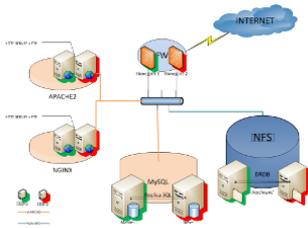




## Progetto Virtual Hosting in High-Availability Area Della Ricerca RM 1.<sup>†</sup>

Luca Ianniello,<sup>a</sup> Giuseppe Nantista,<sup>a</sup> Andrea Lora<sup>a</sup> and Augusto Pifferi<sup>a</sup>



Gestione dei servizi web 2.0 in High Availability: il numero dei servizi di rete da gestire aumenta di pari passo con le necessità dell'utenza dell'Area di Ricerca di Montelibretti. Tra la varietà dei servizi richiesti risulta evidente la necessità di requisiti minimi comuni per l'esecuzione del software di turno, ad esempio un database Mysql o un CGI (Common Graphic Interface), in grado di eseguire il codice dell'evenienza PHP, PERL, PYTHON, JAVA etc.. La soluzione adottata mira a garantire la massima disponibilità per i dati e continuità di servizio h24.

**Keywords:** Virtual Hosting, Web Services, Linux, Gentoo, Backup, High Availability, HTTP Server, Apache, Nginx, CGI.

### 1 Introduzione

La varietà e il numero, per diffusione e accessibilità, dei dati offerta dal Web fa sì che sia quasi impossibile che esista un unico software in grado di fornire servizi in modo esclusivo e inimitabile. Il quasi è dovuto esclusivamente da eventuali limitazioni hardware, imposte dal Brand di produzione ed applicate a speciali dispositivi accompagnate da software proprietario (**Adobe Flash**, **SAN**, **AXIS** Camera Station etc..), ottenendo di fatto l'esclusività di un prodotto.

Nel mondo dell'Open Source, invece, dove le limitazioni quasi scompaiono, è possibile trovare diverse soluzioni per le stesse necessità, si è quindi in grado di scegliere quale software utilizzare a seconda del servizio richiesto.

La pagina di un sito Web, allo stato pratico, è un'interfaccia grafica descritta e modellata da una serie di comandi predefiniti, che cambiano sintassi a seconda del linguaggio di programmazione utilizzato, rinchiusi generalmente in uno o più files che saran-

no processati dall'apposito CGI nel momento in cui l'utente (client) ne fa espressa richiesta.

Il **CGI** è l'interfaccia che genera la pagina Web attraverso i comandi contenuti nei programmi che risiedono all'interno del sito consultato.

Per rendere fruibili questi contenuti a tutti i client del World Wide Web è necessario un ulteriore software l'**HTTP Server** che svolge il compito di tenere aperta una porta di comunicazione in attesa di richieste di consultazione.

I servizi del Web 2.0, grazie al loro incessante progresso, diventano sempre più user-friendly e graficamente piacevoli a discapito della potenza di calcolo e della complessità della struttura di supporto, la caratteristica intrinseca del Web è la fruibilità h24 dei contenuti, sostenere questo status (on-line) nel tempo non è cosa banale soprattutto per via del continuo aggiornamento, necessario o richiesto, dai software utilizzati.

Per poter fornire un servizio in alta disponibilità occorre creare una struttura ridondata sia a livello hardware che software con procedure di fall-back pronte a superare eventuali blocchi o malfunzionamenti occasionali a tutti i livelli.

Nella struttura che abbiamo realizzato il requisito

<sup>a</sup> CNR - Istituto di Cristallografia, Strada Provinciale 35/d, Montelibretti, Italia

Creative Commons Attribution - Non commerciale - Condividi allo stesso modo 4.0 Internazionale

<sup>†</sup> Rapporto tecnico IC-RM 2013/11 protocollato in data 20/12/2013 n. IC 1881

di alta disponibilità è stato applicato anche ai dati ospitati, quindi anche tutti i dati sono ridondati fisicamente con procedure di sincronizzazione e backup automatizzati.

## 2 Software per i servizi web

L' HTTP Server è la porzione di software che si occupa di rendere reperibili dati e le informazioni presenti nelle pagine Web.

Agli inizi il contenuto di una pagina Web era composto di solo codice in linguaggio **HTML** (*HyperText Markup Language*); attraverso questo potente linguaggio è stato possibile creare contenuti Web rappresentati in maniera statica e graficamente molto più gradevole (testo e immagini) rispetto al classico terminale (solo testo). L'evoluzione del WWW ha reso possibile incorporare porzioni di codice di linguaggi differenti dall'HTML, come Javascript, Python, Php etc., all'interno delle pagine Web rendendo possibile la produzione di contenuti graficamente dinamici e strutturalmente molto più complessi (**multi-tier architecture**).

Con l'**HTML5** e le sue nuove API è possibile generare contenuti graficamente dinamici alla stregua di **Adobe Flash**, il software che per anni è stato eccellenza per le animazioni Web. L'HTML è uno standard e può essere interpretato universalmente: tutti i dispositivi muniti di un browser in grado di richiedere e rappresentare contenuti HTML possono visualizzare lo stesso dato.

Questa considerazione, seppur ovvia, sta spingendo i maggiori produttori di dispositivi e di software verso l'uso del Web per l'interazione tra l'utente finale e i servizi offerti in quanto oggi, con l'avvento e la diffusione degli smartphone e i tablet, il Web è il primo mezzo utile attraverso il quale è possibile controllare o consultare lo stato di apparecchiature di uso comune da remoto (interruttori telecomandati, telecamere di sorveglianza, condizionatori, router etc.). Ne consegue che una gran parte dei dispositivi elettronici oggi presenti sul mercato globale contengono un, seppur minimo, HTTP Server consultabile da un browser.

Ovviamente ogni dispositivo ha il suo sistema operativo e per ogni sistema c'è un HTTP Server differente con una differente capacità di lavoro, dovendo costruire una struttura in grado di ospitare n- siti web abbiamo installato su tutti e 4 gli HTTP Ser-

ver Gentoo-Linux su architettura x64 come sistema operativo di base.

Come riportato dalle pagine wiki, il software che per anni è stato la colonna portante tra gli HTTP Server è **Apache** e la versione da noi utilizzata al momento della stesura è la 2.2.25 (Unix).

Apache2 presenta un'architettura modulare, quindi ad ogni richiesta del client vengono svolte funzioni specifiche da ogni modulo di cui è composto, come unità indipendenti. Ciascun modulo si occupa di una funzionalità, ed il controllo è gestito dal core.

Questo tipo di approccio orientato ai thread o ai processi nella gestione delle richieste risulta abbastanza oneroso in termini computazionali ed è per questo che è stato completamente stravolto con **Nginx**, successore di Apache2 (n.d.r.), un altro HTTP Server che è stato sviluppato con la prerogativa di dover servire centinaia di milioni di richieste giornaliere utilizzando un approccio asincrono basato su eventi nella gestione delle richieste in modo da ottenere prestazioni più prevedibili sotto stress.

**Nginx**, fondamentalmente, è single-threaded ma può utilizzare più processi e più core. C'è un evento principale ciclico che attende istruzioni dal sistema operativo, ad esempio la lettura di un file, a questo punto il file viene letto in un buffer ed elaborato. Il singolo thread può servire in maniera molto efficiente decine di migliaia di connessioni simultanee (mentre il modello thread-per-connection non riesce molto bene in questo a causa della enorme contenuto di dati contestuale che dovrebbe gestire con un grande consumo di memoria, poiché ogni thread necessita di un proprio stack).

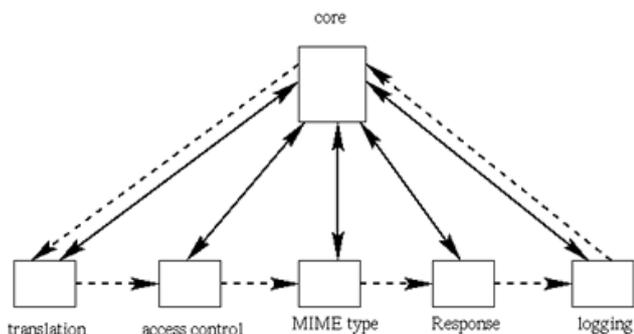
I siti più comuni hanno pagine generate da sistemi di gestione dei contenuti i **CMS** (*Content Management System*), questi sistemi oltre all'HTTP Server, per poter funzionare, hanno come requisito un database di dati e generalmente una CGI PHP da qui nasce l'acronimo **LAMP** che sta a indicare la struttura composta da Linux server con Apache come HTTP Server, Mysql per il database dei dati e l'interprete dei comandi Php.

## 3 Descrizione di High Availability

All'interno dell'Area di Ricerca Roma 1 esistono numerosi siti Web costruiti su piattaforma LAMP e proprio per favorirne e migliorarne l'utilizzo è stata creata una infrastruttura LAMP in High Availability

	HTML5 standard	HTML web browser features	Adobe Flash features
<b>Date started</b>	Work began in 2003 Working Draft as of 2011	N/A	Work began in 1996 Version 1 released in 1997
<b>Desktop operating systems</b>	N/A	Microsoft Windows, Apple Mac OS X, Linux Windows Phone 8+; Android 2.3+; Apple iOS 6+; Symbian Belle+; BlackBerry OS 7+	Microsoft Windows, Apple Mac OS X, Linux (PPAPI-only after 11.2), Solaris Up to Android 4.0 (unofficially for Android 4.1), Windows RT
<b>Mobile operating systems</b>	N/A	Xbox 360, [10][11] Sony PlayStation 3, Nintendo Wii	Sony PlayStation 3 (Flash 9 only), Nintendo Wii (Flash Lite only)
<b>Video game consoles</b>	PlayStation Vita	Limited access to web camera, microphone, accelerometer, GPS	Full access to web camera, microphone, accelerometer, GPS
<b>Devices support</b>	N/A		
<b>Market penetration</b>	N/A	~96% of internet connected PCs are CSS2/3 ACD compliant, ~99% support JavaScript 1.5 or above, ~84% of devices support basic SVG	~95% of internet connected PCs, (~83% have Flash Player 11, ~17% have 10 and below)
<b>Vector graphics formats</b>	N/A	Scalable Vector Graphics (SVG) supported on ~84% of browsers	SWF with embedded graphics
<b>Bitmap effects</b>	Yes	Varying support of Filters in CSS3 (eg. Glow, Blur, Drop Shadow, Sepia)	Yes, applied to text or graphics (eg. Glow, Drop Shadow, Bevel)[23]
<b>Vector text display</b>	Yes	Yes	Yes, with Saffron Type System
<b>Font support</b>	N/A	Installed fonts and custom fonts using CSS3 web fonts	Installed fonts and embedded fonts
<b>Text anti-aliasing</b>	N/A	Yes, implemented in most browsers, for system and custom fonts	Yes, in most cases
<b>Text tab stops</b>	No	Only supported inside "pre" tags	Yes, with Text Layout Framework
<b>Liquid text layout</b>	Yes	Yes, using the "div" tag and CSS styling	No, but text fields can be resized in ActionScript
<b>Tabular data</b>	Yes	Yes, using the "table" tag	No, but text fields can be arranged into a grid
<b>Linked text frames</b>	No	No	Yes, with Text Layout Framework
<b>Programming languages</b>	Depends	JavaScript	ActionScript, Pixel Bender
<b>C++ support</b>	Cross-compiling of C++ code to JavaScript using Emscripten	Native code execution with Google Native Client in Google Chrome only	Cross-compiling of C++ code to run in Flash Player using FlasCC
<b>Source code format</b>	Plaintext	Plaintext JavaScript with limited obfuscation	Compiled bytecode, can be obfuscated
<b>Data formats</b>	Depends	CSS3, HTML, XML, JSON	JSON, XML, Subset of CSS 1
<b>Data compression</b>	No	GZIP compression for HTML, JS and CSS files (on supported servers)	LZMA or DEFLATE for SWF files
<b>Image formats</b>	Depends	PNG, JPEG, Animated GIF	PNG, JPEG, JPEG-XR, Single-frame GIF
<b>Video formats</b>	Depends	Varying support of H.264, WebM and Ogg Theora (see HTML5 video)	H.264, Sorenson Spark, and On2 VP6
<b>Streaming video</b>	No	Pseudo-streaming only of WebM and Ogg Theora using video tag[33][33]	Flash Video, H.264 and partial support for MP4
<b>Audio formats</b>	Depends	Varying support of MP3, Ogg Vorbis, WAV PCM, AAC and WebM Vorbis (see HTML5 audio)	MP3, WAV and AAC audio files or embedded sound
<b>Fullscreen support</b>	No	Supported on some browsers	Yes, with warning displayed
<b>Encryption/DRM</b>	With obfuscation	No, all files being plaintext, except for obfuscation	Yes, being binary formatted files, unless decompiled
<b>File system access</b>	No	Varying support of single file upload, and drag and drop of files onto browser	Support for single file upload and generation, AIR only; full create/read/write access to file system
<b>Bitmap manipulation</b>	No	Varying support for HTML5 "canvas" element	Yes, using the BitmapData class
<b>Binary manipulation</b>	No	No, unless JavaScript Arrays are used for the same	Yes, using the ByteArray class
<b>Large binary data</b>	No	Using Web Sockets to stream binary or XML data	Yes, embedded or streaming binary data
<b>Offline storage</b>	Depends	Using Web storage or cookies to store binary or XML data	Using Local Shared Objects to store AMF-formatted data
<b>Metadata</b>	Meta tags	Can be included in meta tags	Extensible Metadata Platform

Comparazione HTML5 vs Flash. estratta da [wikipedia](#)



**Fig. 1** Schema funzionamento Apache2 thread-per-connection

a disposizione di tutti gli istituti afferenti all'Area. L' High Availability è la capacità, di una risorsa o di un servizio fornito da un sistema, di essere sempre disponibile.

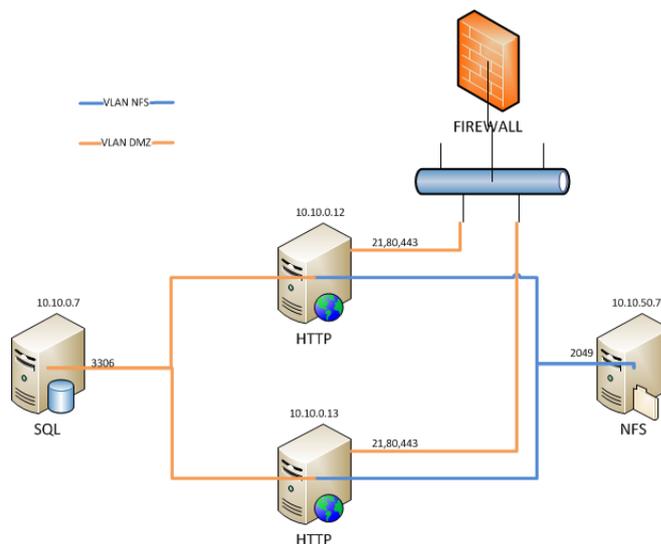
La diponibilità di un servizio è generalmente espressa in percentuale in base al tempo di diponibilità effettiva (uptime) nell'arco di un anno.

La tabella riportata di seguito ([http://en.wikipedia.org/wiki/High\\_availability](http://en.wikipedia.org/wiki/High_availability)) riporta il calcolo delle percentuali di disponibilità in base ai tempi di uptime e i diversi periodi di downtime:

Per ottenere l'High Availability si usa ridondare i sistemi (in ingegneria raddoppio delle componenti):

**Tabella 1** Estratta da [wikipedia](#)

Availability %	Downtime per year	Downtime per month*	Downtime per week
90%("one nine")	36.5 days	72 hours	16.8 hours
95%	18.25 days	36 hours	8.4 hours
97%	10.96 days	21.6 hours	5.04 hours
98%	7.30 days	14.4 hours	3.36 hours
99%("two nines")	3.65 days	7.20 hours	1.68 hours
99.5%	1.83 days	3.60 hours	50.4 minutes
99.8%	17.52 hours	86.23 minutes	20.16 minutes
99.9%("three nines")	8.76 hours	43.8 minutes	10.1 minutes
99.95%	4.38 hours	21.56 minutes	5.04 minutes
99.99%("four nines")	52.56 minutes	4.32 minutes	1.01 minutes
99.999%("five nines")	5.26 minutes	25.9 seconds	6.05 seconds
99.9999%("six nines")	31.5 seconds	2.59 seconds	0.605 seconds
99.99999%("seven nines")	3.15seconds	0.259 seconds	0.0605 seconds



**Fig. 2** Schema della struttura LAMP

la ridondanza può essere attiva o passiva.

La ridondanza attiva viene utilizzata soprattutto per sopperire ad elevati livelli di carico di lavoro, bilanciando i task tra i componenti attivi mentre nella ridondanza passiva una componente rimane allineata e pronta ad intervenire in caso di failure della componente attiva garantendo continuità al servizio.

La scelta di utilizzare l'una o l'altra soluzione va presa in base al carico di lavoro da supportare ed in base all'hardware a disposizione oltre che dal tipo di servizio richiesto.

Nella struttura realizzata sono stati applicati entrambi i criteri per la ridondanza (Attiva/Passiva) in base alle necessità incontrate.

#### 4 Infrastruttura delle piattaforme

Per ridondare un'infrastruttura di tipo LAMP è possibile utilizzare diversi approcci, quello riportato di seguito è focalizzato verso l'isolamento dei singoli servizi necessari alla struttura.

Sono stati preparati, dunque, un host per ogni singolo servizio

- 1 host HTTP Server + CGI PHP
- 1 host per il database MySQL

Essendo i due host separati e distinti è stato possibile applicare ridondanze specifiche per ogni servizio.

Per permettere a due o più HTTP Server differenti di indirizzare la stessa porzione di files, abbiamo posizionato i files dei siti ospitati, su condivisioni NFS. Questa soluzione permette di scalare orizzontalmente la struttura nel caso vi fosse necessità di un incremento delle performance semplicemen-

te aggiungendo un ulteriore Host per il medesimo servizio.

I problemi incontrati con questa soluzione non sono risolvibili in maniera immediata: la prima difficoltà, che porta con se una struttura così duplicata, è stata quella di impostare gli stessi UID e GID ai demoni che interagiscono con i files dei siti (Apache, Ftp, SAN) in modo da non creare eventuali errori di scrittura o negazione di permessi. Il secondo problema da risolvere con questa struttura è legata al demone FTP. Ritenendo che utilizzare un Host per il solo servizio FTP fosse eccessivo, abbiamo aggiunto il demone **Proftpd** sulle macchine con a bordo l'HTTP Server e, per non dover duplicare le credenziali d'accesso su entrambi i Server, abbiamo optato per l'utilizzo di un database comune come registro per le utenze FTP.

Un altro problema, conseguenziale al raddoppio degli HTTP Server, è la gestione delle connessioni in ingresso: anche qui si possono applicare diverse soluzioni che possono andare dal semplice bilanciamento di carico per mezzo del ROUND ROBIN tra multipli A Record nel DNS ad un eventuale load-balancer in grado di ripartire il traffico in ingresso tra gli Host a disposizione o in caso di configurazione Active/Passive, di ridirigere il traffico verso l'Host primario con l'incarico di passarlo tutto all'Host secondario nel momento in cui il primario risulta off-line.

In Figura 2 è riportato lo schema della struttura LAMP con i servizi isolati e raddoppiati per la parte HTTP Server.

Le connessioni in arrivo verso i server Web sono state bilanciate a monte dal Firewall di frontiera in modalità Active/Active.

## 5 Load Balancing e Fail-over su Stonegate

Il Firewall di frontiera verso la rete Internet di Montelibretti è di tipo software (Stonegate), questa applicazione mette a disposizione degli amministratori un Monitoring Agent proprietario, installabile su qualsiasi server, in grado di intervenire sulle decisioni del Firewall per l'instradamento dei pacchetti attraverso le informazioni raccolte dall'Agent in base a parametri prestabiliti. Anche il Firewall è ridondato in modalità Active/Active: 2 Host in bilanciamento di carico per la parte di Front-End più un ulteriore Host per la parte di Management come richiesto e previsto dalle best practice Stonegate.

```
# StoneGate Monitoring Agent global configuration file
# Package generated default configuration
begin host genweb4
config boot-delay 120
end

alert-interval 240
load-index 1500 load-average-1
config load-index-action exclude 10

test http_answering
interval 30
action exclude
command httpsaver 4 3500 /www.mlib.cnr.it/ /ARRM1v3/menu/menu.doc 80

test "port listening test"
interval 2:00
action exclude
host genweb4
recovery always
command portlistening 80 10.10.0.13

filesystem /dev/sda3
1024000
```

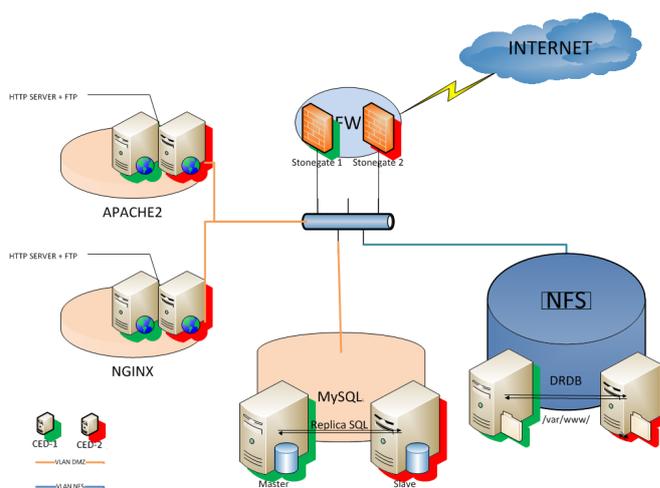
**Fig. 3** sgagent.conf file di configurazione dell'Agent Stonegate

Attraverso l'Agent è possibile gestire anche il meccanismo di Fail-over tra due o più Host: nella sezione "test http\_answering" della configurazione dell'Agent si imposta la verifica della presenza di un determinato file ogni 30 sec. Nel caso in cui questo controllo ritorni esito negativo l'Host viene automaticamente escluso dalla ricezione di nuove richieste HTTP che saranno quindi girate all'Host successivo. Oltre al meccanismo di Fail-over, gestito nella sezione "test" della configurazione dell'Agent, è stato abilitato un tetto massimo di carico per il bilanciamento delle connessioni tra i nodi Web.

Avendo monitorato a lungo i livelli di stress a cui sono stati sottoposti i Web Server preesistenti, si è desunto che, sia per quantità di carico di lavoro che per dimensione dei singoli database, con un unico Host è possibile soddisfare ogni esigenza di servizio per l'intera Area della Ricerca di Montelibretti, que-

sto anche in virtù del fatto che, l'Host in questione, deve farsi carico di un solo servizio rispetto alla soluzione "all-in-one" che prevede l'installazione dell'intera struttura LAMP su un unico Host. Questa conclusione ci ha portato a scegliere di utilizzare, per la gestione dei database, una ridondanza di tipo Active/Passive ampiamente supportata dal demone **MySQL**.

La porzione dei dati relativi alle pagine Web è montata all'interno dei web server come share NFS. Questo ci ha consentito di implementare una pila separata per l'alta affidabilità dello storage. Dal punto di vista dei server web l'unica direttiva necessaria è il mountpoint che indirizza all'IP del cluster NFS. Tale cluster è di tipo active/passive, offre la share NFS tramite un IP flottante che viene preso in carico dal server NFS attivo. Ognuno dei due server NFS monta una LUN ISCSI offerta da due diverse SAN, su queste LUN vengono scritti i dati riguardanti la share NFS. Con l'uso del software DRBD, che si frappone tra il layer disco e quello di filesystem, le due LUN vengono mantenute identiche in maniera sincrona a livello di blocco, garantendo la coerenza dei dati su due diversi backend disco. In caso di guasto del nodo attivo i meccanismi di clustering provvedono ad eseguire una procedura di STONITH sul nodo fallito per preservare l'integrità dei dati e a migrare il server NFS sul nodo in stand-by. Un terzo nodo partecipa al cluster per portare il quorum a 3, rendendo impossibili situazioni di split-brain.



**Fig. 4**

Avendo raddoppiato sia a livello fisico sia a livello applicativo l'intera struttura, ed avendo adeguati spazi a disposizione, è stato adottato il criterio previ-

sto dal **Disaster Recovery**, ossia abbiamo posizionato gli Host gemelli in sale CED differenti ottenendo una struttura rappresentabile come riportato nella figura di seguito.

Va però detto che per essere pienamente compliant con le direttive imposte a una struttura con Disaster Recovery le sale CED dovrebbero essere distanti almeno 20 Km per poter prevenire e superare i casi di calamità naturale di vasta entità o di bombardamento.

Confidando in un futuro privo di calamità possiamo dire che la struttura qui rappresentata è una struttura in alta affidabilità pronta per un Disaster Recovery.

## 6 Virtual Hosting

Il Virtual Hosting è il metodo utilizzato per ospitare più siti all'interno di un singolo Host (o gruppi di Host) in modo da sfruttare al massimo le risorse messe a disposizione dal Server ospitante (memoria RAM, CPU, IP).

Il Virtual Hosting può essere di tre tipologie: Name-based, IP-based o Port-based. Il Name-based Virtual Hosting è la tecnica che permette di associare, grazie al lavoro di traduzione svolto dall'HTTP Server, contenuti Web ad un nome di dominio: nella richiesta di contenuti effettuata dai client il nome di dominio viene anteposto come prima informazione da elaborare subito dopo la necessaria procedura di **handshake**, l'HTTP Server che riceve questa informazione si occupa di tradurla fornendo i corretti contenuti associati a quel determinato nome di dominio. Questa procedura, permette di risparmiare indirizzi IP in quanto è possibile associare N nomi di dominio ad un singolo IP. Va però tenuto presente che anche questo tipo di soluzione ha i suoi limiti: nel momento in cui si vogliono implementare certificati di sicurezza per il traffico criptato su più di un nome a dominio si crea un'ambiguità nel servizio per cui l'HTTP Server non sa quale certificato utilizzare per instaurare la connessione al momento dell'**handshake**. Esistono diversi work-around noti che permettono comunque di affrontare queste evenienze ma possono risultare inefficaci nel momento in cui il client non è in grado di recepire determinati accorgimenti utilizzati. Oltre al limite appena descritto c'è da aggiungere che, stante la peculiarità dei nomi di dominio, il sito non sarebbe più rag-

giungibile nel caso in cui la traduzione iniziale effettuata dal DNS non dovesse avvenire correttamente. Nell'implementazione IP-based ogni sito ha un suo indirizzo IP quindi sull'Host che farà da Server andranno configurati i relativi indirizzi. Questo tipo di implementazione ha il vantaggio di escludere il client dal processo di traduzione, su accennato, e quindi esente da problemi di compatibilità. L'altra faccia della medaglia dell'approccio IP-based è l'elevata complessità di amministrazione che si viene a creare oltre al consumo di un indirizzo IPv4 per ogni sito. Per questo motivo questa modalità è utilizzata solo in casi particolari. L'ultima tipologia la Port-based sfrutta la possibilità, fornita dai software di Web services, di fornire contenuti Web diversi a seconda della porta di comunicazione utilizzata. La porta di comunicazione comunemente usata per i contenuti Web è la 80, nel caso si utilizzi la metodologia Port-based si arriverà ad avere (utilizzando diverse porte es. 8080,81 etc.) siti raggiungibili solo da chi conosce la giusta porta. Questo tipo di implementazione viene tipicamente utilizzata nelle intranet per i servizi di amministrazione dove i contenuti Web devono essere consultati da un numero ristretto di utenti di provenienza nota. Pertanto la modalità da noi utilizzata, la Name-based, risulta essere la più conveniente in termini di risorse sia fisiche che amministrative. Sia APACHE2 che NGINX supportano il Virtual Hosting e considerando le peculiarità dei due software si è optato per l'utilizzo di APACHE2 per i siti degli Istituti dell'Area che hanno generalmente un carico di lavoro elevato ma distribuito nel tempo, mentre i Server con a bordo NGINX vengono utilizzati per le piattaforme di e-learning che generano un traffico massivo per brevi periodi.

In tabella 2 l'elenco dei siti presenti sui server APACHE2 dell'Area di Montelibretti.

All'interno di questo elenco è possibile trovare siti web statici e siti web dinamici tutti comunque prevedono PHP come interprete dei comandi.

La versione dell'interprete PHP installato sulla struttura APACHE2 è PHP 5.3.27. anche il PHP è in continuo aggiornamento tanto che al momento della stesura di questo documento la versione disponibile è la 5.5.7. Per ogni nuova versione supportata di PHP ce ne è una che diventa DEPRECATED (non supportata) rendendo l'aggiornamento inevitabile; è solo questione di tempo.

**Tabella 2** Elenco siti presenti sui server Apache 2.

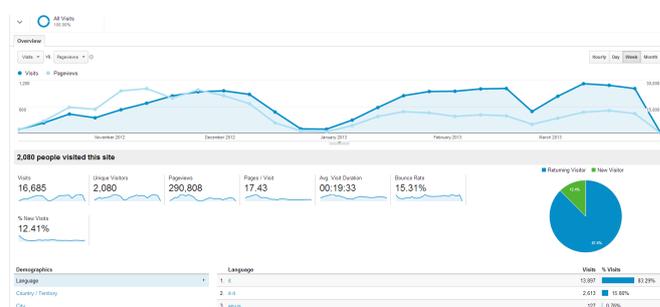
URL	Descrizione
http://biblio.mlib.cnr.it/	Biblioteca dell'Area della Ricerca di Roma 1
http://calliope.mlib.cnr.it/	Editore Digitale a cura dell'I.C.
http://durc.mlib.cnr.it/	Portale per censimento DURC – Intranet Mlib
http://dvr.mlib.cnr.it/	Portale per il censimento DVR – Intranet Mlib
http://dyna.imip.cnr.it/	I.M.I.P. - Progetto Dynamics at nano scale
http://genesi.mlib.cnr.it/	Progetto GENESI – Rete integrata per il trasferimento di tecnologie e metodologie per la generazione di sviluppo industriale in sistemi di produzione.
http://gire.mlib.cnr.it/	I.B.A.F. - Gruppo Italiano Resistenza Erbicidi
http://greenability2011.mlib.cnr.it	I.T.A.B.C - Forum Internazionale sulla sostenibilità e la conservazione delle città storiche
http://grs-lab.itabc.cnr.it/	I.T.A.B.C - Ground Remote Sensing LAB
http://helpdesk.mlib.cnr.it/	S.R.A. – Front End telematico per la gestione dell'HelpDesk per l'Area di Ricerca RM1
http://icm2003.mlib.cnr.it/	I.M.C - Istituzionale
http://icmontelibretti.gov.it/	Istituto Comprensivo di Montelibretti – Progetto Scuole
http://impreseaimo.mlib.cnr.it/	Portale Imprese CAIMO
http://link.ism.cnr.it/	I.S.M.N - Laboratories for Integrated Nanoscale Knowledge
http://magnet.mlib.cnr.it/	CNR/INFN, CNISM, INSTM – I Convegno nazionale di Magnetismo
http://nano2010.mlib.cnr.it/	I.S.M – Conferenza Nano 2010
http://nmr.imc.cnr.it/	I.M.C. - Laboratorio Risonanza Magnetica
http://portaleucbs.mlib.cnr.it/	Unione di Comuni della bassa Sabina a cura dell' I.C.
http://postesuap.mlib.cnr.it/	Poste Italiane - Sportello SUAP
http://progetto.mlib.cnr.it/	Progetto Anti Digital Divide a cura dell'I.C.
http://www.16issssrome2010.it	I.M.C. - 16th International Symposium on Separation Sciences
http://www.appia.itabc.cnr.it/	I.T.A.B.C - Archaeological Park of ancient via Appia
http://www.ibaf.cnr.it/	I.B.A.F - Istituzionale
http://www.ibba.mlib.cnr.it/	I.B.B.A – Istituzionale sez. Montelibretti
http://www.ic.cnr.it/	I.C - Istituzionale
http://www.icfpm.mlib.cnr.it/	I.C.F.M.P. – Istituzionale sez. Montelibretti
http://www.imc.cnr.it/	I.M.C. - Istituzionale
http://www.imip.cnr.it/	I.M.I.P. - Istituzionale
http://www.isc.mlib.cnr.it/	I.S.C. – Istituzionale sez. Montelibretti
http://www.iscima.cnr.it/	I.S.C.I.M.A. - Istituzionale
http://www.isma.cnr.it/	I.S.M.A. - Istituzionale
http://www.itabc.cnr.it/	I.T.A.B.C - Istituzionale
http://www.iwp_rixs2014.mlib.cnr.it/	International Workshop on Photoionization and Resonant Inelastic X-Ray Scattering
http://www.manfor.eu/	I.B.A.F. - Management of Forests, Carbon and Biodiversity
http://www.mlib.cnr.it/	Area della Ricerca di Montelibretti RM1
http://www.phototech.eu/	Photosynthetic proteins for technological applications
http://www.rstfen.iscima.cnr.it/	I.S.C.I.M.A. - Rivista di Studi Fenici
http://www.sra.mlib.cnr.it/	Servizio Reti Area – RM1
http://xenios.mlib.cnr.it/	Gestione telematica dei convegni scientifici a cura dell'I.C.

In tabella 3 l'elenco dei siti ospitati dalla struttura NGINX.

Come riportato da numerosi test e benchmark disponibili online NGINX risulta essere l'HTTP Server migliore, per velocità nell'offrire contenuti statici e

**Tabella 3** Elenco siti presenti sui server NGINX.

URL	Descrizione
http://minerva.mlib.cnr.it/	e-learning Formazione a Distanza
http://verdeorienta.mlib.cnr.it/	e-learning in collaborazione con ITCCG "E. Fermi"
http://fermi.mlib.cnr.it/	e-learning in collaborazione con: Istituto Tecnico Commerciale Geometri "Enrico Fermi" Istituto Tecnico Industriale "Alessandro Volta" Istituto d' Istruzione Superiore "Via Falisca snc" Istituto d'Istruzione Superiore "Via Pedemontana" Istituto d' Istruzione Superiore "G.Quarenghi"
http://romaforma.mlib.cnr.it/	e-learning in collaborazione con Roma Capitale
http://puentes.mlib.cnr.it/	e-learning in collaborazione con Universidad Mundial e Università degli Studi di Teramo
http://openopportunity.mlib.cnr.it/	Gestione Meeting Open Opportunity
http://catullo.mlib.cnr.it/	e-learning in collaborazione con il Liceo Classico e Linguistico "Catullo"
http://peano.mlib.cnr.it/	e-learning in collaborazione con Liceo Scientifico Statale "G. Peano" di Monterotondo



**Fig. 5** Confronto andamento tra le visite registrate e il numero di pagine richieste.

per maggior quantità di connessioni servite rispetto ad APACHE2. Il divario diminuisce se si confrontano i 2 software su contenuti dinamici generati da codice PHP in quanto il linguaggio utilizzato nei siti dinamici aggiunge la medesima componente a carico della CPU che dovrà farsi carico di interpretare il codice e generare il contenuto. A questo va aggiunto che la differenza di prestazioni, a parità di hardware tra i due HTTP Server, la si può apprezzare solo con grandi carichi di lavoro (dell'ordine di milioni di richieste al giorno, qualche centinaio al secondo). Pur essendo lontani da così alti carichi di lavoro l'utilizzo di NGINX sul fronte dell'e-learning si è rivelata, fino ad oggi, una scelta corretta. Di seguito è riportato un estratto (vedi figura 5) dei grafici del carico di lavoro sopportato dalla sola piattaforma Roma-Forma allestita per i corsi di formazione del personale di Roma Capitale. I grafici fanno riferimento a un periodo di 6 mesi: dal 7 Ottobre 2012 al 30 Marzo 2013.

## 7 Conclusioni

La struttura così come descritta si colloca, ad oggi, al livello “four nine” della tabella dell’ High Availability sopra riportata, il costante lavoro dell’IC mira ad attestare la propria struttura almeno al “five nine” ovvero a mantenere lo status “on-line” per un intero anno con al massimo 5 min di downtime.

Le soluzioni tecniche e il software utilizzato per la realizzazione della struttura in High Availability si sono dimostrate efficienti e rispondenti agli obiettivi del progetto. La qualità tecnica dell’infrastruttura e i tempi di risposta ottenuti nell’erogazione del servizio hanno consentito di soddisfare tutte le richieste giunte al servizio reti dell’Area di Ricerca di Roma 1. Si prevede che i futuri sviluppi dell’infrastruttura qui descritta sarà orientata ad all’implementazione della virtualizzazione degli Host. L’implementazione dell’utilizzo di macchine virtuali, attraverso i potenti tool disponibili nell’ambiente VSphere (come la clonazione di un intera Virtual Machine a caldo o la possibilità di effettuare una snapshot di un intera Virtual Machine che consente il ripristino della stessa, riportandola al momento dello snapshot, con tempi brevissimi), rafforza ulteriormente l’intera struttura.

### Riferimenti

- 1 [http://en.wikipedia.org/wiki/High\\_availability/](http://en.wikipedia.org/wiki/High_availability/).
- 2 <http://www.php.net/>.
- 3 <http://www.mysql.com/>.
- 4 <https://httpd.apache.org/>.
- 5 [http://wiki.dreamhost.com/Web\\_Server\\_Performance\\_Comparison/](http://wiki.dreamhost.com/Web_Server_Performance_Comparison/).
- 6 [http://www.wikivs.com/wiki/apache\\_vs\\_nginx/](http://www.wikivs.com/wiki/apache_vs_nginx/).

## 8 Glossario

**Browser:** un programma che consente di usufruire dei servizi di connettività in Internet, o di una rete di computer, e di navigare sul World Wide Web

**Client:** indica una componente che accede ai servizi o alle risorse di un’altra componente detta server  
**Disaster Recovery:** l’insieme delle misure tecnologiche e logistico/organizzative atte a ripristinare sistemi, dati e infrastrutture necessarie all’erogazione di servizi a fronte di gravi emergenze che ne intacchino la regolare attività.

**DRDB:** Distributed Replicated Block Device

**HTTP Server:** software di gestione del server Web.

**Multi-tier architecture:** è un architettura client-server nella quale le funzioni di presentazione, elaborazione delle applicazioni e gestione dati sono logicamente separate.

**MySQL:** è un sistema di gestione relazionale dei database, composto da un pacchetto client e da un pacchetto server.

**PHP:** linguaggio di programmazione interpretato, concepito per la programmazione Web e ora utilizzato principalmente per sviluppare applicazioni web dal lato del server.

**SAN:** Storage Area Network è una parte di una rete ad alta velocità di trasmissione costituita esclusivamente da dispositivi di memorizzazione di massa, Il suo scopo è quello di rendere tale storage disponibili per qualsiasi computer connesso ad essa.

**VSphere:** una suite potenziata di strumenti per la gestione del cloud computing utilizzato da VMware ESX/ESXi

**VMWare:** software che provvede alla virtualizzazione e al cloud computing.