



### ALCUNE TECNICHE PER GRID E DINTORNI

Alberto Ciampa<sup>1</sup>, Silvia Arezzini<sup>1</sup>, Dario Fabiani<sup>1</sup>, Enrico Mazzoni<sup>1</sup>

<sup>1</sup>INFN - Sezione di Pisa, Largo B. Pontecorvo, 3, I-56127 Pisa, Italy

#### Abstract

Nel presente lavoro verranno descritte le seguenti tecniche realizzate e messe in produzione nel GRID Data Center della Sezione di Pisa dell'INFN:

- Utilizzo di *chroot* per l'implementazione dei worker node e dei server di middleware di GRID. L'obiettivo principale che si vuole perseguire è il disaccoppiamento tra l'hardware (in generale il sistema sottostante) ed il server che gira l'applicativo. Per questa strada si può ottenere anche un altro importante risultato per i worker node: la possibilità di offrire diverse configurazioni e tipologie di server, anche con controllo dinamico. Riferendosi al punto successivo, si possono gestire dinamicamente configurazioni per worker node e nodi per code locali.
- Coesistenza di *code locali e GRID* sulla stessa farm con possibilità di dare accesso alle code locali anche ad una comunità remotamente distribuita. Il primo punto ha l'obiettivo di estendere idealmente la modalità di condivisione di risorse anche al di fuori del paradigma GRID, tendendo ad includere quegli utenti che presentano esigenze più spostate verso l'interattivo. Il secondo punto, che rappresenta un esempio dell'utilizzo e dell'utilità del progetto AAI, tende a dare alle code locali lo stesso tipo di canale di fruizione che caratterizza le code GRID. L'infrastruttura AAI utilizzata per dare accesso remoto alle code locali può addirittura essere utilizzata per accedere alle stesse code GRID.
- L'utilizzo di *GPFS come file system sottostante ad un SRM* (a Pisa abbiamo in produzione sia dCache che StoRM), permette il facile import/export di file tra aree sotto SRM e aree accedute direttamente via Posix. Questo è un tassello importante nell'integrazione dell'ambiente GRID con l'ambiente delle code locali, specialmente se pensiamo ad una realizzazione coerente di una struttura che metta risorse in condivisione tra i Tier2 e i futuri Tier3.

## Indice

<b>1</b>	<b>chroot</b>	<b>3</b>
1.1	Implementazione attuale . . . . .	3
1.2	Possibili sviluppi . . . . .	5
<b>2</b>	<b>Integrazione farm locali</b>	<b>6</b>
2.1	Integrazione delle code locali e GRID . . . . .	6
2.2	Accesso remoto alle code locali: AAI . . . . .	8
2.2.1	Caratteristiche di INFN-AAI . . . . .	10
2.2.2	Sistemi di Autenticazione e Autorizzazione . . . . .	11
2.2.3	Perchè non si può fare a meno di INFN-AAI . . . . .	14
<b>3</b>	<b>Infrastruttura di storage</b>	<b>15</b>
3.1	Introduzione a GPFS . . . . .	16
3.2	Farm di esperimento . . . . .	18
3.3	Struttura GRID . . . . .	18
3.4	Cluster ad alte prestazioni . . . . .	19
3.5	Descrizione della nuova infrastruttura di storage . . . . .	19
3.6	Considerazioni finali sullo Storage . . . . .	21
<b>4</b>	<b>Conclusioni</b>	<b>22</b>

## 1 chroot

### 1.1 Implementazione attuale

Nella gestione di una farm di qualunque tipo essa sia, ma in particolare se di tipo GRID, è frequente imbattersi in una serie di problemi tra i quali i seguenti occupano un posto preminente:

1. gestione di hardware relativamente moderno rispetto alla versione di sistema operativo richiesta, sia dalle necessità del middleware GRID che da quelle degli applicativi utente;
2. gestione degli aggiornamenti, in particolare per la mancanza di possibilità di roll-back in caso di problemi conseguenti all'aggiornamento stesso;
3. utilizzo dinamico delle risorse, cioè la possibilità di cambiare la versione di sistema operativo e/o software che il server utilizza;

una risposta a questo tipo di problematiche è costituita dall'uso di sistemi di virtualizzazione. Questo approccio, se da un lato permette di risolvere le questioni su enunciate, dall'altro aggiunge overhead, per quanto piccoli, all'uso delle risorse computazionali. Inoltre tipicamente si tende ad installare sui server un sistema Linux avanzato all'interno del quale attivare lo strato di virtualizzazione per utilizzare altri sistemi Linux e quindi sostanzialmente per virtualizzare ben poche funzionalità. Ben diverso è il caso in cui si virtualizza un sistema operativo completamente diverso e quindi con funzionalità e caratteristiche completamente diverse. Ecco perchè riteniamo "intollerabile" pagare un overhead di questo tipo per il funzionamento dell'infrastruttura.

Alla luce di quanto sopra detto in INFN-Pisa abbiamo deciso di utilizzare il meccanismo del chroot per risolvere i problemi sopra citati e non pagare l'overhead derivante da un'infrastruttura di virtualizzazione tradizionale. L'idea è stata quella di installare su tutte le macchine della infrastruttura del Grid Data Center un sistema Linux di tipo professionale, nello specifico Suse Linux Enterprise Server (SLES) 10sp2, in versione minimale e lanciare quindi i servizi GRID all'interno di un chroot.

"chroot", contrazione di change root, è un metodo di sicurezza solitamente usato per isolare i limiti operativi di una applicazione. Normalmente, infatti, un software può accedere a tutti i dischi e le risorse del sistema operativo, compatibilmente con i permessi; l'operazione di chroot consiste nell'eseguire il programma bloccandolo dentro una sotto-directory e permettendogli così di accedere solo alle risorse di cui ha strettamente bisogno. È un metodo usato solitamente per motivi legati alla sicurezza di un sistema, infatti, se un hacker riesce a prendere il controllo di una applicazione, avrà tutti i privilegi associati

all'applicazione e potrebbe essere in grado di compromettere la sicurezza dell'intero sistema, ad esempio installando un rootkit. Al contrario se il programma compromesso si trova in un chroot l'hacker non potrà uscire dal suo perimetro, potrà cambiare il contenuto di file del programma stesso, ma senza poter accedere ad altre risorse.

L'estensione di questo metodo oltre i limiti specifici di protezione permette di rendere disponibile alle applicazioni utente un intero ambiente differente da quello nativo del sistema, lasciando al kernel nativo la responsabilità della gestione dei device e dei filesystem. Se, infatti, si esegue un servizio di login tipo "ssh" all'interno del chroot tutte le connessioni che saranno eseguite attraverso tale servizio avranno disponibile solo l'ambiente del chroot. Cosa analoga accade per un job che venga eseguito da un sistema di batch che "giri" all'interno del chroot. Sono queste le caratteristiche su cui abbiamo basato la nostra implementazione. L'utilizzo del meccanismo di chroot per questo tipo di applicazioni è già stata riportata nel 2008 dal Jozef Stefan Institute durante la periodica riunione autunnale di HEPiX svoltasi a Taipei (<sup>1</sup>).

Mediante l'uso di "chroot" nel nostro ambiente operativo si ottiene che:

- il supporto dell'hardware viene demandato alla SLES e quindi con pieno supporto visto che la grande maggioranza dei produttori certificano le loro piattaforme in questo ambiente o comunque ne rilasciano i driver;
- il supporto dei filesystem viene demandato alla SLES che a sua volta li rende disponibili o con opportuno mount all'interno del chroot o attraverso l'opzione "bind" nel caso il filesystem sia condiviso fra più sistemi. Di nuovo in questo modo si guadagna la certificazione dei produttori nel caso di filesystem proprietari come ad esempio GPFS;
- il sistema che rappresenta il servizio GRID si riduce ad un tar file che viene decompresso in una area disco del sistema installato con SLES. In questo modo è piuttosto semplice sia implementare politiche di rollback, perché basta farsi una copia dell'area disco prima di eseguire aggiornamenti, sia mantenere versioni diverse di sistema operativo e/o applicativi sullo stesso hardware commutando dall'una all'altra versione senza bisogno di riavviare il sistema stesso;
- si semplifica enormemente l'installazione e la gestione della infrastruttura. Il sistema SLES è installato in versione minimale e la sua configurazione è identica su tutti quanti i nodi quindi l'automatizzazione della sua installazione risulta praticabile in maniera assai semplice. L'eventuale sistema ospitato invece si riduce ad un file

---

<sup>1</sup> Andrej Filipčič, Jozef Stefan Institute, Slovenia,

<http://indico.twgrid.org/getFile.py/access?contribId=90&sessionId=36&resId=0&materialId=slides&confId=471>

da scompattare e quindi una volta eseguita la sua “installazione” e configurazione basta crearsene una copia campione da replicare su tutti i sistemi funzionalmente equivalenti.

## 1.2 Possibili sviluppi

Ci sono due possibili linee di sviluppo dell’architettura, entrambe interessanti. L’obiettivo è quello di sfruttare il meccanismo per implementare sulla stessa piattaforma hardware versioni diverse di Worker Node e quindi avere la possibilità di usarle in contemporanea allocando in modo dinamico alle varie versioni le risorse hardware disponibili. Per esempio potremmo installare sul solito hardware:

- nodo grid standard;
- nodo grid glexec;
- nodo per coda locale 32bit puro;
- nodo XXX con caratteristiche speciali;

Già con l’implementazione attuale potremmo installare queste diverse versioni sullo stesso nodo fisico, semplicemente mettendo le copie dei vari ambienti in folder diversi ed effettuando poi chroot su quello che interessa. In questo modo però ne avremmo funzionante solo uno alla volta, mentre il nostro obiettivo è quello di disporre delle diverse versioni contemporaneamente. Come accennato, ci sono due possibili strade.

1. superare il semplice chroot e utilizzare il concetto dei container grazie al software OpenVZ. In questo modo si può attivare su ogni nodo fisico una “macchina virtuale” chroot per ciascuna delle tipologie che interessano. Ciascuna tipologia avrebbe il proprio indirizzo IP e per ciascuna tipologia si definirebbe un cluster LSF a sè stante a cui i vari utenti punterebbero per accedere al tipo di risorse di loro interesse. Su ciascun nodo fisico, poi, si potrebbe gestire il numero totale di job slot disponibili dividendoli in maniera statica fra le varie configurazioni attraverso la relativa configurazione LSF, oppure implementando un meccanismo dinamico grazie al sistema di pre-exec di LSF.
2. mantenere l’attuale approccio chroot, con differenti possibili nodi predisposti, spostando però il servizio LSF da dentro il chroot per farlo eseguire al sistema vero (SLES nel nostro caso). In questo modo sarebbe la configurazione delle code LSF a prendersi carico di intercettare il comando dell’utente e ad incapsularlo in un opportuno script per eseguire tale comando all’interno del chroot giusto. In questo modo

si avrebbe un solo cluster LSF e le varie tipologie di nodo si rispecchierebbero in diverse code ciascuna con il suo wrapper.

La soluzione 2 può essere vista come una generalizzazione della soluzione di WN on demand sviluppata al CNAF ma senza l'overhead del sistema di virtualizzazione (grazie alla soluzione chroot). Inoltre, eseguendo il sistema di batch all'esterno del chroot non si hanno problemi relativi alle gestione delle licenze di LSF perché il sistema di batch ha la visione corretta dell'hardware a prescindere della allocazione di risorse ai vari "nodi virtuali".

## **2 Integrazione farm locali**

### **2.1 Integrazione delle code locali e GRID**

Le resistenze maggiori da parte dell'utenza nel confluire in una architettura completamente gridificata sono riassumibili in:

1. esigenze computazionali particolari (che non trovano facilmente soluzione attraverso l'utilizzo di GRID) tipicamente legate alla versione di Linux da utilizzare, a particolari esigenze hardware (per esempio elevata richiesta di RAM) o alla lunghezza dei job da sottomettere;
2. difficoltà nel migrare il proprio ambiente di lavoro dall'ambito della farm locale a quello GRID;
3. necessità di un lavoro quasi interattivo, nel senso che si vuol sottomettere il proprio job e averne il risultato nell'arco della giornata lavorativa.

questi problemi (o supposti tali) fanno sì che restino al momento in funzione farm specifiche di gruppo con relativo aggravio di lavoro per la loro manutenzione. Paradossalmente però l'utenza accede a queste risorse attraverso meccanismi batch e quindi con modalità molto simili a quelle GRID native.

Da un punto di vista gestionale e di massimizzazione dello sfruttamento delle risorse computazionali sarebbe utile poter integrare quanto più possibile tutte le risorse, sia GRID che specifiche di gruppo, in un'unica struttura. Lavorando su questo fronte e sfruttando il fatto che, da un punto di vista logico, una farm GRID altro non è che un insieme di Computing Element che sottomettono i propri job ad una farm dotata di opportuno sistema di batch si è potuta creare una configurazione integrata fra i due mondi. L'idea è quella di utilizzare l'infrastruttura LSF, su cui si appoggia GRID, suddividendo le risorse hardware in varie partizioni, ciascuna con il proprio fair share, da dedicare alle varie attività. In

questo modo si affianca alle code GRID standard (una per ciascuna VO supportata) una coda da dedicare alla sottomissione locale ed aperta agli utenti locali che confluiscono nella struttura.

Utilizzando LSF si hanno due strumenti per modificare il comportamento FIFO, il fair share e la priorità:

- **fair share:** si può definire in maniera esclusiva per gruppi di host o per coda e determina il peso relativo dell'uso delle risorse da parte degli utenti abilitati ad usare la risorsa stessa. Quindi, se lo si definisce per gruppi di macchine, permette di controllare quanti job slot (fra quelli erogabili dal gruppo di server) possono essere usati da un certo gruppo di utenti a prescindere dalla coda tramite cui il job arriva sul server. Se invece lo si definisce per coda si fa una cosa analoga, ma agendo sui job che sono in una singola coda, perciò non regolando l'utilizzo della risorsa "job slot" ma regolando solo la priorità degli utenti all'interno della coda. Pertanto in un ambiente come GRID, in cui la risorsa da gestire è il "job slot" e ogni comunità di utenti ha la sua coda, è indispensabile implementare il fair share applicandolo al gruppo di host.
- **priorità:** serve per definire l'ordine fra le code. In situazioni in cui LSF ha tutte le code cariche è attraverso la priorità che si decide da quale coda andare a prendere il job che può entrare non appena si libera un "job slot".

È chiaro quindi che l'accesso di un job ad un certo job slot dipende dalla combinazione dei due meccanismi, possiamo anzi dire che, in un certo senso, LSF si calcola una priorità rolling per i vari gruppi di host. Questa priorità rolling tiene conto del pregresso (per ottemperare ai vincoli di fair share) e della priorità delle code. Ed è in base a questa "priorità" calcolata che si decide se e chi deve occupare il job slot libero.

Veniamo quindi al nostro caso specifico. Noi abbiamo due gruppi di risorse:

- H1 risorse GRID su cui è definito il fair share FS1;
- H2 risorse per accesso locale su cui è definito il fair share FS2;

e due gruppi diversi di code;

- Q1 sono le code per GRID, quindi una per ciascuna VO supportata;
- Q2 è la coda per l'accesso tramite sottomissione locale (tutti gli utenti che ne hanno diritto finiscono qui dentro).

Per implementare la soluzione integrata con code GRID e locali, e utenti GRID e locali, abbiamo combinato le caratteristiche nella maniera più funzionale alle nostre necessità accoppiando risorse e code in questo modo: Q1+H1+H2 e Q2+H2. Più nel dettaglio

1. i job grid possono finire sui server dedicati alla coda locale ma regolati dal FS2 per cui se una VO ha delle risorse locali userà il suo fair share su quel gruppo di risorse, se non ne ha userà quello che è lasciato libero per la comunità;
2. i job locali rimangono limitati alle risorse ad essi dedicati con il loro fair share;
3. mettendo la coda Q2 a più alta priorità rispetto a quelle Q1 appena si libera un job slot di quelli delle risorse H2 verrà preferito un job locale ad uno grid;
4. non avendo la coda Q2 possibilità di girare su risorse H1 non succederà che la sua più alta priorità porti job locali ad impadronirsi di risorse GRID.

## **2.2 Accesso remoto alle code locali: AAI**

Il paradigma GRID permette l'uso di risorse condivise ad una comunità distribuita. A nessuno sfugge la potenza di questo paradigma dal punto di vista degli accessi. Grazie ad una opportuna infrastruttura, quella dei certificati X509 e delle Virtual Organization, risulta agevolmente possibile ad un utente scientifico di un qualunque organismo mondiale accreditato usufruire delle risorse GRID messe a disposizione da una qualunque delle sedi mondiali accreditate ad offrire risorse.

Questo mezzo così potente non si rivela però particolarmente appropriato per risolvere le problematiche derivanti dalla necessità, in molti casi, di integrare le risorse messe a disposizione world wide da GRID e le risorse locali. Un esempio tipico è dato dalle risorse di storage. Capita, su scala locale ad esempio, di desiderare di poter accedere agli spazi globali GRID, “anche” con mezzi diversi cioè tipicamente come se si trattasse “anche” di spazio disco non globale, ma locale. A Pisa, per ovviare a questo tipo di problemi e rispondere alla necessità degli utenti locali che desideravano un accesso facilitato allo spazio disco è stato da tempo studiato un meccanismo di “remapping” locale degli utenti GRID per far sì che quando a Pisa atterra un job di un utente di Pisa UID e GID del job che gira sotto la coda batch vengano trasformati in UID e GID dell'utente locale di Pisa. In questo modo l'output ha la possibilità di essere scritto sugli spazi opportuni con UID e GID di un utente locale e l'utente locale, dotato di opportune credenziali risulta in grado di accedere al file anche come se si trattasse di un file locale e non solo di un file generato dal sistema GRID. Per come il meccanismo è implementato, al momento il remapping avviene solo per la parte GID (vedi successivo Cap. 3), ma è estensibile anche alla parte UID.

Analogamente, quando sono state introdotte le code locali sul sistema di batch LSF, si è fatto in modo che potessero coesistere job provenienti da GRID e job locali mettendo in piedi un sistema che permette, con diverse procedure di accesso, di sottomettere job allo stesso insieme di core disponibili per il calcolo (vedi paragrafo 2.1). Questo meccanismo si attua perciò realizzando di fatto una integrazione tra il sistema di AA (Autenticazione ed Autorizzazione) locale e il sistema di AA di GRID. Le operazioni che si svolgono sono funzionali all’ottenere il risultato desiderato che è quello dell’accesso contemporaneo su due scale differenti: quella GRID e quella locale, senza modificare in alcun modo nessuno dei due meccanismi.

Nel caso dei Tier3 o in quello di una facility di calcolo come quella che si sta implementando a Pisa per il GRUPPO IV, familiarmente denominata SuperCluster, si aggiunge un’altra difficoltà. E cioè la necessità di considerare come utenti locali anche utenti che non lo sono, perché i T3 o il SuperCluster si collocano ad un ulteriore livello di distribuzione rispetto a GRID:

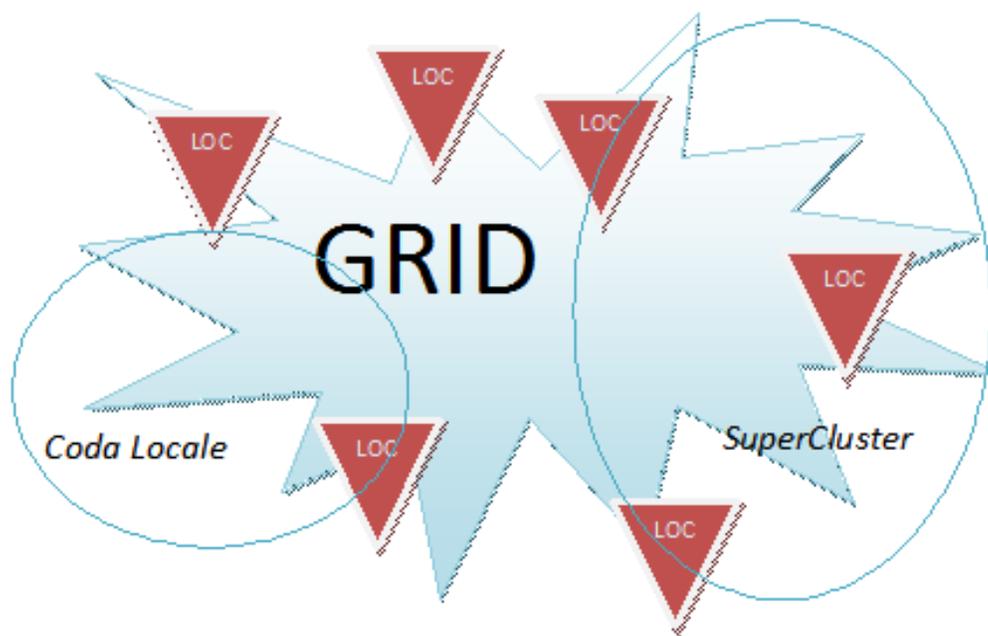


Figura 1: Schema architetturale del sito grid INFN-Pisa

E a questo punto viene in aiuto una infrastruttura nazionale, denominata INFN-AAI che sta per entrare in funzione in maniera completa e che al momento è utilizzata in una sua forma ridotta denominata “Proto AAI”.

### 2.2.1 Caratteristiche di INFN-AAI

INFN-AAI è un progetto di Commissione Calcolo e Reti del 2008 (<sup>2</sup> <sup>3</sup> <sup>4</sup> <sup>5</sup>).

Il progetto ha come suo scopo principale quello di definire e implementare una Infrastruttura di Autenticazione ed Autorizzazione (AAI) unica per l'INFN, basata su Autenticazione Kerberos5 (e PKI, cioè certificati X509) ed Autorizzazione attraverso un Directory Service LDAP.

L'infrastruttura risultante permette di:

1. offrire uno strumento utilizzabile per l'accesso alle applicazioni centralizzate INFN;
2. presentare l'INFN come unica entità nella partecipazione a federazioni di AAI;
3. fornire alle singole sedi uno strumento unico in grado di sostituire efficacemente i vari sistemi di Autenticazione ed Autorizzazione in uso, armonizzandoli in una unica Infrastruttura, integrata con quella nazionale.

Oggi che si è ormai vicini all'implementazione di INFN-AAI nelle varie sedi, si possono cogliere anche altri benefici derivanti dall'adozione di questa infrastruttura.

In particolare è possibile prevedere senza difficoltà un uso INFN-WIDE di identità e credenziali anche per l'accesso a utilità di calcolo scientifico.

Come già evidenziato in precedenza si potrebbe obiettare a questa affermazione facendo notare che GRID, per sua natura offre già questa disponibilità INFN-wide, anzi world-wide, ma questo è vero solo se il paradigma utilizzato è esclusivamente quello GRID. L'accesso a risorse di storage o l'implementazione di code locali determina la necessità di disporre di un maggiore e più specializzato numero di attributi di quelli disponibili nativamente in GRID.

Da qui l'idea di utilizzare l'infrastruttura INFN-AAI al fine di garantire l'accesso a code locali di una singola sede da parte di una comunità dislocata su varie sedi INFN.

INFN-AAI mette a disposizione di ogni sede INFN un albero LDAP all'interno del quale sono registrati gli utenti locali. Questo albero locale è in realtà un ramo dell'albero LDAP nazionale INFN.

Questo significa che qualunque utente INFN è inserito in questo "contenitore" e vi è univocamente definito e riconosciuto. Non importa a quale titolo vi sia presente, può

---

<sup>2</sup>CCR-24/2008/P: Progetto INFN-AAI: Conceptual Design Report Scritto da E.M.V. Fasanelli et al.

<sup>3</sup>CCR-27/2008/P: Progetto INFN-AAI: Dettagli richiesti dal comitato di revisione (Addendum 1) Scritto da E.M.V. Fasanelli et al.

<sup>4</sup>CCR-28/2008/P: Progetto INFN-AAI: Documento di sintesi del CDR (Addendum 2) Scritto da E.M.V. Fasanelli et al.

<sup>5</sup>CCR-31/2009/P: INFN-AAI TDR: Core Services Scritto da E. M. V. Fasanelli et al.

essere un dipendente, un associato, ma anche un ospite temporaneo. Se ha un rapporto di lavoro con l'INFN compare in questo albero LDAP.

Vi compare con tutta una serie di caratteristiche la cui definizione è molto flessibile. È possibile cioè prevedere l'implementazione di attributi ad hoc, sulla base delle necessità delle applicazioni che fanno uso dell'albero.

In una sede, di solito, si usa l'albero LDAP locale per permettere agli utenti locali di accedere le macchine locali. Ma, dato che l'infrastruttura è nazionale, nulla vieta che per una specifica macchina o applicazione si possa abilitare l'accesso NON della comunità locale, ma di una comunità più vasta, ad esempio di quella che ha un particolare attributo (l'autorizzazione ad accedere ad un gruppo di code).

Per come è strutturato e gestito l'albero LDAP è possibile definire uno o più manager che gestiscono l'attributo in questione per tutti gli utenti.

Quindi tutto il personale INFN, se dotato di uno specifico attributo, può accedere ad una utility di una qualunque sede. Basterà che la sede definisca quale attributo identifica l'accesso e predisponga localmente l'ambiente per concretizzare l'accesso di uno qualsiasi degli utenti, quelli locali e quelli non locali.

### *2.2.2 Sistemi di Autenticazione e Autorizzazione*

Scendiamo ora un po' più nel dettaglio delle caratteristiche dei sistemi di AA con i quali ci si confronta e dei possibili meccanismi di integrazione per gli usi di calcolo scientifico.

Una AAI fornisce una infrastruttura di Autenticazione ed Autorizzazione. Autenticazione significa identificazione, mentre Autorizzazione significa accesso e attribuzione di ruoli.

La parte di Autenticazione di INFN-AAI nel progetto originale è demandata a Kerberos5, ma fin dalla prima stesura, quella approvata, è stato inserito come ulteriore sistema di autenticazione PKI cioè i certificati X509. È stato sviluppato inoltre un opportuno "plug-in" che permette la "kerberizzazione" a livello LDAP di tutto ciò che non si può kerberizzare a livello di applicazione. Di fatto, quindi, tramite INFN-AAI si ha un sistema molto completo di autenticazioni sicure. L'autorizzazione è demandata invece alla struttura LDAP.

La prima implementazione di INFN-AAI, denominata "proto-AAI" ha in realtà aggiunto anche un'altra forma di autenticazione: LDAP stesso, che conservando username e password fornisce comunque la possibilità di identificazione.

Le applicazioni centralizzate al momento non effettuano altro che l'accesso. Se si è in LDAP, tramite uno dei sistemi previsti (kerberos5, certificato o username/password) si è prima identificati, poi autorizzati cioè riconosciuti come appartenenti all'insieme giusto

ed è possibile quindi “entrare” cioè accedere al servizio. Non solo si entra, ma si è anche riconosciuti per quello che si è grazie ad un opportuno mapping. Per l’attribuzione dei ruoli in questo momento non si recupera nulla da LDAP. I ruoli che si utilizzano per certe applicazioni centralizzate tipo preventivi o assegnazioni, si prendono da un DB ulteriore, gestito dal servizio DATAWEB. In questo DB al momento da un punto di vista anagrafico risiede il master degli associati INFN e una copia dei dipendenti. Solo queste due categorie infatti possono accedere alle applicazioni più pregiate.

Una funzione di INFN-AAI, cosa questa che la fa denominare al momento INFN-AAI Plus per differenziarla dal progetto originario, è la possibilità di essere utilizzata anche come “Provider di Identità”, cioè non solo come il contenitore di tutte le identità INFN, ma anche come lo strumento tramite il quale si ha la possibilità di riempire il contenitore stesso. Questo strumento si chiama GODiVA (Gestione Ospiti, Dipendenti, Visitatori e Associati) e permette di fatto di realizzare una fotografia, momento per momento del personale INFN.

GODiVA infatti deriva i DIPENDENTI da HR, il DB ufficiale dei dipendenti INFN (con indicazione anche del periodo nel quale si appartiene alla struttura per i dipendenti a tempo determinato). Deriva poi gli associati dal DB gestito dal servizio DATAWEB, importando anche i loro ruoli di appartenenza a gruppi e responsabilità. Quest’ultima funzionalità è però temporanea perchè il servizio DATAWEB utilizzerà a regime direttamente il GODiVA-DB abbandonando il DB attuale. Per Visitatori e Ospiti, anagrafica e ruoli vengono infine scritti direttamente sul server LDAP.

Per il Calcolo Scientifico, che nell’INFN è praticamente GRID, l’accesso naturale sembrerebbe, come già visto la struttura di AA di GRID. Vediamo un po’ più in dettaglio in cosa consiste. C’è una infrastruttura che rilascia certificati X509 che servono per la parte di identificazione dell’utenza. Coloro ai quali viene rilasciato un certificato X509 dalla Certification Authority dell’INFN hanno la possibilità di usare le risorse GRID. Infatti, una volta che si è in possesso di un certificato X509, si può fare richiesta di essere iscritti in un ulteriore registro nel quale si viene collegati a degli attributi. Il registro è quello delle Virtual Organization e lo strumento per la gestione è VOMS. Di fatto dentro a VOMS si scrive che un utente appartiene a un certo gruppo e/o sottogruppo di un gruppo e se l’utente è il manager di una certa VO.

È chiaro a questo punto che un dipendente o associato INFN in effetti può trovare scritte le sue informazioni in due strutture parallele: INFN AAI e X509/VOMS. Esiste in realtà un punto di contatto, ma nulla esclude che ne possano esistere altri, anzi varrebbe la pena di crearne e realizzarne di nuovi con l’intento di offrire all’utenza meccanismi sempre più flessibili e pratici da usare. Il punto di contatto è dato dal fatto che il certificato X509 rilasciato dalla CA INFN può essere usato come strumento di autenticazione in

INFN-AAI. Cioè un utente con certificato X509 della CA INFN risulta avere accesso diretto alla INFN AAI e quindi ai servizi messi a disposizione centralmente.

Un secondo punto di contatto potrebbe concretizzarsi con la possibilità di ottenere un certificato X.509 utilizzabile per l'accesso ai sistemi GRID, a partire dalle credenziali Kerberos. Il progetto INFN-AAI infatti prevede la partecipazione alla federazione GARR-IDEM attraverso la quale ogni utente identificato attraverso le proprie credenziali Kerberos, potrà ottenere un certificato X.509 di tipo e-Science, rilasciato on-line dal servizio TCS di TERENA e fornito attraverso l'NREN partner, che in Italia è il GARR. Questo secondo punto di contatto permetterebbe in effetti di unificare il processo di accreditamento e riconoscimento degli utenti, evitando il passaggio attraverso una Registration Authority.

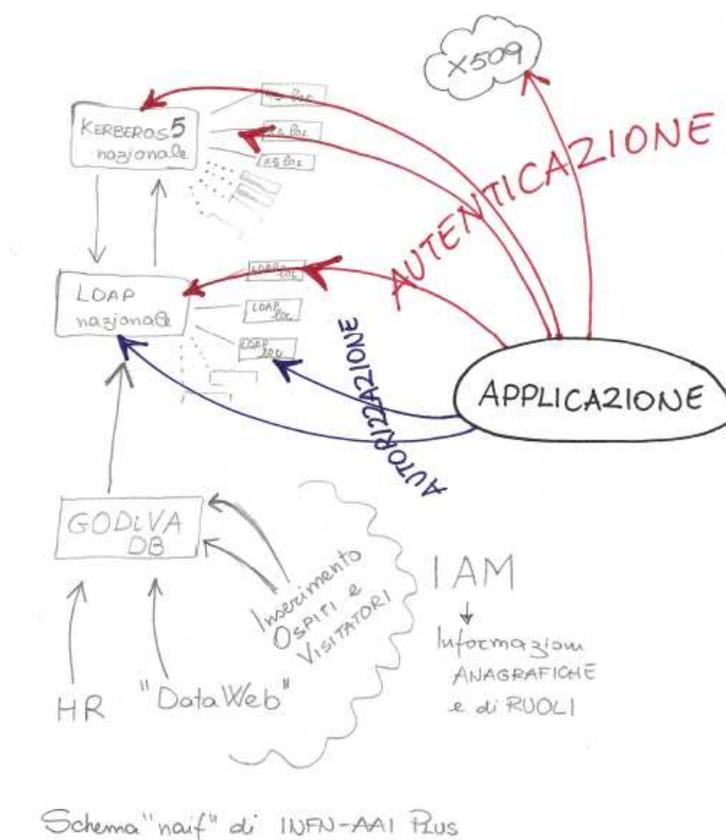


Figura 2: Schema del meccanismo di autenticazione e autorizzazione di una sessione di login usando INFN-AAI.

### 2.2.3 *Perchè non si può fare a meno di INFN-AAI*

Vediamo ora l'approccio che possiamo seguire per realizzare un accesso ad una risorsa di Calcolo Scientifico, sia essa intesa come una coda GRID o una coda locale o una risorsa ad esempio di tipo storage.

Si può predisporre una apposita User Interface sulla quale effettuare il login. Il login a cui si deve pensare, in questa sede, non è necessariamente un SSH con username e password, ma per semplicità e per intendersi correttamente sulla questione possiamo riferirci a questo esempio, anche perchè su scala locale di solito si lavora proprio se si è "loggati" su una macchina. Sulla UI appositamente predisposta si può fare login solo se si ha titolo. Questo si può tradurre in "possedere un attributo" o un "ruolo" in un meccanismo di AA sul quale si appoggia la UI in questione.

È a questo punto, ovviamente, banale il caso in cui si decida di considerare questa UI come una risorsa centrale dell'INFN (analoga quindi ai servizi centralizzati già disponibili sul portale centrale). In questo caso basterà fare in modo che il PAM della macchina sia configurato opportunamente, cosa possibile perchè il PAM è in grado di appoggiarsi ad un server LDAP per le autenticazioni. Questo livello è però molto leggero anche perchè data la particolarità dell'applicazione, una volta che si è fatto login occorre comunque trovare un modo per identificare chi ha davvero titolo ad effettuare specifiche operazioni sulla risorsa.

Ad un livello base potremmo allora prevedere la generazione di opportuni file `/etc/passwd` e `/etc/group`, derivandoli dalle informazioni contenute in LDAP. Entrando un po' più in dettaglio, al momento a Pisa l'autenticazione viene effettuata tramite Kerberos versione 4 e gli username ed i gruppi vengono letti da una tabella che conserva le informazioni di tutti coloro che hanno diritto ad accedere al sistema. A partire da questa tabella vengono generati opportuni file `/etc/passwd` e `/etc/group`. Il sistema di autenticazione della macchina riconosce l'utente perchè è scritto in `/etc/passwd` e `/etc/group` ed a quel punto interroga Kerberos 4 per sapere se l'utente è presente e se la password inserita dall'utente è corretta.

Se Kerberos5 permettesse qualcosa di analogo basterebbe generare opportuni file `/etc/passwd` e `/etc/group` a partire dal sistema LDAP nazionale facendo in modo che i file in questione contengano solo coloro che possiedono un preciso attributo. In questa ipotesi basterebbe generare uno script diverso da quello che si usa al momento per generare ogni ora i file `/etc/passwd` e `/etc/group`.

Ma gli strumenti di cui disponiamo sono più sofisticati ed è quindi possibile fare dei passi avanti e rendere questa operazione molto meno naif. Come accennavamo precedentemente questa semplificazione ha soprattutto lo scopo di indirizzare verso l'idea

generale. Infatti, tanto per cominciare, possiamo immaginare un portale WEB (vedi ad esempio il progetto di Calzolari e Licari, citato al capitolo 4).

Sta di fatto però che dovremo decidere come fare per far sì che l'informazione necessaria all'identificazione e all'attribuzione dei ruoli venga scritta una sola volta in modo da ridurre al minimo la possibilità di errori. Dato infatti che ci sono due sistemi paralleli di AA dovremo fare in modo che si decida un master e che poi venga scritto, come derivazione, l'altro DataBase. Il tutto nell'attesa che si possano unificare quanto più possibile le informazioni e che quindi si possa ridurre quanto più possibile il numero di informazioni da scrivere più volte.

Nel caso specifico il diritto ad usare una risorsa di calcolo GRID viene dato dal VO manager. Senza questa autorizzazione non è possibile sottomettere alla coda GRID. Potrebbe essere ragionevole, quindi, che per questa attribuzione si debba passare dal VO manager. Ma se il VO manager, potesse anche “contemporaneamente” scrivere la stessa informazione oltre che su VOMS sul GODiVA-DB e quindi su LDAP, allora entrambe le infrastrutture disporrebbero dell'informazione. E la UI userebbe LDAP per costruire l'accesso locale necessario per accedere alle risorse locali, mentre userebbe VOMS come meccanismo standard per la sottomissione di job GRID.

### **3 Infrastruttura di storage**

Nello sviluppo dell'infrastruttura di storage di INFN-Pisa, dedicata principalmente alle attività di Calcolo Scientifico, abbiamo individuato due ambiti ben distinti:

1. sistemi di storage e filesystem;
2. sistemi di accesso ed SRM;

rendendoci conto che spesso si tende a confondere questi due mondi e quindi delegare compiti specifici dell'uno all'altro. Per sviluppare una infrastruttura di storage in grado di soddisfare le varie esigenze presenti nel nostro Grid Data Center siamo partiti quindi dall'identificare chiaramente i compiti di questi due “mondi”, in modo da:

- scegliere per ciascuno le strutture hardware e software più indicate;
- individuare le procedure operative idonee a mantenere in funzione le strutture;
- chiarire e regolare le responsabilità dei vari attori coinvolti (Servizio Calcolo, esperimenti, fornitori);

Storicamente la struttura di storage di INFN-Pisa si è basata su una architettura di Storage Area Network (SAN) che in questi anni ha dimostrato indubbi vantaggi rispetto ad altre architetture (DAS, NAS ecc..). Per questo motivo è stato naturale sviluppare la nuova infrastruttura basandosi sull'esperienza passata, cosa questa che porta in modo assolutamente naturale la divisione netta fra "sistema di storage" inteso come insieme di spindle e applicazioni che accedono al sistema stesso. Spingendo oltre questo paradigma si è di fatto creato un ulteriore strato di astrazione fra l'applicativo e l'hardware sfruttando il filesystem.

Nel compiere questa operazione e quindi per generare questo nuovo strato di astrazione è stato necessario esaminare le necessità di accesso che il sistema deve soddisfare in modo da compiere le scelte opportune. I sistemi che accedono all'infrastruttura di storage possono essere:

1. farm di esperimento di tipo tradizionale. In questo caso l'accesso al disco è di tipo POSIX attraverso un filesystem di rete, tipicamente NFS;
2. struttura GRID principalmente con accesso mediato attraverso sistemi SRM. Dato che ogni sistema SRM attualmente disponibile alla fine utilizza un filesystem per mantenere le proprie informazioni circa i dati e spesso anche i metadati, si può concludere che per questi sistemi sia necessario un accesso di tipo POSIX;
3. cluster per calcolo parallelo. In questo caso l'accesso è nuovamente di tipo POSIX via rete ma con necessità di alte prestazioni non sempre garantibili attraverso NFS.

Alla luce di quanto sopra esposto si è deciso di utilizzare un filesystem di tipo cluster ad alte prestazioni e quindi la scelta naturale è stata GPFS. Dopo un paragrafo introduttivo su GPFS andremo ad analizzare un po' più in dettaglio i vari punti.

### **3.1 Introduzione a GPFS**

GPFS (General Parallel File System) di IBM è un file system che nasce riferendosi ad attività di tipo HPC e più precisamente per permettere ad applicazioni parallele di accedere simultaneamente allo stesso file system da ciascun nodo del cluster su cui il file system stesso è montato. Data la sua natura intrinsecamente parallela permette di ottenere una banda aggregata molto superiore rispetto ad un file system distribuito su più server di tipo tradizionale.

I componenti chiave dell'architettura sono:

- i nodi che compongono il cluster, che possono essere di varia architettura hardware (x86 o Power) e con diverso sistema operativo (AIX o Linux) anche mescolati all'interno del solito cluster;

- Network Shared Disk (NSD) che sono i dischi fisici su cui insiste il file system ed il cui accesso è garantito da alcuni nodi specifici del cluster. Una caratteristica fondamentale risiede nel fatto che un disco fisico può essere “visto” da più nodi permettendo quindi un elevato grado di resistenza del file system al fallimento del singolo NSD server;
- rete di interconnessione fra i nodi per comunicazioni sia di servizio che per il traffico dei dati. Per i due tipi di traffico si possono anche utilizzare due reti diverse, GPFS è infatti in grado di funzionare sia con Ethernet che con InfiniBand.

I punti di forza di GPFS sono:

- l’accesso al file system da più cluster, ciascuno dei quali gestito separatamente. In questo caso le connessioni sono autenticate e autorizzate tramite OpenSSL;
- le elevate prestazioni ottenibili grazie all’accesso parallelo al file system in maniera posix standard. Inoltre, essendo i file suddivisi su tutti i dischi si ottiene un naturale bilanciamento delle operazioni di I/O su disco oltre allo sfruttamento dell’intera banda passante disponibile sia verso il disco che verso la rete;
- l’elevato grado di “data availability” ottenuto sfruttando la capacità del file system di replicare sia i dati che i metadati. In congiunzione con questa caratteristica è molto importante anche la possibilità di dividere le risorse hardware in segmenti indipendenti completamente separabili in caso di guasto. Si riesce così ad ottenere una configurazione in grado di sopravvivere anche alla perdita di interi gruppi di dischi;
- l’alto grado di gestione e flessibilità. Infatti quasi tutte le normali operazioni di gestione del file system come l’aggiunta o la rimozione di dischi, il bilanciamento del loro utilizzo, l’aggiunta o la rimozione di nodi, possono essere eseguite senza interruzione nell’I/O.

In generale una infrastruttura GPFS permette di semplificare enormemente la gestione dello storage mantenendo semplice al contempo la vita degli utenti che possono trarre beneficio dalle prestazioni del file system senza dover ricompilare i loro codici. Anche dal punto di vista della gestione, essendo il file system posix, si possono usare tutti i tool UNIX standard senza bisogno di modificarli.

### 3.2 Farm di esperimento

Nella precedente infrastruttura SAN ogni farm aveva il proprio disk-server che accedeva direttamente alla “proprie LUN” sulla SAN. Questo approccio comportava tutta una serie di problemi:

- ogni server di disco aveva le proprie peculiarità e quindi un elevato carico di lavoro per il Servizio Calcolo a cui tipicamente ci si rivolgeva per la risoluzione dei problemi;
- problemi di accesso alle LUN di altri gruppi, problema questo risolvibile solo attraverso il masking delle LUN stesse. Questo approccio però introduceva ulteriore carico di lavoro per il servizio oltre a rendere più rigida l’infrastruttura SAN;
- lunghi tempi di attesa fra l’ordine di nuovo spazio disco e la sua effettiva disponibilità per gli utenti. Dato che ogni gruppo aveva le proprie LUN era necessario non solo procurare il disco ma anche installarlo fisicamente nel sistema, creare le LUN, modificare il masking e la configurazione dei server stessi prima di poter usare il nuovo spazio disco. Inevitabilmente potevano passare diversi mesi fra l’ordine di nuovo spazio disco e la sua effettiva disponibilità;
- data la rigidità del sistema era praticamente impossibile assegnare “on-spot” spazio disco ad una comunità che ne avesse bisogno magari per poco tempo.

La nuova infrastruttura ha permesso di superare questi limiti grazie all’astrazione fra lo strato hardware e quello che l’utente vede ed utilizza.

### 3.3 Struttura GRID

La gestione dello storage nel paradigma GRID è demandato ai servizi di Storage Management System (SRM) che da un lato forniscono una interfaccia comune di accesso ai dati attraverso un opportuno set di comandi e dall’altro permettono di astrarre dalla reale configurazione hardware su cui essi si basano. Inoltre questi sistemi garantiscono la persistenza delle informazioni relative ai dati presenti in un sito e la condivisione in maniera efficiente degli stessi fra siti geograficamente distribuiti. È quindi chiaro che un sistema di tipo SRM sia indispensabile per il corretto funzionamento di un sito GRID. D’altro canto però l’utilizzo di tali interfacce di accesso ai dati, per quanto semplificate possano essere, può risultare eccessivamente oneroso specialmente da parte quegli utenti che non hanno necessità di condivisione dei loro dati con altri siti ma che utilizzano GRID, in particolare il sito INFN-Pisa, come sostituto delle farm di gruppo. La nuova infrastruttura di storage è in grado di soddisfare le richieste di entrambi i tipi di comunità GRID.

In particolare osservando il funzionamento dei sistemi SRM si noterà che nella gestione dello spazio disco tutti si appoggiano su di un filesystem standard su cui archiviano le proprie informazioni di dati e metadati. Sfruttando questa caratteristica si può concludere che anche per i sistemi SRM una architettura in cui l'accesso ai sistemi di storage è mediata da un filesystem di tipo cluster come GPFS risponde alle necessità. Inoltre se tale infrastruttura è affiancata da un sistema come StoRM si riescono a concentrare all'interno dell'SRM sia le necessità di condivisione su scala geografica dei dati sia l'accesso diretto alla POSIX.

Quindi la nostra nuova infrastruttura garantisce l'accesso al filesystem per sistemi SRM generici, oltre ad essere in grado di permettere un accesso uniforme ai dati degli utenti: sia quelli gestiti attraverso SRM che gli altri. Per questo motivo si è ritenuto utile affiancare al dCache utilizzato da CMS un servizio SRM basato su StoRM.

### **3.4 Cluster ad alte prestazioni**

Al crescere delle richieste di configurazioni di tipo cluster e delle dimensioni dei cluster stessi non è più possibile garantire il necessario livello di prestazioni attraverso accesso NFS al disco. Il superamento di questi limiti passa da un lato attraverso lo sfruttamento di filesystem di rete con maggiori prestazioni e scalabilità come GPFS e dall'altro dall'utilizzo di reti trasmissive con più alte prestazioni. Dato che ogni cluster è comunque dotato, per definizione stessa di cluster, di una rete ad alte prestazioni (alta velocità e bassa latenza) appare naturale pensare di utilizzare tale rete per l'accesso al disco. In quest'ottica nuovamente GPFS sembra essere la scelta più naturale dato che nasce con il supporto per InfiniBand e che tale protocollo di rete è molto diffuso nell'ambito del clustering. Ciò non toglie comunque la possibilità di utilizzare altri tipi di rete HPC (come ad esempio Myrinet o 10Gbs ethernet) su cui sia possibile implementare lo stack TCP ove queste siano disponibili sul cluster in questione. Nel caso di Pisa la rete InfiniBand è quella di gran lunga più utilizzata e su cui si basa anche il cluster nazionale di gruppo4, per cui è stato naturale nella nuova architettura prevedere anche la connettività di tipo IB.

### **3.5 Descrizione della nuova infrastruttura di storage**

La nuova infrastruttura di storage si basa sui seguenti componenti hardware:

- 1 sistema di storage DDN S2A9900 con 140TB RAW installati;
- 2 switch Fibre Channel QLogic SANBox 5802V 8Gbs;
- 2 NSD server con FC a 4Gbs e rete di tipo 1GE;

- 4 NSD server con FC a 8Gbs e rete di tipo 10GE e IB;
- GPFS versione 3.2.1

Ciascun NSD server accede a tutte le LUN del sistema DDN con un doppio cammino attraverso i due switch FC (vedi figura 3): i cammini possono, secondo le necessità e la versione di driver, essere utilizzati in configurazione active-active oppure active-standby.

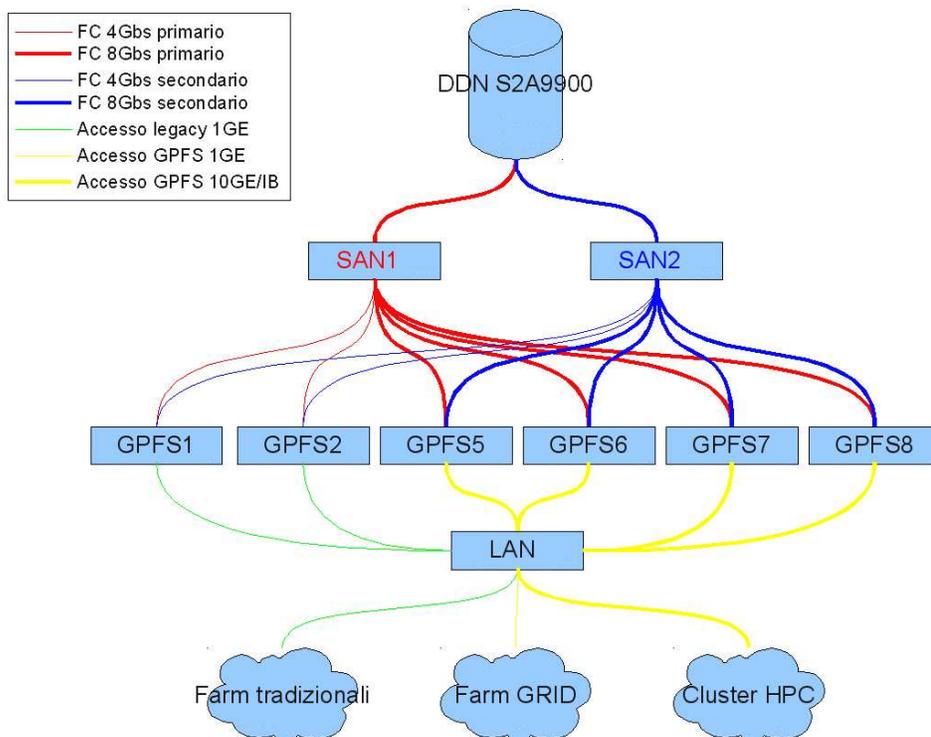


Figura 3: Schema della infrastruttura SAN di INFN-Pisa.

Tutto lo spazio disco RAW disponibile sul DDN è utilizzato per creare un unico filesystem GPFS sul quale ospitare le diverse categorie di utenti/accessi. Dato che lo spazio disco è indiviso si utilizza il meccanismo della quota per fileset, tipico di GPFS, per garantire ai vari gruppi l'allocazione delle proprie risorse. In questo modo, sfruttando la struttura ad albero del filesystem per rispecchiare le diverse tipologie di accesso e utilizzando le quote per fileset di GPFS, è possibile allocare in modo molto dinamico e flessibile le risorse di un gruppo non solo nella loro totalità ma anche secondo le varie tipologie di accesso.

I due server con FC a 4Gbs garantiscono l'accesso legacy al sistema esportando vari pezzi dell'albero GPFS via NFS alle farm di esperimento e comunque a tutti quei sistemi

che non sono in grado di effettuare accesso GFPS nativo. I rimanenti 4 server dotati di interfacce FC a 8Gbs garantiscono l'accesso GFPS nativo all'infrastruttura attraverso rete 10GbE e in prospettiva attraverso InfiniBand per le necessità di cluster ad alte prestazioni.

A scopo di esempio riportiamo l'organizzazione dello spazio disco di CMS in questa infrastruttura. CMS è un buon esempio perché al suo interno racchiude quasi tutte le necessità di accesso e quindi la sua quota totale di 73TB è divisa fra le varie tipologie di accesso:

1. aree per uso interattivo degli utenti;
2. aree SRM servite via dCache;
3. aree SRM servite via StoRM e quindi accedibili anche direttamente POSIX dalla farm grid;

questi tre tipi di disco corrispondono a path differenti a cui sono associate quote diverse come di seguito descritto:

- /gpfs/gpfsddn/cms → area per interattivo quota = 2.2TB
- /gpfs/gpfsddn/cmsdc → area SRM-dCcache quota = 70.5TB
- /gpfs/gpfsddn/srm → parte del filesystem gestito da StoRM
- /gpfs/gpfsddn/srm/cms → area SRM-StoRM quota = 0.3TB

### **3.6 Considerazioni finali sullo Storage**

La nuova infrastruttura di storage ci ha permesso di superare i limiti e le rigidità strutturali di una normale architettura SAN con disk-server dedicati ai singoli gruppi di utenti e/o applicazioni, rendendo di fatto molto più flessibile l'allocazione dello spazio disco anche per necessità estemporanee e limitate nel tempo. Inoltre l'utilizzo di una struttura omogenea fra le varie risorse computazionali di INFN-Pisa ha fornito un ulteriore grado di libertà nell'uso dello storage da parte degli utenti. In particolare risulta essere utilissima la possibilità di accedere, attraverso normali job grid, alle stesse aree disco utilizzate per il proprio lavoro interattivo. Inoltre l'utilizzo di StoRM come SRM permette di ottimizzare i processi di lavoro garantendo l'accesso SRM per i trasferimenti fra siti diversi ma salvaguardando le prestazioni tipiche dell'accesso POSIX locale per job.

Infine l'utilizzo di questa infrastruttura altamente integrata è stato un punto essenziale nel processo di integrazione delle farm locali all'interno della più generale infrastruttura di Calcolo Scientifico rappresentata dal Grid Data Center.

## 4 Conclusioni

Gli argomenti trattati sono stati studiati ed implementati a Pisa separatamente tra loro ed in momenti diversi nel corso dell'ultimo anno e mezzo. Sono tra loro strettamente collegati concorrendo tutti a perseguire un unico obiettivo:

- l'allargamento dell'orizzonte di utilizzo del paradigma GRID (non vogliamo nominare il termine CLOUD) ed il suo collegamento al mondo degli utenti e delle applicazioni batch non-GRID, fino all'interattivo.

In questa ottica si possono leggere i 3 capitoli che compongono il lavoro come le 3 parti di un unico disegno, tra loro complementari. Nello specifico:

- il primo capitolo (*chroot*) riguarda la questione della messa a disposizione dei sistemi target per le applicazioni in oggetto (anche se la tecnica è usata anche per i server di servizio);
- il secondo affronta la questione dell'integrazione tra una infrastruttura GRID ed una realizzata per applicazioni batch più generali, suddividendo il problema in due parti:
  - l'integrazione a livello di farm che insistono sullo stesso hardware, mediante opportuna gestione delle code tramite LSF
  - modalità di accesso da parte di una comunità di utenti remotamente distribuita (utilizzo di AAI per il Calcolo Scientifico)
- il terzo capitolo è dedicato ad una parte specifica dell'architettura che puntiamo a realizzare: lo storage. Anche in questo caso con l'obiettivo di collegare, e ove possibile integrare, le aree di storage utilizzate in GRID con quelle fuori GRID.

Le motivazioni per questa complessa ricerca possono, a loro volta, essere viste in modo "evolutivo":

- all'inizio, circa un anno e mezzo fa, si partì dalla constatazione che l'efficienza dell'utilizzo delle farm di esperimento (o di gruppo) era molto più bassa di quanto non fosse la produzione in ambito GRID <sup>(6)</sup>. Questo, essendo la parte preponderante dell'utilizzo delle farm comunque batch, portò all'idea di "farm on demand", che poi si è evoluta nell'integrazione delle farm con l'ambiente GRID;

---

<sup>6</sup>CCR-33/2009/P: Calcolo Scientifico: prime metodologie quantitative per un ambiente di produzione - A. Ciampa, E. Mazzoni

- alla fine dell'anno scorso abbiamo elaborato una proposta per la realizzazione di un cluster per il calcolo parallelo (mediante MPI) per la CSN4 (<sup>7</sup>). Poichè doveva essere considerata l'eventualità che non fosse da subito possibile l'utilizzo di job MPI in GRID abbiamo pensato di organizzare, per un periodo iniziale, la fruizione della risorsa mediante coda locale, che però doveva essere accessibile ad una comunità di utenti geograficamente (a livello nazionale) distribuita. Da qui la ricerca per applicare l'infrastruttura AAI per il calcolo scientifico, unitamente allo sviluppo, con la Scuola Normale Superiore di Pisa, di un'interfaccia web-based che facilitasse l'accesso GRID alla risorsa per utenti non particolarmente esperti (<sup>8</sup>);
- da ultimo, nel corso di quest'anno, abbiamo assistito all'inizio della discussione, anche in Italia, per la realizzazione di strutture T3 per gli esperimenti LHC. Ci è subito parso che una delle strade promettenti fosse mettere a frutto il lavoro fatto per proporre un approccio di "contiguità" tra tali strutture e quelle dei T2 che si potrebbe evolvere, in alcuni siti, in integrazione.

---

<sup>7</sup>CCR-38/2010/P: Progetto Cluster GRID CSN4: la proposta di Pisa - A. Ciampa, E. Vicari

<sup>8</sup>Daniele Licari, Federico Calzolari [2010]: "L-GRID: A Light portal to access Grid via Web", Computing in High Energy and Nuclear Physics CHEP 10 (Distributed Processing and Analysis) proceedings, Taipei [Taiwan]