# Realizzazione di un'infrastruttura Cloud pilota basata su OpenStack

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

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

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

## 1. Introduzione

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

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

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

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

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

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

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

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



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

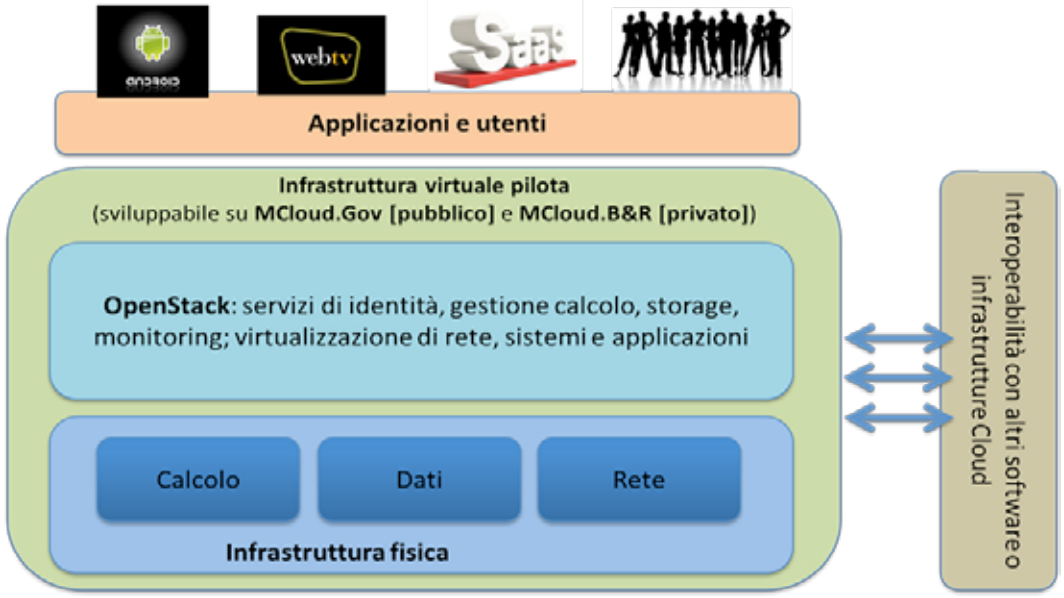


**Fig 1 -** Architettura del progetto MCloud

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

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

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

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

*Storage*: OpenStack supporta due tipi di Cloud storage:

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

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

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

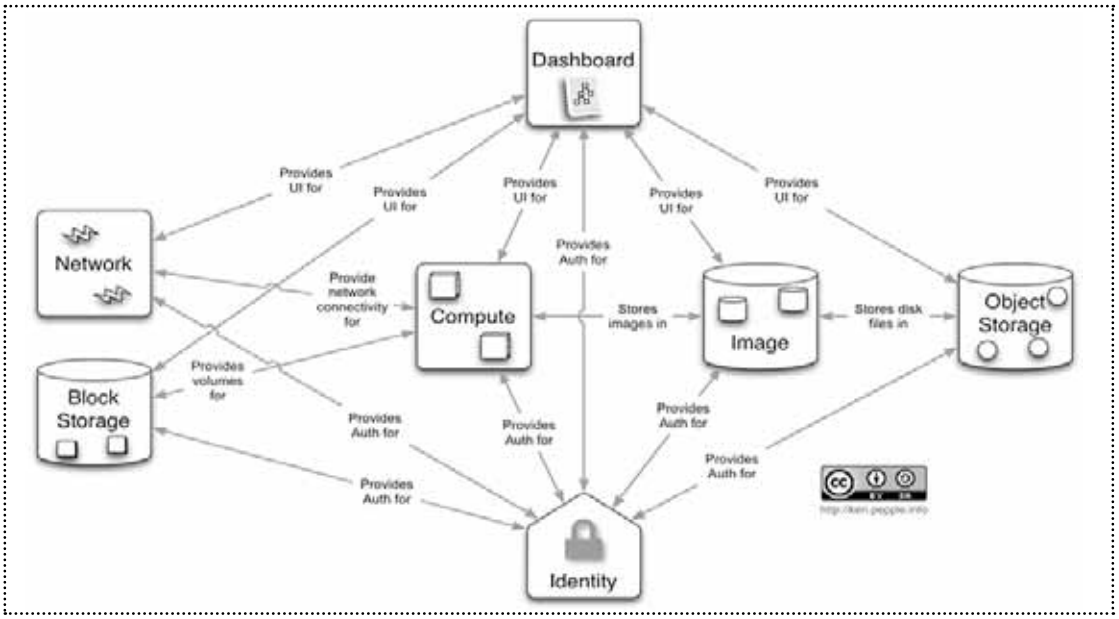


**Fig 2 -** Architettura di OpenStack "Folsom"

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

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

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

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

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

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

La figura 3 mostra come attraverso *GlusterFS*

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

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

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

Sull'infrastruttura pilota sono state abilitate le seguenti funzionalità:

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

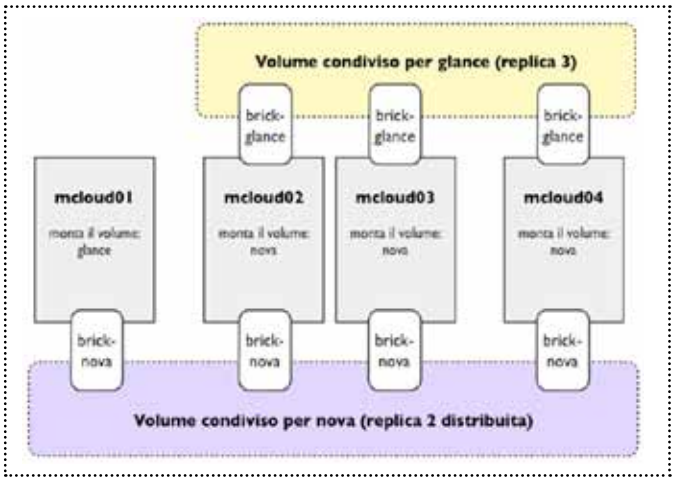


**Fig 3 -** Organizzazione spazio dati dell'infrastruttura pilota

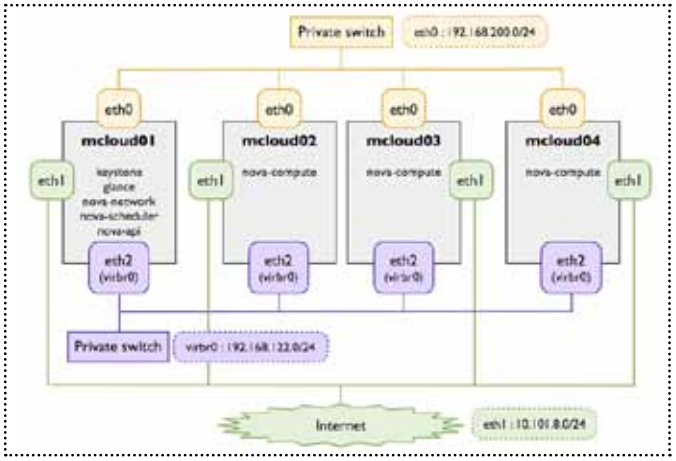


**Fig 4 -** La configurazione di rete dell'infrastruttura pilota

## 3. Valutazione delle esigenze e identificazione dei dati utilizzabili

**per monitoring e accounting**

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

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

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

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

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

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

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

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

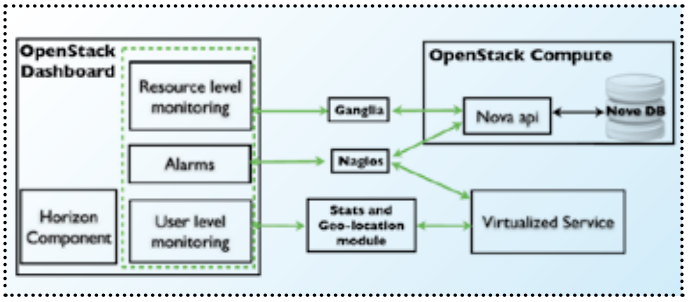


**Fig 5 -** Architettura delle componenti per il monitoring e accounting

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

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

nor diffusione in soluzioni basate su OpenStack.

## 4. Integrazione del monitoring e dell'accounting nella dashboard

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

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

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

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

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

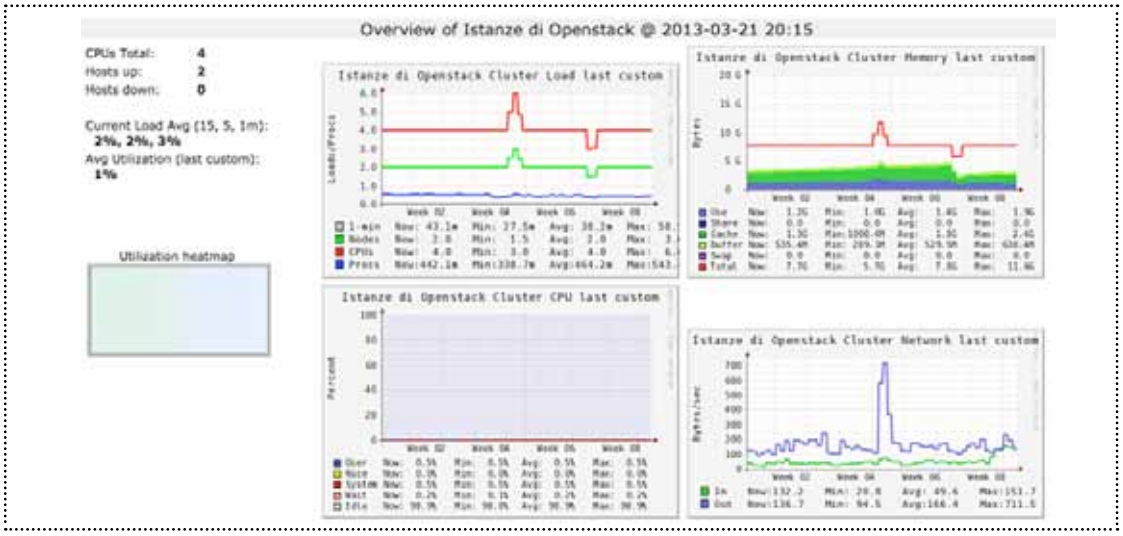


**Fig 6 -** Media dei consumi delle risorse virtuali.

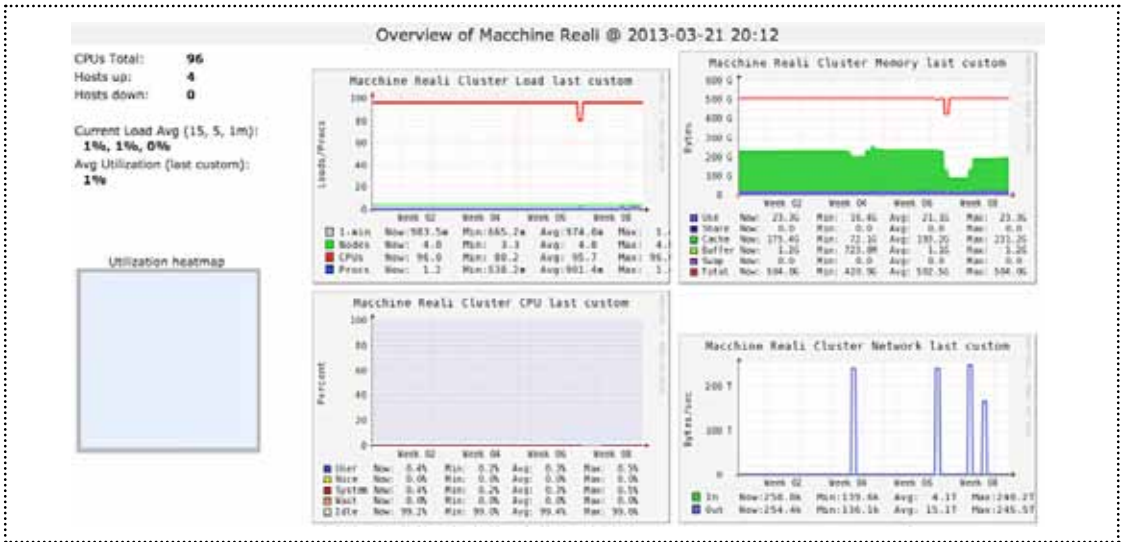


**Fig 7 -** Media dei consumi delle risorse fisiche

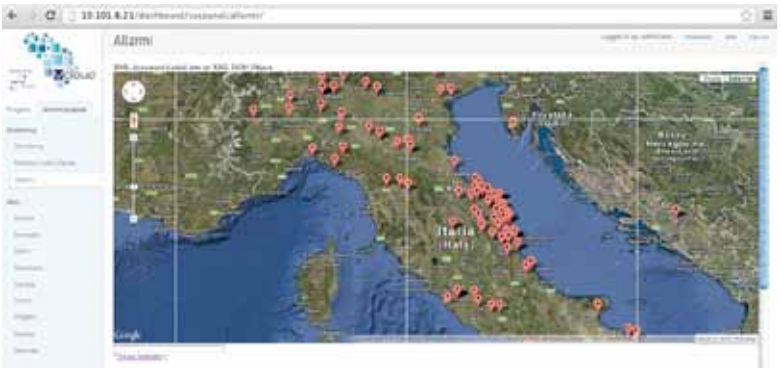


**Fig 8 -** Dettagli geografico degli accessi dell'utenza.

di log.

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

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

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

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

## 5. Conclusioni e sviluppi futuri

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

## Riferimenti Bibliografici

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

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

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

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

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

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

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

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