dei servizi, basi solide sulle quali lavorare, nostro obiettivo è anche rendere disponibile l'esperienza fin qui sviluppata per eventualmente collaborare con chi interessato nello sviluppo di applicazioni analoghe.

BIBLIOGRAFIA

[1] Apache Sorl, http://lucene.apache.org/solr/, visitato ottobre 2011

[2] Apache Tomcat, http://tomcat.apache.org/, visitato ottobre 2011

[3] Bess - biblioteca elettronica di scienze sociali ed economiche del Piemonte, http://www.bess-piemonte.it, visitato ottobre 2011

[4] Bess Repository, http://demo.bess-piemonte.it, visitato ottobre 2011

[5] DRBD, Software development for high availability clusters, http://www.drbd.org/, visitato ottobre 2011

[6] Drupal Come for the software, stay for the community, http://drupal.org/, visitato ottobre 2011

[7] Dublin Core MetadataInitiative, http://dublincore.org/, visitato ottobre 2011

[8] Fedora Commons software repository, http://fedora-commons.org/, visitato ottobre 2011

[9] Infrastruttura di rete del CNR in Piemonte, http://www.to.cnr.it, visitato ottobre 2011

[10] Islandora, building a rich digital repository ecosystem, http://islandora.ca/, visitato ottobre 2011

[11] Islandora forum sviluppatori, https://github.com/islandora, visitato ottobre 2011

[12] Islandora guide, https://wiki.duraspace.org/display/ISLANDORA/Islandora, visitato ottobre 2011

[13] OAI-PMH The Open Archives Initiative Protocol for Metadata Harvesting, http://www.openarchives.org/OAI/openarchivesprotocol.html, visitato ottobre 2011

[14] Rapporto tecnico Ceris-CNR n.37 Storage in HA: cluster attivo/passivo open source. Giancarlo Birello, Ivano Fucile, Valter Giovanetti, Anna Perin, giugno 2011

Working Paper Cnr-Ceris

ISSN (*print*): 1591-0709 ISSN (*on line*): 2036-8216

Download

↓

http://www.ceris.cnr.it/index.php?option=com_content&task=section&id=4&Itemid=64

Hard copies are available on request,
please, write to:

Cnr-Ceris
Via Real Collegio, n. 30
10024 Moncalieri (Torino), Italy
Tel. +39 011 6824.911 Fax +39 011 6824.966
segreteria@ceris.cnr.it http://www.ceris.cnr.it