ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA

FACOLTA' di SCIENZE MATEMATICHE FISICHE E NATURALI

Corso di Laurea in Informatica

Cloud Computing su sistemi Linux

Tesi di Laurea in Architettura degli Elaboratori

Presentata da:
Carla Rasenti

Relatore:
Chiar.mo Prof.
Vittorio Ghini

Sessione III

Anno Accademico 2010/2011

ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA

FACOLTA' di SCIENZE MATEMATICHE FISICHE E NATURALI

Corso di Laurea in Informatica

Cloud Computing su sistemi Linux

Tesi di Laurea in Architettura degli Elaboratori

Presentata da:
Carla Rasenti

Relatore:
Chiar.mo Prof.
Vittorio Ghini

Sessione III

Anno Accademico 2010/2011

Parole chiave: Cloud Computing, Linux, portabilità, open source, gestione risorse

Indice

Introduzione

La parola Cloud Computing è emersa recentemente come una delle parole d'ordine nel settore ICT. È diventato un'ottima soluzione per fornire un sistema flessibile, on-demand e dinamicamente scalabile dell'infrastruttura di calcolo per molte applicazioni.

Numerosi fornitori IT promettono di offrire risorse di calcolo, storage e servizi di hosting e di fornire una copertura in vari continenti, offrendo prestazioni service-level agreement (SLA)[1] e promettendo l'uptime per i loro servizi.
In un'ottica tradizionale un client (il vostro computer) invia una richiesta ad un preciso server (quello su cui ha sede il vostro sito preferito) che si preoccupa di soddisfare la richiesta restituendo la pagina a cui siete interessati. Il computer che vi risponderà sarà sempre e solo il destinatario finale della richiesta che avete inviato e – questo è fondamentale – le risorse (memoria, potenza di calcolo, archivio dati) che verranno utilizzate saranno unicamente quelle del singolo computer che avete contattato.

Negli ultimi anni invece, con lo sviluppo delle reti e la sempre maggiore disponibilità di banda, è diventato possibile utilizzare da remoto apposite macchine messe a disposizione da servizi di hosting e housing, i quali si occupano del collocamento e della gestione fisica delle stesse. Oggi molte aziende utilizzano servizi di questo tipo, evitando così di dover sostenere i costi e occuparsi del collocamento, dell'alimentazione e della climatizzazione del data center. I sistemi di pagamento si basano su abbonamenti di durata mensile o annuale stipulati con i clienti, a seconda della tipologia di servizio offerta. Le soluzioni di hosting e housing risultano oggi le più utilizzate, in particolare dalle piccole e medie aziende; tuttavia esse presentano una limitazione: costringono gli utenti a dimensionare in maniera preventiva le proprie risorse IT, cercando di prevedere quanto grande sarà il carico di lavoro da svolgere e

1 *Accordo sul livello del servizio. C*ontratto tangibile tra due parti che, se da un lato assicura la fornitura dei servizi a livelli pre-negoziati, dall'altro comporta il pagamento di penalità in caso di mancato raggiungimento di tali livelli.

come si dovranno gestire eventuali picchi di richieste.
Per ovviare a questo problema, sono state sviluppate le nuove tecnologie di Cloud Computing. I servizi offerti si basano sul concetto di “utility computing”: le risorse IT vengono fornite in base alle specifiche richieste dei clienti, sfruttando la rete Internet per la distribuzione e applicando tariffe secondo un modello di "pay-as-you-go" in cui i clienti pagano in base al reale utilizzo delle risorse computazionali, di storage e di trasferimento dei dati. Ogni utente può ottenere la quantità di risorse di cui necessita, che vengono rese immediatamente disponibili e pronte per essere utilizzate, non appena vengono richieste. Si possono richiedere esclusivamente le risorse necessarie per eseguire le proprie operazioni, poiché in ogni momento è possibile aumentarne o diminuirne la quantità allocata, qualora la situazione lo richiedesse.

Queste "nuvole" sono quindi una naturale evoluzione dei tradizionali data center.
Le risorse offerte dai provider cloud sono molteplici e si possono dividere in tre categorie: quelle che offrono l'accesso alle infrastrutture, alle piattaforme o alle applicazioni che sono rispettivamente definiti come IaaS (Infrastructure as a Service), PaaS (Platform as a Service) e SaaS (Software as a Service). Questi servizi hanno aumentato l'interoperabilità, l'usabilità e ridotto il costo della computazione, application hosting, di storage e consegna di diversi ordini di grandezza, ma vi è una notevole complessità nel garantire che le applicazioni e i servizi siano in grado di scalare in base alle esigenze del cliente per realizzare, in modo coerente e affidabile, il funzionamento con dei picchi di carico.

Il Cloud Computing è quindi un'architettura di servizi IT innovativa che permette l’utilizzo di risorse hardware e software distribuite in remoto ed accessibili direttamente on-line; un servizio di outsourcing informatico innovativo che sta riscuotendo sempre maggior interesse da parte di singoli privati, aziende, enti pubblici ed organizzazioni di tutto il mondo.
Permette di concentrare in grandi data center le proprie risorse informatiche, di qualsiasi dimensione esse siano, per poi accedervi facilmente quando si vogliono

compiere tutte le operazioni di elaborazione dati necessarie. In questo modo questa nuova tecnologia permette di eliminare tutti i costi ed i problemi legati all'acquisto e alla gestione di software ed attrezzature hardware (licenze, manutenzione, energia per l'alimentazione ed il raffreddamento dei data server, ecc.).
In definitiva, con un'applicazione cloud, per operare al cliente basterebbe possedere un PC, notebook, smartphone o palmare dotato unicamente di sistema operativo e di un browser per la navigazione in Internet.
Consente una piena personalizzazione dei servizi permessa dalla maggior parte delle piattaforme di Cloud Computing: il fruitore finale può configurarli in base alle sue esigenze e fruirne in piena libertà e mobilità.

Oggi, sempre più aziende ed organizzazioni si trovano a dover affrontare gli elevati costi in termini di energia, stoccaggio e manutenzione delle loro risorse IT.
Se a questo si aggiunge, d'altro canto, un concomitante aumento della domanda di servizi informatici da parte del mercato, si capisce bene perché molte di queste realtà stanno decidendo di affidarsi sempre di più al Cloud Computing e di trasferire sempre maggiori risorse 'tra le nuvole'.

La tesi si articola in due parti, per un totale di 7 capitoli.
La prima parte analizza il concetto di Cloud Computing studiandone l'architettura, le tipologie e le caratteristiche principali. Nella seconda parte l'attenzione è focalizzata in particolare sulle opportunità del Cloud Computing in ambito IaaS/Linux.

Nel capitolo 1 viene spiegato il significato del termine, come si è arrivati al Cloud Computing e quali sono le sue caratteristiche principali, dalla sua utilità alla sua struttura. Nel capitolo 2 analizziamo più in dettaglio i livelli dell'architettura: il client, SaaS, PaaS, IaaS e la virtualizzazione, mentre nel capitolo 3 vengono presentati i tre tipi di cloud (pubblico, privato, ibrido). Il capitolo 4 è incentrato sulle tecnologie che rendono possibile il Cloud Computing, con un occhio di riguardo alla virtualizzazione e alle Virtual Machine Monitor Xen, KVM e VMWare. La prima parte termina con il

capitolo 5 dove vengono affrontate le caratteristiche e il tema della sicurezza nella cloud.

Nel capitolo 6 vengono descritti alcuni dei principali “attori” dei sistemi cloud di tipo IaaS, quali OpenNebula, Haizea e Amazon EC2. Infine il capitolo delle conclusioni, il numero 7, da cosa abbiamo visto a cosa ci aspettiamo nel futuro.

1 Cosa si intende per Cloud Computing

Il termine "cloud" è usato come metafora per Internet, sulla base del disegno della nube utilizzato in passato per rappresentare la rete telefonica, e successivamente per rappresentare Internet in diagrammi di rete di computer come un'astrazione dell'infrastruttura sottostante rappresenta.

Il National Institute of Standards and Technology (NIST) lo definisce così:
"Cloud Computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction."
["NIST.gov - Computer Security Division - Computer Security Resource Center". Csrc.nist.gov.]

Con il termine Cloud Computing intendiamo un insieme di tecnologie informatiche che permettono l'utilizzo di risorse (hardware e/o software) dinamicamente scalabili e spesso virtualizzate, distribuite in remoto (fornite come un servizio via internet).
Il Cloud Computing è quindi, in realtà, niente di più che l'evoluzione di una serie di tecnologie le quali sono in grado di rivoluzionare le modalità con cui le organizzazioni costruiscono le proprie infrastrutture informatiche, grazie alla possibilità di scalare in modo dinamico i servizi ad altri computer e storage in modo semplice e trasparente.

La cloud può essere sia software che infrastruttura hardware, ovvero può essere un'applicazione a cui accedere tramite il Web, oppure un server che attiviamo solo quando serve e solo per il tempo strettamente necessario.
Si può così utilizzare la tecnologia solo per il tempo che serve, non un minuto di più, senza dover installare nulla sulla macchina (sul desktop/pc) e senza dover pagare ciò che non si sta usando.

Possiamo vedere il Cloud Computing come una naturale evoluzione della diffusa adozione della virtualizzazione, dell'architettura service-oriented e dell'utility computing. Rende disponibili all'utilizzatore le risorse (reti, server, storage, applicazioni e servizi) come se fossero implementate da sistemi (server o periferiche personali) "standard". L'implementazione effettiva delle risorse non è definita in modo dettagliato; anzi l'idea è proprio che l'implementazione sia un insieme eterogeneo e distribuito - *the cloud* - di risorse le cui caratteristiche non sono note all'utilizzatore.

Il Cloud Computing consente l'accesso agli utenti di computer di massa alle risorse di storage senza bisogno di sapere dove tali risorse sono o come sono configurate; ovvero non è richiesta la conoscenza, da parte degli utenti finali, della localizzazione fisica e della configurazione del sistema che eroga i servizi né tanto meno competenze specifiche sull'infrastruttura tecnologica “in the cloud” che li fornisce.
Tale concetto può essere paragonato con la rete elettrica in cui gli utenti finali consumano le risorse (l'elettricità) senza aver bisogno di comprendere il funzionamento dei dispositivi che consentono la fornitura di tale servizio o di sapere dove risiede la fonte primaria.

Una delle idee più importanti dietro il Cloud Computing è la scalabilità e la tecnologia chiave che la rende possibile è la virtualizzazione. La virtualizzazione consente un migliore utilizzo di un server aggregando più sistemi operativi e applicazioni su un singolo computer condiviso. La virtualizzazione consente anche la migrazione on-line in modo che se un server si sovraccarica, è possibile migrare a un nuovo server meno affollato.

In sintesi possiamo dire che dal punto di vista esterno, il Cloud Computing è semplicemente la migrazione di calcolo e di storage al di fuori di un impresa e nella nuvola. L'utente definisce il fabbisogno di risorse e il provider di cloud assembla virtualmente questi componenti all'interno delle sue infrastrutture, come mostrato nella Figura 1.

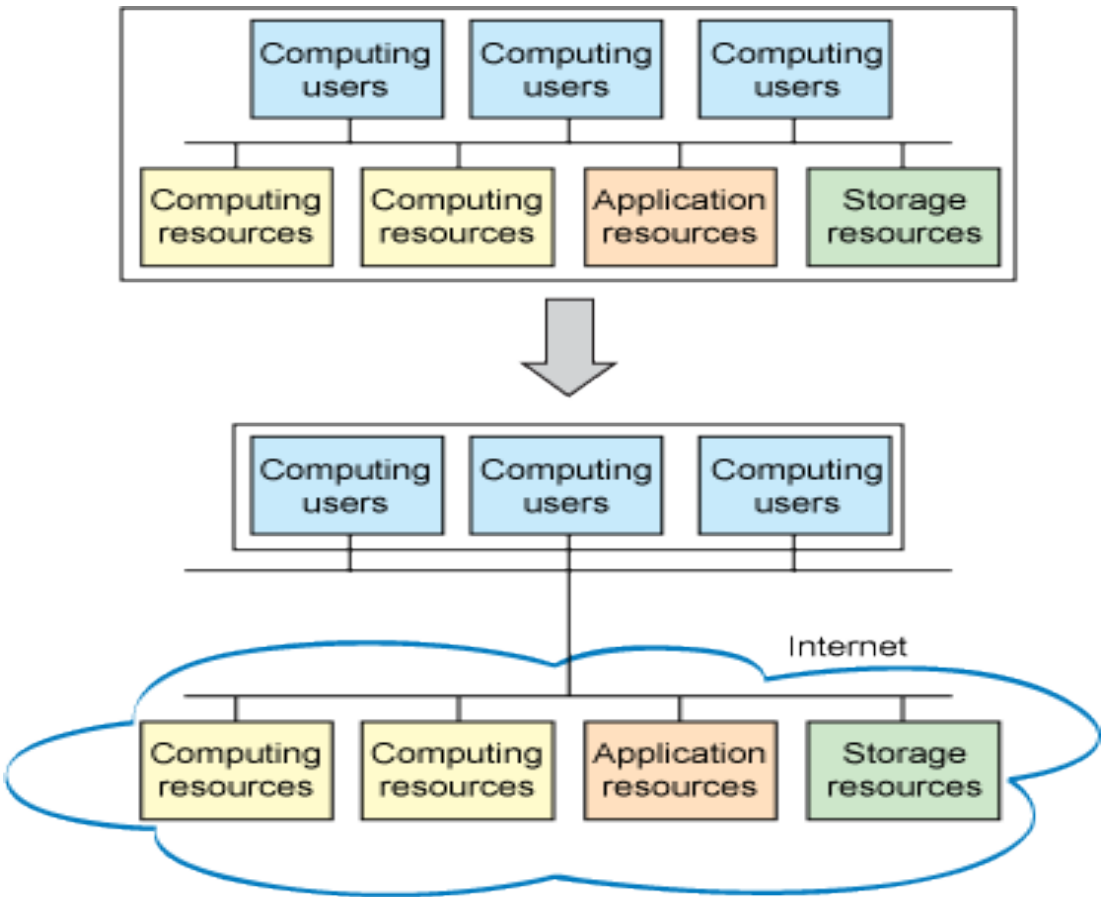

Figura 1 Mostra come le risorse migrano nella cloud dentro internet

1.1 Le 6 fasi per arrivare al Cloud Computing

Le sei fasi dei paradigmi di calcolo, dal mainframe, al pc, al network, a internet, al grid e infine al Cloud Computing, sono mostrate in Figura 1.1.

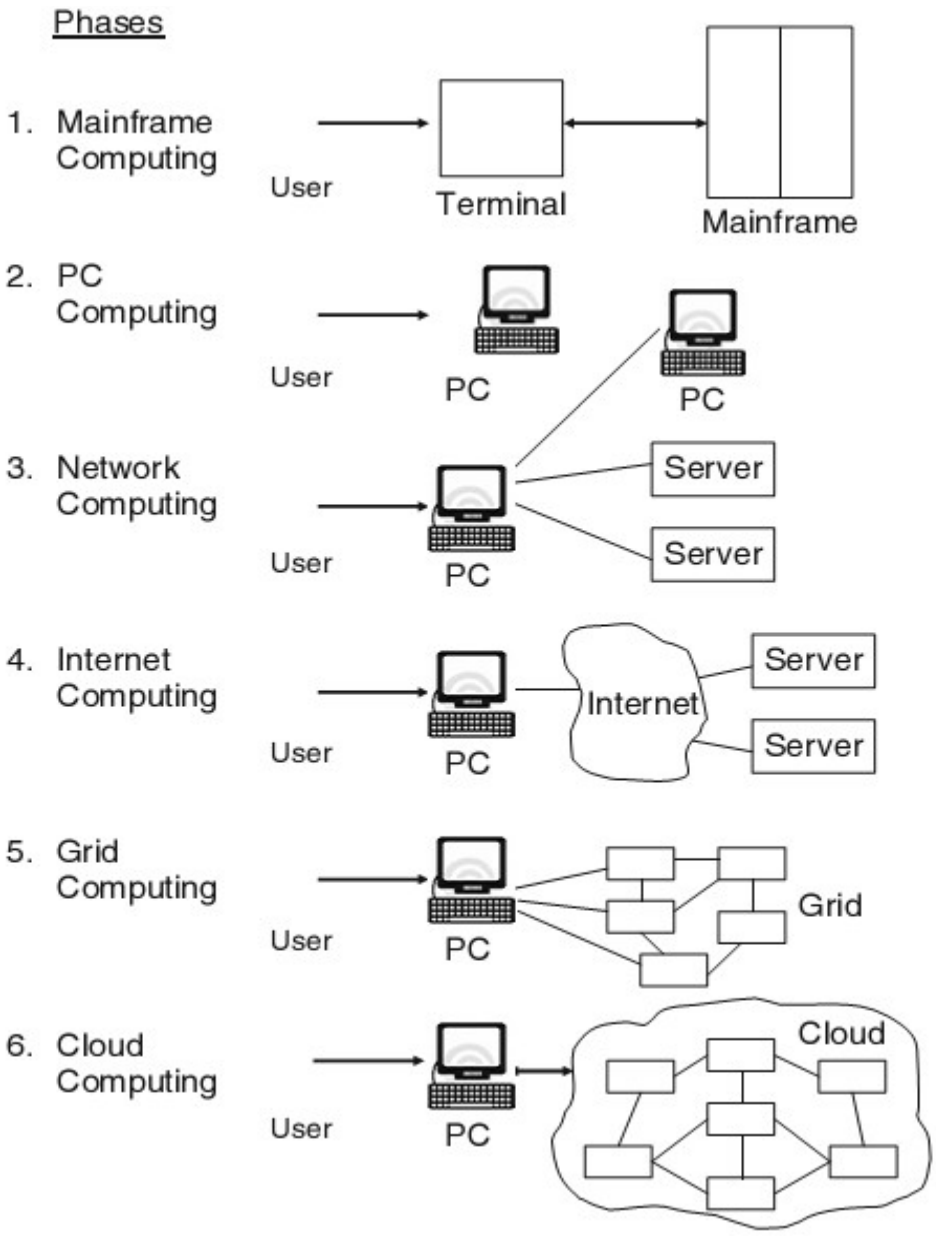

Figura 1.1 Mostra le 6 fasi per arrivare al Cloud Computing

Fase 1: molti utenti condividono un potente mainframe utilizzando un terminale remoto;

Fase 2: PC indipendenti diventano potenti abbastanza da soddisfare i maggiori bisogni degli utenti;

Fase 3: PC, portatili e server sono connessi insieme attraverso network locali per condividere risorse e incrementare le performance;

Fase 4: Local Network sono connessi con altri local network formando un network globale come internet per utilizzare applicazioni e risorse remote;

Fase 5: Grid computer condivide la potenza di calcolo e la memoria attraverso un sistema di calcolo distribuito;

Fase 6: Cloud Computing prevede risorse condivise su internet in modo semplice e scalabile.

Confrontando questi sei paradigmi di calcolo, sembrerebbe che il Cloud Computing sia un ritorno al paradigma originale, cioè il mainframe.
La differenza sostanziale tra loro è che, mentre il mainframe offre una finita potenza di calcolo, il Cloud Computing fornisce potenza e capacità quasi infinite. Inoltre nel mainframe i terminali remoti (dummy terminals) agiscono come dispositivi di interfaccia utente, mentre nel Cloud Computing la potenza dei pc è in grado di fornire una potenza locale di calcolo.

1.2 Utilità del Cloud Computing

Il Cloud Computing rappresenta un modo per sviluppare applicazioni in un ambiente virtuale, dove le capacità di calcolo, banda, storage, sicurezza e affidabilità non sono dei problemi (non essendo necessario installare il software sul proprio sistema). In un ambiente di elaborazione virtuale, è possibile sviluppare, distribuire e gestire applicazioni, pagando solo per il tempo e le capacità utilizzate.

Le situazioni tipiche in cui è possibile ritrovarsi sono sostanzialmente tre:

1) il server viene enormemente sovradimensionato e risulta economicamente dispendioso mantenere in vita il sistema con la conseguente perdita di denaro e di risorse macchina inutilizzate;

2) il server è correttamente dimensionato ma un improvviso aumento del traffico, magari per l'uscita di un nuovo prodotto della ditta proprietaria dello stesso, mette in crisi tutta l'infrastruttura fino a portarla al collasso e ad una situazione di non raggiungibilità delle pagine o dei servizi erogati;

3) il server è sottodimensionato e non regge nemmeno 10 minuti on-line vista la mole di traffico che deve sostenere.

Solitamente le situazioni 1) e 3) non si verificano facilmente, contrariamente invece la situazione 2) trova riscontro nella vita reale con una certa frequenza.

Le soluzioni possono essere principalmente due, un'ottima gestione del parco macchine da parte degli amministratori (difficile da realizzare) o il ricorso a tecnologie evolute come quelle del Cloud Computing.

I risvolti positivi sono tanti ma possono essere così riassunti:

- *scalabilità e costi*: non bisognerà pagare per spazio fisico o elettricità che non si utilizza. Poiché si utilizza una piccola parte di risorse hardware altrui, si ottimizza l'efficienza con cui si soddisfano le proprie esigenze d'infrastruttura. Inoltre, non si dovrà pagare per l'intero complesso di server pur senza usarne gran parte delle risorse, che consumano comunque elettricità.
- la possibilità di *virtualizzare e condividere* le risorse tra le diverse applicazioni per un miglior utilizzo del server.
- la possibilità di *aggiungere nuove risorse* hardware nel preciso istante in cui esse si rendono necessarie e non un secondo prima. Non si avrà alcun costo d'investimento iniziale e quindi nessuna spesa in conto capitale, ma solo d'esercizio. Infine, si può disporre delle risorse nella cloud nel giro di pochi minuti da quando ne sorge l'esigenza.
- non ci si dovrà preoccupare della gestione dell'hardware.
- quando non si ha più bisogno di una risorsa, oppure quando si vuole modificare configurazione, basterà disabilitarla, senza preoccuparsi di sbarazzarsi dell'hardware.

1.3 Paradigma del Cloud Computing

Il Cloud Computing fornisce una collezione di servizi i quali possono essere presentati come strati della sua dell'architettura come mostrato in Figura 1.2.

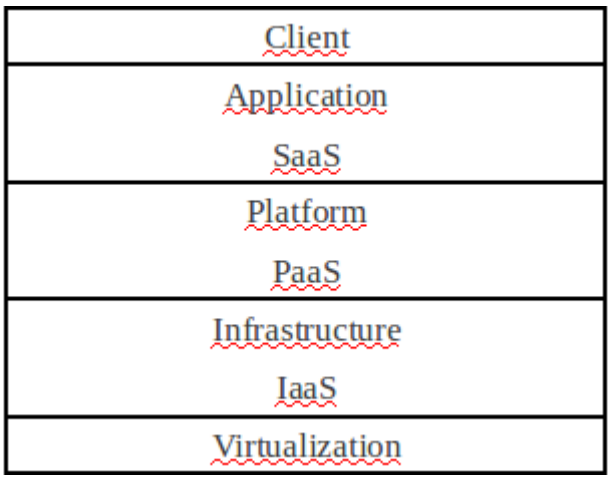

Figura 1.2 Mostra i livelli dell'architettura del Cloud Computing.

Analizziamo brevemente queste principali figure del paradigma del Cloud Computing:

- *Client*: può essere amministratore e/o finale. Nel primo caso sceglie e configura i servizi offerti dal fornitore, generalmente offrendo un valore aggiunto come ad esempio applicazioni software; nel secondo caso utilizza i servizi opportunamente configurati dal cliente amministratore. In determinati casi d'uso il cliente amministratore e il cliente finale possono coincidere. Ad esempio un cliente può utilizzare un servizio di storage per effettuare il backup dei propri dati, in questo caso il cliente provvede a configurare e utilizzare il servizio.

- *Application*: i servizi applicativi cloud, chiamati SaaS (Software-as-a-Service), permettono agli utenti di far girare del software in remoto sulle loro macchine mettendolo a disposizione come servizio su Internet; dal lato cliente, ciò

significa nessun investimento iniziale in server o licenze software (ma solo pagare per il reale tempo di utilizzo), sul lato provider, con una sola applicazione da mantenere, i costi sono bassi rispetto ai tradizionali hosting.

- *Platform*: la piattaforma di servizi PaaS (Platform-as-a-Service) offre ambienti di sviluppo come un servizio, includendo sistemi operativi e i servizi (programmi, librerie) necessari per una particolare applicazione. Ad esempio, un PaaS oltre a server virtualizzati e storage prevede un particolare sistema operativo e un set di applicazioni (in genere una macchina virtuale) insieme con l'accesso ai servizi necessari, ad esempio MySQL database o altro, risorse locali specializzate. In altre parole, PaaS è IaaS con un software personalizzato per date applicazioni. (Questi servizi sono limitati alla progettazione del venditore e così non si ha una completa libertà)

- *Infrastructure*: infrastruttura come servizio, IaaS (Infrastructure-as-a-Service), il più basso livello di servizio offerto. Offre appunto un'infrastruttura di computer virtuali con potenza di elaborazione garantita e larghezza di banda riservata per la memorizzazione e l'accesso a Internet. Piuttosto che l'acquisto di server, di software, di spazio per i data center o di apparecchiature di rete, i clienti acquistano tali risorse come un servizio completamente in outsourcing; utilizzano così risorse hardware in remoto su richiesta nel momento in cui ne hanno bisogno, senza che esse vengano assegnate a prescindere dal loro utilizzo effettivo. In sostanza, è la capacità di leasing o di un centro informatico di dati con specifici vincoli di qualità del servizio, che ha la capacità di eseguire un arbitrario sistema operativo e software.

- *Virtualization*: tramite la virtualizzazione è possibile suddividere una singola macchina fisica in varie macchine virtuali completamente indipendenti che, come tali, possono utilizzare sistemi operativi diversi e propri spazi di memoria, propri dischi e proprie CPU.

Un esempio di Paas è mostrato in Figura 1.3.

I Provide Paas IDE (Integrated Development Environment) includono data security, backup e recovery, application hosting e scalable architecture.

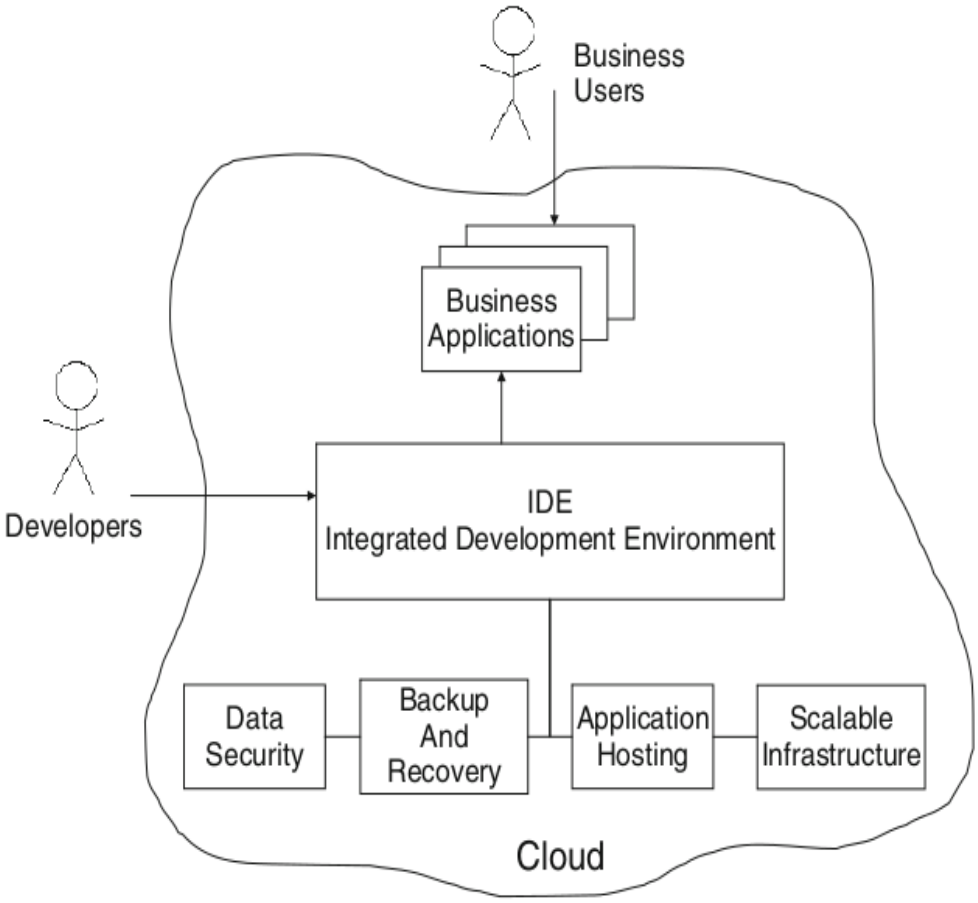

Figura 1.3 Mostra un esempio di provider PaaS IDE.

Esistono 3 categorie di cloud service, come illustrato in figura 1.4.

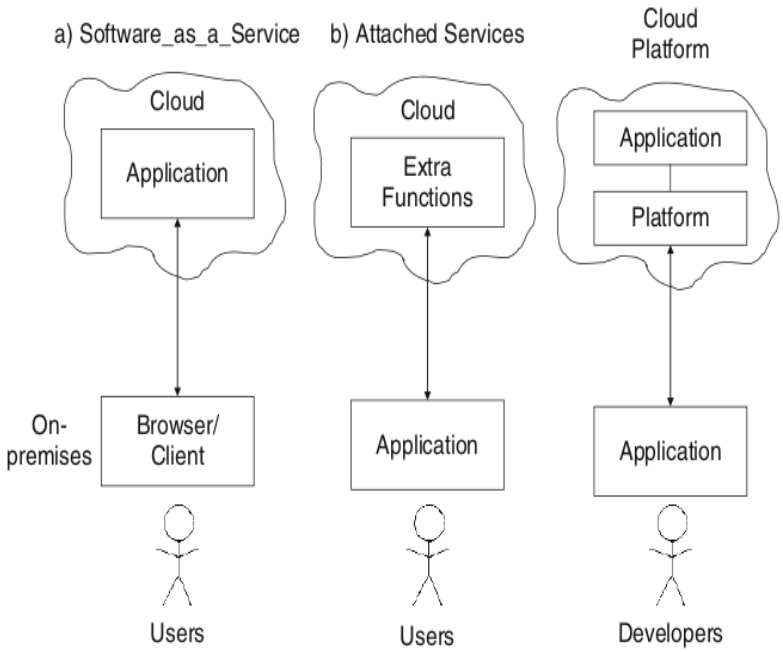

Figura 1.4 Mostra tre diversi tipi di cloud service: a) SaaS; b) l'applicazione gira sul Client; c) piattaforma cloud per creare applicazioni.

Nella Figura 1.4a è mostrato il cloud service SaaS, dove l'intera applicazione gira nella cloud. Il client contiene un semplice browser per accedere all'applicazione. Un esempio di SaaS è saleforce.com.

La Figura 1.4b mostra un altro tipo di cloud service, dove l'applicazione gira sul client; tuttavia si accede a utili funzioni e servizi forniti dalla cloud. Un esempio di questo tipo di cloud service sul desktop è iTunes di Apple. L'applicazione desktop riproduce la musica, mentre il cloud service è utilizzato per acquistare nuove canzoni e video.

La Figura 1.4c mostra una piattaforma cloud per creare applicazioni, la quale è usata dagli sviluppatori. Gli application developers creano una nuova applicazione SaaS usando la piattaforma cloud.

2 Linux nella nuvola

Esaminiamo più in dettaglio come Linux (e la comunità opensource) contribuisce al Cloud Computing.

2.1 SaaS

SaaS da, come servizio, la possibilità di accedere al software attraverso Internet o di eseguirlo dietro a un firewall su una rete locale o su un personal computer.
Dal punto di vista degli utenti è del tutto trasparente dove il software è ospitato, sotto che sistema operativo gira e in che linguaggio è scritto. Soprattutto non è necessario installare nulla in locale, visto che l'unica cosa che serve è un qualsiasi browser web (es: Gmail, Saleforce.com è un sistema di CMR, Enterprise Customer Relationship Management, che permette a chi si occupa di vendite di gestire clienti, potenziali ed effettivi, e tutto il ciclo di vendita, dal primo contatto con un prospect fino alla vendita e al post-vendita).

Un primo approccio a SaaS è stata la Application Service Provider (ASP). Le ASP forniscono abbonamenti a software che è reso disponibile su Internet. L'ASP fornisce il software e applica una tariffa in base al suo effettivo utilizzo. In questo modo, non si acquista il software, ma semplicemente lo si “affitta” in base alla necessità del momento.
Questo approccio alla distribuzione delle applicazioni è parte del modello di utility computing, dove tutte le tecnologie sono nella "nuvola" e sono accessibili su Internet come servizio.

Un'altra prospettiva sul SaaS è l'uso del software su Internet che viene eseguito in modalità remota. Questo software può essere sotto forma di servizi utilizzati da una applicazione locale (definita come Web Service) o di una applicazione remota attraverso un Web browser.

Un esempio di servizio di applicazioni remote è Google Apps, che fornisce diverse applicazioni aziendali attraverso un browser Web standard. Solitamente da remoto le applicazioni in esecuzione si basano su un application server per offrire i servizi necessari. Un application server è un framework software che offre le API per servizi software (come la gestione delle transazioni o l'accesso al database). Gli esempi includono Red Hat JBoss Application Server, Apache Geronimo, e IBM WebSphere Application Server, ecc.

Un altro esempio di SaaS è il browser Chrome di Google. Il browser è un ambiente ideale, come un nuovo desktop attraverso il quale le domande possono essere consegnate (o in locale o in remoto) in aggiunta alla tradizionale navigazione web.

2.1.1 Livelli SaaS

Il servizio SaaS può essere pensato su 4 livelli che si distinguono tra loro sulla base di tre attributi (configurabilità, efficienza multi-tenant e scalabilità).

Livello 1: ad-hoc/custom
Ogni cliente ha una versione personalizzata di hosting di applicazioni eseguita tramite la propria istanza sui server dell'host. La migrazione di una tradizionale applicazione non collegata in rete a questo livello di SaaS richiede in genere il minimo sforzo di sviluppo, e riduce i costi operativi attraverso il consolidamento e l'amministrazione di server.

Livello 2: configurabile
Questo aggiunge flessibilità ai programmi, attraverso metadati configurabili, così molti clienti possono utilizzare istanze separate del codice dell'applicazione stessa. Questo permette al venditore di soddisfare le diverse esigenze dei clienti attraverso le opzioni di configurazione dettagliate, semplificando nel contempo una base di codice comune di manutenzione e aggiornamento.

Livello 3: configurabile, multi-tenant-efficient

Questo aggiunge multi-tenant per il secondo livello, così che una singola istanza di programma serva a tutti i clienti. Ciò consente un utilizzo più efficiente delle risorse server senza un'apparente differenza per l'utente finale, ma limita la scalabilità.

Livello 4: scalabile, configurabile, multi-tenant-efficient

Aggiunge scalabilità attraverso un'architettura multi-livello che supporta un server farm[2] con un carico bilanciato di istanze identiche che girano su un numero variabile di server. Il provider può regolare la capacità del sistema per soddisfare la domanda, aggiungendo o rimuovendo i server senza alterare ulteriormente l'architettura software.

2.1.2 Principali caratteristiche

- *Disponibilità on-demand*: dopo aver acquistato l'accesso a un sistema SaaS, è possibile utilizzarlo da qualsiasi luogo e in qualunque momento.
- *Disponibilità tramite un qualsiasi browser*: le applicazioni SaaS non richiedono mai l'installazione di software sul client. Le attività vengono gestite da posizioni centrali, invece che presso la sede di ciascun cliente, consentendo ai clienti di accedere alle applicazioni da remoto via web. Per utilizzarle basta quindi un browser, con un eventuale plug-in, e una connessione internet.
- *Scalabilità*: Per supportare la scalabilità l'applicazione è installata su più computer.
- *Manutenzione del sistema (backup, aggiornamenti di sicurezza, etc)*: spesso incluso nel servizio. La funzione di aggiornamento centralizzata evita così la necessità per gli utenti finali di scaricare patch e aggiornamenti.
- *Aggiornamenti frequenti*: le applicazioni SaaS sono aggiornate più frequentemente dei software tradizionali, in molti casi su base settimanale o mensile, questo perché:

2 Il termine **Server Farm** (letteralmente *Fattoria di Server*) (anche chiamata **webfarm**) è utilizzato per indicare una serie di server collocati in un ambiente unico in modo da poterne centralizzare la gestione, la manutenzione e la sicurezza.

a) l'applicazione è ospitata centralmente, in modo che le nuove versioni possano essere messe in atto senza richiedere ai clienti di installare fisicamente un nuovo software.
b) l'applicazione ha solo una singola configurazione, rendendo i test di sviluppo più veloci.
c) il venditore delle applicazioni ha accesso a tutti i dati dei clienti, accelerando così la progettazione e i test.

- *Sicuro*: anche se gli utenti con elevati requisiti di sicurezza (ad esempio, le grandi aziende) possono trovare in SaaS un problema di sicurezza
- *Costi legati all'effettivo utilizzo*: i sistemi SaaS non richiedono alcun investimento iniziale in termini di infrastruttura o di setup e il loro utilizzo viene fatturato esclusivamente in base ai moduli che vengono utilizzati e al tempo d'utilizzo effettivo. Quando un sistema SaaS non viene più usato non si paga più nulla. Di conseguenza, il costo di setup iniziale per SaaS è di solito inferiore al software equivalente che si comprerebbe normalmente con la licenza.
- *Facilita l'aggregazione dei dati*: invece di raccogliere dati da più origini con schemi di database differenti, tutti i dati per tutti i clienti vengono memorizzati in un unico schema di database (cioè, multi-tenant) semplificando l'esecuzione di query su client.
- *Requisiti IT molto ridotti*: se non vi è necessità di procurarsi alcun server e l'unica cosa che è necessaria è una connessione a internet, non è necessaria alcuna infrastruttura IT particolare. Sebbene i sistemi SaaS possano a volte richiedere un minimo di competenze tecnologiche per essere configurate, queste sono alla portata di normali utenti esperti e non richiedono amministratori di rete di particolare esperienza.

Alcune applicazioni SaaS sono free per l'utente, con introiti ricavati da fonti alternative come la pubblicità, o l'aggiornamento di prezzi per una migliore funzionalità. Esempi di applicazioni SaaS free includono grandi operatori quali Gmail e Google Docs, così come piccoli provider come Wave Accounting (contabilità gratuito) e Freshbooks (time tracking freemium e fatturazione).

SaaS è diventato un modello comune per molte applicazioni aziendali compresa contabilità, customer relationship management (CRM), pianificazione delle risorse aziendali (ERP), fatturazione, gestione delle risorse umane (HRM), gestione dei contenuti (CM) e gestione del service desk.

Possibili casi in cui non è consigliato usare SaaS:

- quando i dati vengono memorizzati sui server del venditore, la sicurezza dei dati diventa un problema;
- quando le applicazioni richiedono tempi di risposta inferiori al secondo; in quanto essendo ospitate nella nuvola, lontano dagli utenti delle applicazioni, si crea della latenza nell'ambiente;
- quando le applicazioni o suite di applicazioni richiedono il controllo e i permessi al livello di accesso, è consigliabile delegare i diritti di amministrazione ai client. Questo riduce i tempi di amministrazione e i costi ma è difficile da implementare e può esporre le applicazioni a ulteriori minacce;
- quando le applicazioni richiedono l'accesso o l'integrazione con i dati attuali del cliente e le quantità di dati sono considerevoli o sensibili (ad esempio dati personali) si integra con il software da remoto (è costoso e / o rischioso);
- per architetture multi-tenant in quanto SaaS non consente la personalizzazione vera e propria delle applicazioni per clienti di grandi dimensioni.

2.2 PaaS

L'ambiente PaaS può essere descritto come un' intera piattaforma virtualizzata che comprende uno o più server (virtualizzato sul set di server fisici), sistemi operativi, applicazioni specifiche (come Apache e MySQL per le applicazioni Web-based) e ambienti di sviluppo completi. In alcuni casi, queste piattaforme possono essere predefinite e selezionate, in altri, è possibile fornire una immagine VM che contiene tutte le applicazioni necessarie.
La programmazione avviene utilizzando la specifica piattaforma di sviluppo messa a disposizione dal fornitore e si lascia che sia il fornitore a preoccuparsi dei dettagli della messa in produzione.

2.2.1 Tipi

- *Add-on development facilities*: queste strutture consentono la personalizzazione di applicazioni SaaS esistenti, e in qualche modo sono l'equivalente delle strutture di personalizzazione di macro linguaggio (macro language customization facilities) dotate di pacchetti di applicazioni software come Lotus Notes o Microsoft Word. (Spesso queste applicazioni richiedono sviluppatori PaaS e i loro utenti sottoscrivono abbonamenti per tali applicazioni SaaS.)

- *Stand alone development environments*: gli ambienti PaaS Stand-alone non includono tecniche, licenze o dipendenze finanziarie su specifiche applicazioni SaaS o web service, ma sono destinati a fornire un ambiente di sviluppo generalizzato.

- *Application delivery-only environments*: alcune applicazioni PaaS offrono debug e test, e servizi di hosting-level come anche la sicurezza e la scalabilità on-demand.

- *Open Platform as a Service*: consente allo sviluppatore di utilizzare qualsiasi linguaggio di programmazione, qualsiasi database, qualsiasi sistema operativo, qualsiasi server, ecc.

2.2.2 Principali caratteristiche

- *Services to develop, test, deploy, host and maintain applications in the same integrated development environment*: differenti offerte PaaS forniscono diverse combinazioni di servizi che supportano lo sviluppo del ciclo di vita dell'applicazione. PaaS, in generale, dovrebbe fornire tutte le opzioni di servizio in un ambiente di sviluppo integrato all'interno dell'attuale piattaforma di destinazione, con controllo del codice sorgente, controllo della versione, test dinamici su utenti multipli, roll out e roll back (ripristinare) con la capacità di controllare e tenere traccia di chi fa modifiche.

- *Web based user interface creation tools*: le offerte PaaS in genere forniscono un certo livello di sostegno per facilitare la creazione di interfacce utente, sia sulla base di standard come HTML e JavaScript o altre tecnologie Rich Internet Application come Adobe Flex, Flash ed AIR. Gli strumenti per la creazione di interfacce permettono di poterle definire per diversi profili utente (per funzione o competenza). PaaS offre una migliore “esperienza utente” incorporando i feedback in tutto il processo di creazione, progettazione, sviluppo, test, roll-out, produzione ... l'intero ciclo di vita.

- *Multi-tenant architecture*: le offerte PaaS tipicamente tentano di sostenere l'utilizzo dell'applicazione da molti utenti contemporaneamente, fornendo la gestione della concorrenza, scalabilità, fail-over e della sicurezza. L'architettura permette di definire il "rapporto di fiducia" tra gli utenti in sicurezza, accesso, distribuzione del codice sorgente, cronologia, profili utente (persone e dispositivo) e utilizzo delle applicazioni.

• *Integration with web services and data bases*: il supporto per le interfacce SOAP e REST consente alle offerte PaaS di creare composizioni di servizi web multipli, chiamate "mashup", come anche l'accesso ai database e il riutilizzo dei servizi di accesso mantenendoli all'interno di network privati.

• *Support for development team collaboration*: la capacità di formare e condividere il codice ad hoc con i team, migliora notevolmente la produttività delle offerte PaaS. I programmi, gli obiettivi, i team, i proprietari di diverse aree di responsabilità, i ruoli (designer, sviluppatori, tester, QC) possono essere definiti, aggiornati e monitorati sulla base di diritti di accesso.

• *Utility-grade instrumentation*: le offerte PaaS forniscono agli sviluppatori informazioni sul funzionamento interno delle loro applicazioni, e sul comportamento dei loro utenti. Alcune offerte PaaS utilizzano le informazioni sui comportamenti degli utenti per:

– stabilire se i servizi sono di valore per gli utenti/clienti;
– confrontare il valore dei diversi servizi;
– tener traccia dei costi delle attività in base e ricavi.

2.2.3 Esempi

Un esempio interessante di PaaS è Google App Engine.
App Engine è un servizio che permette di distribuire le applicazioni Web su un'architettura scalabile di Google. Con GAE le applicazioni devono essere scritte in Python o in Java, utilizzando gli strumenti di sviluppo messi a disposizione da Google, tra cui quelli per utilizzare il Google filesystem e i data repository.
Questo tipo di approccio funziona bene per applicazioni che devono essere implementate rapidamente e che non hanno particolari esigenze di integrazione.
Lo svantaggio di PaaS è che si diventa “prigionieri” del fornitore. Con Google, per

esempio, le applicazioni possono essere scritte solo nei linguaggi supportati usando delle API proprietarie e le applicazioni non possono funzionare al di fuori dell'infrastruttura messa a disposizione da Google.

Un altro esempio di PaaS è 10gen, il quale è sia una piattaforma cloud che un pacchetto open source scaricabile per la creazione di cloud private. 10gen fornisce funzionalità simili ad App Engine, con alcune differenze. Con 10gen, è possibile sviluppare applicazioni in Python così come programmare nei linguaggi Ruby e JavaScript. La piattaforma utilizza anche il concetto di sandbox per isolare le applicazioni e fornire un ambiente affidabile su un numero elevato di computer (costruito, ovviamente, su Linux), utilizzando il proprio server di applicazioni.

2.3 IaaS

Infrastructure as a Service è un modello di fornitura di servizi, compreso lo storage, hardware, CPU, memoria, server e componenti di rete, in base alle richieste dell'utente, che ne avrà un controllo completo e potrà gestire in maniera autonoma come utilizzare le risorse fornite dal cloud provider.

Vengono anche offerti servizi complementari, come la gestione delle policy di sicurezza e il monitoraggio delle risorse utilizzate. La maggior parte dei servizi di tipo IaaS fornisce delle macchine virtuali, le cui caratteristiche coincidono con la quantità di risorse richieste dall'utente; le modalità con cui viene eseguita effettivamente ogni macchina virtuale vengono gestite dal service provider e non sono visibili all'utente finale, a cui vengono forniti solamente gli strumenti per poter accedere a tale macchina, da qui in poi denominata anche istanza.

Ogni service provider IaaS espone delle interfacce (utilizzabili tramite linea di comando) che consentono ad ogni utente di richiedere l'avvio delle istanze di cui ha bisogno; il sistema si occuperà in maniera autonoma, e non visibile all'utente, di allocare le risorse nel data center dedicato e di effettuare le operazioni per rendere la macchina virtuale accessibile tramite Internet. Si possono scegliere, oltre alla quantità di risorse allocate per ogni istanza, anche quale sistema operativo utilizzare; solitamente i server provider rendono disponibili delle immagini di sistemi operativi già pronte all'uso, con installate le principali distribuzioni linux o windows, ogni utente però può scegliere di utilizzare una propria versione personalizzata del sistema operativo oppure modificarne una di quelle predefinite.

Utilizzando un servizio IaaS si può ottenere una macchina virtuale completamente personalizzabile in tutti i suoi parametri, sia hardware che software, di conseguenza questo tipo di soluzione si presta molto bene per tutte quelle situazioni che richiedono una gestione completa delle applicazioni che si vogliono eseguire. Ovviamente il

livello così basso di astrazione richiede che l'utente abbia le conoscenze necessarie a gestire l'installazione e l'esecuzione del software, di conseguenza la tipologia di utilizzatori di servizi IaaS è solitamente costituita da sviluppatori di applicazioni, che hanno conoscenze avanzate nel settore. Anche in questo caso il sistema di pagamento si basa sul modello pay-as-you-go, in cui vengono calcolate le ore-macchina che ogni utente consuma, il cui prezzo varia a seconda della quantità di risorse allocate per ogni macchina virtuale, a cui si aggiungono i costi relativi alla quantità di spazio occupato, calcolato in GB/mese e alla quantità di traffico generata in ingresso e in uscita dall'istanza.
Questo strato si differenzia da PaaS in quanto è fornito l'hardware virtuale senza un software stack[3].

2.3.1 Esempi

Uno dei principali fornitori di servizi IaaS è Amazon e i suoi Web Services (AWS).
Gli AWS sono basati sulla virtualizzazione pura. Amazon possiede l'hardware, controlla l'infrastruttura di rete e ha la proprietà diretta su tutto il software, dal sistema operativo in su (virtualizzazione pura).
Uno dei più famosi provider commerciali IaaS è Amazon Elastic Compute Cloud (EC2). In EC2, è possibile specificare una particolare macchina virtuale (sistema operativo e un set di applicazioni) e quindi distribuire le applicazioni su di essa o fornire una propria immagine per eseguire le macchine virtuali sui server.

Un altro esempio di EC2 IaaS è la piattaforma di Cloud Computing Enomalism. Enomalism è un progetto open source che fornisce un quadro di Cloud Computing con funzionalità simili a EC2. Enomalism è basato su Linux, con supporto sia per Xen che per il Kernel Virtual Machine (KVM). Ma a differenza di altre soluzioni di puro IaaS, Enomalism fornisce uno software stack basato sul framework TurboGears applicazione Web e Python.

3 Un set di programmi che lavorano insieme per produrre un risultato.

Anche AppNexus permette ai propri utenti di accedere dinamicamente a dei server, tuttavia fornisce agli utenti macchine dedicate sopra cui implementare la virtualizzazione. Così si è sicuri che le applicazioni non dovranno mai condividere risorse hardware fisiche con altri clienti e che ogni requisito, che richiede il controllo completo di tutte le risorse disponibili, sia soddisfatto (hardware dedicato).

Non sempre una soluzione di virtualizzazione pura può essere la soluzione migliore, in quanto:

- requisiti posti da regolamentazioni particolari potrebbero imporre che certe funzioni vengano implementate su hardware dedicato;
- requisiti prestazionali, in particolare per quanto riguarda l'input/output, potrebbero impedire il supporto di certe funzioni su hardware virtualizzato;
- requisiti d'integrazione con applicazioni legacy[4] che non prevedono alcun meccanismo di accesso via web.

Vediamo invece ora quali sono i **vantaggi** di utilizzare una soluzione orientata ai servizi come IaaS:

- *Riduzione dei costi di gestione*: in un’azienda la gestione e il mantenimento dei sistemi informatici richiede molte risorse, sia umane che monetarie, si richiede infatti in un primo momento una serie di investimenti iniziali molto importanti, per l’acquisto e il collocamento di tutte la macchine fisiche in un data center dedicato; oltre a questo tipo di investimenti la gestione interna delle risorse IT richiede anche di sostenere delle spese per l’energia di alimentazione e climatizzazione, oltre ai costi di eventuali interventi di manutenzione in caso di malfunzionamenti dell’hardware. Un’ultima categoria di spese che un’azienda deve affrontare sono quelle necessarie per la gestione del software dell’intero data center, bisogna infatti assumere personale specializzato che si occupi di mantenere operative e funzionanti le applicazioni installate. Se si utilizza un servizio di Infrastructure-as-a-Service, tutte le spese appena

4 Con questo termine si indicano i sistemi IT che utilizzano tecnologie meno recenti e per questo motivo sono molto difficili da interfacciare con i sistemi più recenti.

elencate non devono più essere sostenute, si richiede solamente di corrispondere al service provider il costo relativo alle risorse effettivamente utilizzate. Si eliminano quindi tutti costi operazionali dell'hardware, sostituendoli con le spese orarie di utilizzo delle istanze, si vedrà successivamente che a parità di utilizzo di risorse i costi sono circa gli stessi rispetto a una soluzione in-house, ma con una soluzione di tipo IaaS ogni azienda ha la possibilità di dimensionare il numero di risorse utilizzate in ogni momento, riuscendo così a deallocare le risorse non utilizzate in alcuni periodi, riallocandole quando invece la situazione lo richiede.

- *Elasticità, evitare fenomeni di Underprovisioning e Overprovisioning*: una delle caratteristiche peculiari dei servizi IaaS, come del resto di tutti i servizi di tipo Cloud, è quella di essere elastici, che nel caso specifico dei servizi infrastrutturali si traduce nella possibilità di allocare on-demand e in tempi molto brevi la quantità di risorse che si desidera. Sfruttando questa funzionalità si possono risolvere alcune problematiche tipiche di alcune applicazioni, soprattutto se queste sono state sviluppate per interagire con gli utenti tramite il Web o, più in generale, tutte quelle applicazioni che non possono prevedere con certezza il carico di lavoro che devono sopportare. Quando si implementano tipologie di applicazioni soggette a picchi di richieste, come può essere un servizio di e-commerce, se si sceglie una soluzione tradizionale è necessario dimensionare le capacità dei propri server non per riuscire a gestire il livello di carico medio, ma si devono poter soddisfare anche i picchi di richieste che si possono verificare. In alcuni casi si devono valutare anche l'aumento del carico di lavoro che si presentano in certi particolari mesi dell'anno, come ad esempio il periodo natalizio per i servizi di e-commerce. In queste situazioni si devono quindi valutare due situazioni opposte che possono accadere:

- *Overprovisioning*: è stata allocata una quantità troppo elevata di risorse rispetto al carico di lavoro generato dalle applicazioni, di conseguenza tutti i costi relativi alle risorse non utilizzate sono da considerarsi sprechi di denaro per l'azienda.
- *Underprovisioning*: non sono state allocate risorse sufficienti per poter

soddisfare tutte le richieste in arrivo, per questo alcuni utenti non potranno interagire con il sistema in maniera corretta. Gli effetti dal punto di vista economico causati da queste situazioni non sono spesso valutabili in maniera precisa, ma si possono rivelare molto importanti: non solo le richieste non soddisfatte non possono generare ricavi, ma vengono anche percepite come uno scarso livello di servizio da parte degli utenti, bisogna valutare infatti anche la possibile perdita di utenti causata dai disservizi dovuti a underprovisioning delle risorse.

Sfruttando l'elasticità di servizi offerti ogni azienda può ottimizzare la quantità di risorse allocata evitando di sovrastimare la quantità di risorse allocate e nello stesso momento avere le capacità di soddisfare carichi di lavoro maggiori rispetto alla norma. Visto che è possibile aumentare le risorse allocate in pochi minuti, i servizi IaaS sono molto indicati anche per le startup, le quali non conoscono a priori quanto i propri servizi verranno utilizzati e quindi il numero di risorse di cui avranno bisogno. L'utilizzo di servizi IaaS garantisce a tali attività le risorse necessarie per crescere anche in maniera molto importante, senza richiedere alcun tipo di garanzia o investimento preventivo.

3 Tipi di Cloud Computing

Ci sono tre tipi di Cloud Computing, come mostrato in Figura 3:

a) pubblico;

b) privato;

c) ibrido.

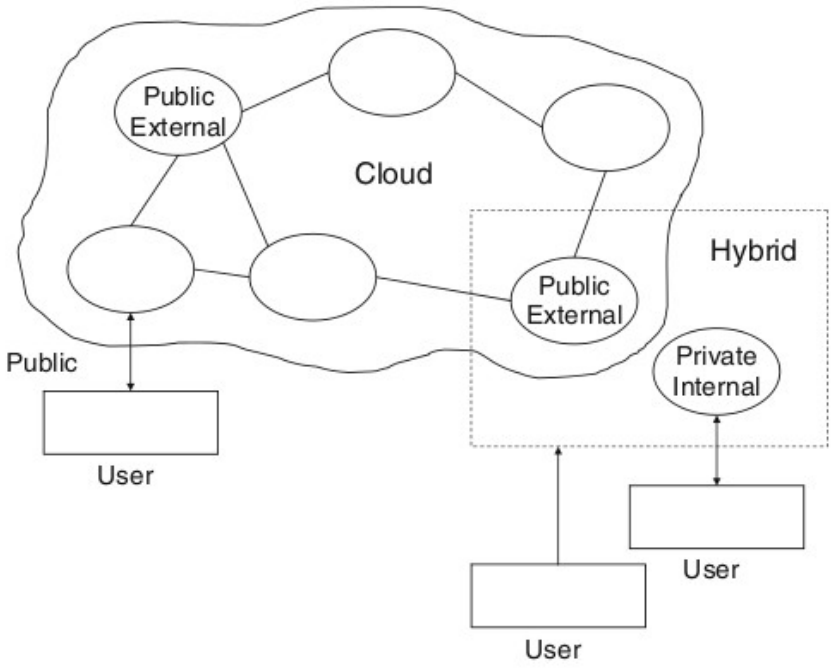

Figura 3 Mostra i tre tipi di Cloud Computing: pubblico, privato, ibrido.

Pubblico

La cloud di tipo pubblico, o cloud esterna, descrive il Cloud Computing nel senso più tradizionale, le risorse sono fornite dinamicamente attraverso internet tramite applicazioni web o servizi web da un provider esterno.

Le cloud pubbliche sono a conduzione di terze parti e le applicazioni che provengono da diversi client sono mescolate insieme nei server della cloud, negli storage system e nei network.

Privato

Le cloud private, o interne, fanno riferimento al Cloud Computing su network privati. Questo tipo di cloud è costruita per l'uso esclusivo di un client ed è utile per aziende che abbiano già significativi investimenti IT o che preferiscano il controllo totale sui dati, sulla sicurezza, sulla qualità del servizio e su tutti gli aspetti della propria infrastruttura. Possono essere costruite e gestite dall'organizzazione IT dell'azienda o da un cloud provider.

Ibrido

Un ambiente ibrido combina più modelli di cloud, pubblici e privati. Le cloud ibride introducono la complessità di determinare come distribuire applicazioni attraverso entrambe le cloud (pubbliche e private). Sono spesso utili per le funzioni di archiviazione e di backup, permettendo ai dati locali di poter essere replicati sulla cloud pubblica.

4 Tecnologie chiave che rendono possibile il Cloud Computing

Le tecnologie chiave che rendono possibile il Cloud Computing sono: la virtualizzazione, il Web service e l'architettura server-oriented, i service flows e i workflows, il Web 2.0 e il mashup.

4.1 Virtualizzazione

Il vantaggio del Cloud Computing è la capacità di virtuallizzare e condividere risorse tra diverse applicazioni con l'obiettivo di un miglior utilizzo del server.
La Figura 4 mostra un esempio di virtualizzazione.

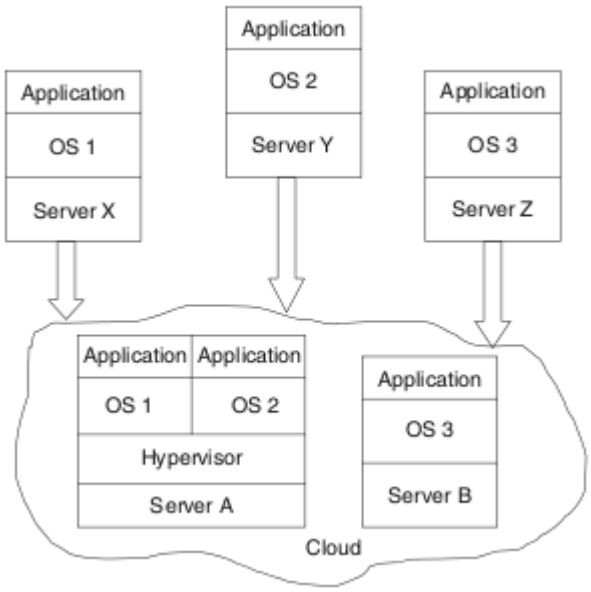

Figura 4 Mostra un esempio di virtualizzazione in cui da una situazione non-cloud, in cui vi è la necessità di tre server, si passa a una situazione di cloud con una conseguente diminuzione del numero di server.

In una situazione non-cloud come nell'esempio mostrato in figura vi sono tre piattaforme indipendenti per tre diverse applicazioni, ognuna in esecuzione sul proprio server. Nella nuvola, i server possono essere condivisi (o virtualizzati) tra più sistemi

operativi e applicazioni per utilizzare al meglio i server, con conseguente minor numero di server. Un numero inferiore di server significa un minor spazio richiesto (minimizzando l'impronta data center) e meno energia per il raffreddamento (riduzione del "carbon footprint").

Alcune tecnologie di virtualizzazione permettono persino di spostare una istanza di un server virtuale da un server fisico ad un altro senza che l'istanza smetta di funzionare.
Dalla prospettiva degli utenti o delle applicazioni che utilizzano un server virtuale, non esiste alcuna indicazione che permetta di stabilire che si sta utilizzando un server virtuale.
Le varie tecnologie adottano diversi approcci per implementare la virtualizzazione per esempio la tecnica delle macchine virtuali come VMware, i virtual network come VPN, o la soluzione di Amazon, che è un'estensione di un noto sistema di virtualizzazione open source chiamato Xen.

Xen fornisce un gestore di macchine virtuali (chiamato hypervisor) entro cui possono operare uno o più sistemi operativi ospite. L'hypervisor crea un'astrazione hardware che permette ai sistemi operativi ospite di condividere le risorse di uno stesso server fisico mantenendo tuttavia i vari server virtuali completamente isolati l'uno dall'altro.
Le macchine virtuali forniscono infrastrutture IT virtualizzate su richiesta, mentre i virtual network supportano gli utenti con un ambiente network su misura per accedere alle risorse cloud.

I servizi di Cloud Computing di solito sono sostenuti da grandi data center composti da migliaia di computer. Tali data center sono fatti per servire molti utenti e diverse applicazioni host. Per questo scopo, la virtualizzazione hardware può essere considerata come un perfetto “rimedio” per superare molti problemi di costruzione e di manutenzione dei data center. L'idea di virtualizzare le risorse di un sistema informatico, tra cui processori, memoria e dispositivi I/O, è consolidata da decenni, ma mira a migliorare la condivisione e l'utilizzo dei sistemi informatici. La

virtualizzazione hardware permette di condividere una singola piattaforma hardware da più sistemi operativi, ognuno dei quali è installato su una diversa macchina virtuale. Come illustrato in Figura 4.1, un livello software, il Virtual Machine Monitor (VMM), chiamato anche hypervisor, media l'accesso all'hardware fisico mostrando a ogni sistema operativo installato la relativa macchina virtuale.

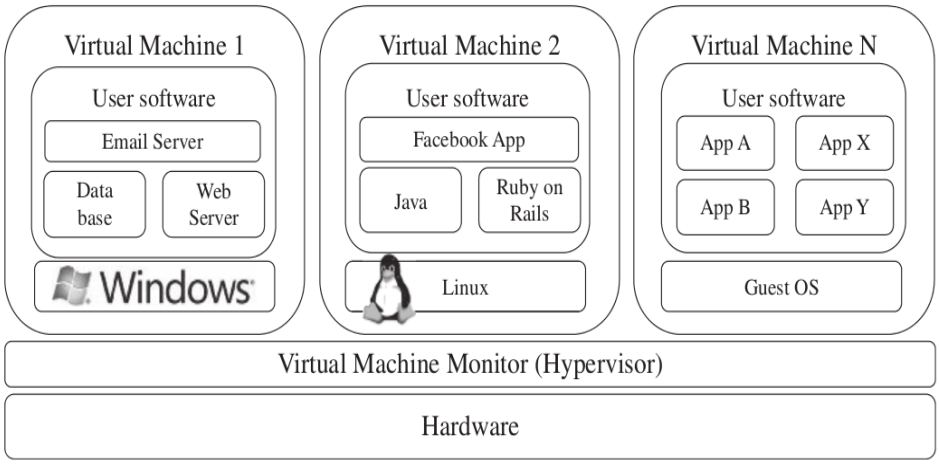

Figura 4.1 Mostra un server hardware virtualizzato che ospita tre macchine virtuali, ognuna in esecuzione su un diverso sistema operativo.

Quindi il compito del VMM è quello di consentire la condivisione da parte di più macchine virtuali di una singola piattaforma hardware. Ogni macchina virtuale è costituita oltre che dall'applicazione che in essa viene eseguita, anche dal sistema operativo utilizzato. Il VMM è il mediatore, nelle interazioni tra le macchine virtuali e l'hardware sottostante, che garantisce l'*isolamento* tra le macchine virtuali e la *stabilità* del sistema.
In generale deve offrire alle diverse macchine virtuali le risorse che sono necessarie per il loro funzionamento, come CPU, memoria e dispositivi I/O.

I benefici percepiti dalla virtualizzazione sono stati miglioramenti in materia di condivisione e di utilizzazione, una migliore gestibilità e maggiore affidabilità. Con

l'adozione della virtualizzazione su una vasta gamma di server e client, si possono sottolineare tre funzionalità di base relative alla gestione del carico di lavoro in un sistema virtualizzato, e cioè l'isolamento, il consolidamento e la migrazione.

L'*isolamento* degli ambienti di esecuzione è raggiunto poiché ogni macchina virtuale definisce un ambiente di esecuzione separato (sandbox) da quelli delle altre. Si ha così la possibilità di effettuare testing di applicazioni preservando l'integrità degli altri ambienti e del VMM. Inoltre si ha maggiore sicurezza e affidabilità in quanto eventuali attacchi da parte di malware o spyware sono confinati alla singola macchina virtuale e non hanno effetti sulle altre. Inoltre, si raggiungono migliori performance dato che l'esecuzione di una VM non influisce sulle prestazioni di un'altra VM.

Il *consolidamento* hardware consiste nella possibilità di concentrare più macchine (es. server) su un'unica architettura hardware per un utilizzo efficiente dell'hardware con il conseguente abbattimento dei costi hw e di amministrazione. Questa pratica viene utilizzata anche per superare le potenziali incompatibilità software e hardware in caso di upgrade, dato che è possibile eseguire più sistemi operativi contemporaneamente.

La *migrazione* del carico di lavoro (workload), riferito anche alla mobilità delle applicazioni, punta a facilitare la manutenzione hardware senza interrompere i servizi forniti dalle macchine virtuali, il workload balancing (alcuni prodotti prevedono anche meccanismi di migrazione automatica per far fronte in modo “automatico” a situazioni di sbilanciamento), e il disaster recovery. Ciò vien fatto incapsulando il sistema operativo ospite all'interno di una VM e permettendogli così di sospendersi, migrare a una piattaforma diversa, riprende immediatamente il funzionamento oppure in un secondo momento. Lo stato di una VM include un intero disco o immagine di una partizione, i file di configurazione, e un'immagine della sua RAM.

4.1.1 Tecniche per realizzare la virtualizzazione

Tre son le soluzioni più adottate per realizzare le funzionalità di virtualizzazione:

- *Traduzione delle binary instructions*: il virtual monitor traduce le richieste kernel delle macchine virtuali in modo da sostituire le istruzioni non virtualizzabili con sequenze di istruzioni che abbiano il medesimo effetto sulla virtual machine; tutte le istruzioni eseguite a livello utente invece, vengono inviate direttamente al processore, in modo da massimizzare le performance. Ogni VM viene dotata di tutte le funzionalità caratteristiche di un sistema fisico reale, quali bios, periferiche e memoria. Questo tipo di virtualizzazione consente di astrarre totalmente la VM dall'hardware sottostante, il sistema guest infatti è inconsapevole di essere virtualizzato e di conseguenza non richiede nessun tipo di modifica preventiva.

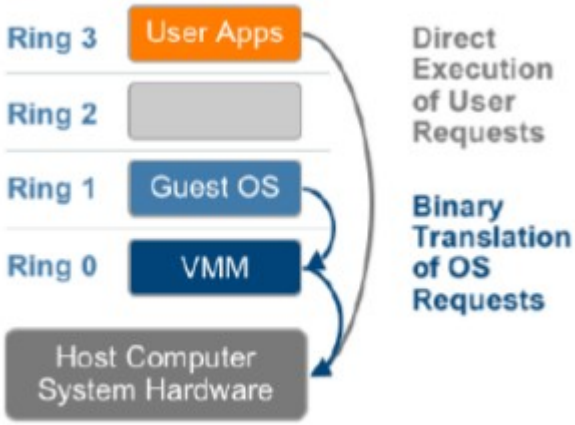

Figura 4.2 Schema di funzionamento per la Traduzione delle Binary Instructions

- *Para-virtualizzazione*: questo tipo di virtualizzazione si basa su un'interazione tra il sistema operativo Guest e l'hypervisor installato sulla macchina host, cercando di migliorare il più possibile le prestazioni del sistema. La para-virtualizzazione richiede che il sistema operativo Guest si basi su un kernel modificato, in grado di sostituire le chiamate non-virtualizzabili con Hypercalls che vengono inviate direttamente

all’hypervisor sull’Host. Dato che si richiede l’utilizzo esclusivo di sistemi modificati, questo metodo di virtualizzazione risulta poco portabile e scarsamente compatibile (non è infatti possibile installare OS non modificabili come Windows). È da sottolineare il fatto che prodotti basati su questo tipo di architettura (come Xen) risultino molto performanti e facilmente scalabili.

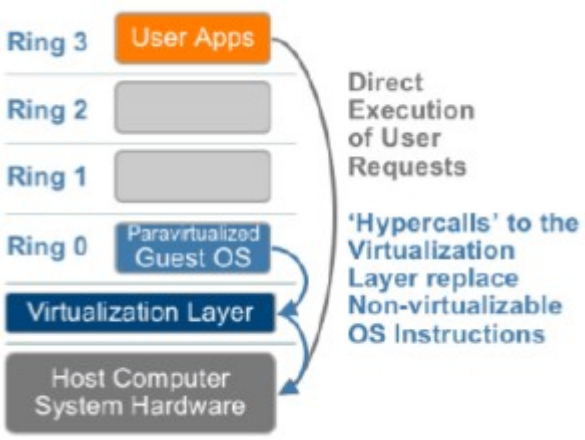

Figura 4.3 Schema di funzionamento della para-virtualizzazione

• *Virtualizzazione mediante il supporto Hardware*: i produttori di CPU hanno introdotto recentemente (2006) alcune funzionalità sui propri processori per facilitare le tecniche di virtualizzazione. Intel Virtualization Tecnology (VT-x) e AMD-V consentono per alcune istruzioni un nuovo livello di esecuzione al di sotto del ring 0 (OS), accessibile dal Virtual Monitor. In questo modo tutte le richieste che dovevano essere tradotte o gestite dalla para-virtualizzazione, vengono intercettate e inviate direttamente alla CPU, che le esegue normalmente, mantenendo tutte le informazioni dell’ambiente di esecuzione della macchina guest (quali registri, ecc). Questo sistema però non sempre consente di ridurre l’overhead di virtualizzazione, infatti ogni transizione dal Virtual Monitor alla VM (VMEntry) e vice-versa (VMExit), comporta un numero elevato di cicli di CPU, che va dalle centinaia alle migliaia di cicli.

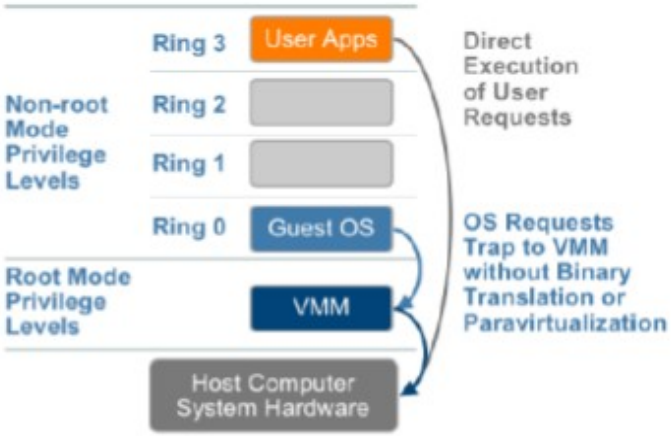

Figura 4.4 Schema di funzionamento della virtualizzazione utilizzando il supporto hardware

Le più note piattaforme VMM sono Xen, KVM e VMWare.

Xen

L'hypervisor Xen è nato come progetto open-source ed è servito come base per altri prodotti di virtualizzazione, sia commerciali che open-source. Ha aperto la strada al concetto di *para-virtualizzazione* (il VMM presenta un'interfaccia diversa da quella dell'architettura hardware), su cui il sistema operativo ospite, per mezzo di un apposito kernel, è in grado di interagire con l'hypervisor, migliorando così, in modo significativo, le prestazioni. Oltre a una distribuzione open-source, Xen costituisce una base degli hypervisor commerciali, in particolare Citrix XenServer e Oracle VM.
Il VMM si occupa della virtualizzazione della CPU, della memoria e dei dispositivi per ogni macchina virtuale.

Esso funziona come anello di congiunzione tra l’host fisico ed i sistemi operativi delle virtual machine (VM) che vengono create dall’utente, fungendo da interfaccia per la richiesta di servizi di Cloud Computing (in particolare per risorse hardware quali RAM, CPU, ecc.) e per l bilanciamento del carico di memoria tra le varie VM attive sull’host. Xen Hypervisor è in grado di supportare processori x86, x86-64, Power PC, ARM e Itanium e permette il funzionamento di sistemi operativi come Linux, Windows, Solaris, NetBSD e FreeBSD.

Xen dispone di un'interfaccia di controllo in grado di gestire la divisione di queste risorse tra i vari domini. L'accesso a questa interfaccia di controllo è ristretta: può essere controllata solo utilizzando una VM privilegiata denominata *domain 0* (avente sistema Linux). Questo dominio utilizza l'applicazione software che gestisce il controllo di tutta la piattaforma, tale software di controllo è eseguito nel domain 0 separato dallo stesso hypervisor (separazione dei meccanismi dalle politiche, all'interno del sistema). Ovvero fa da collegamento tra i vari sistemi operativi dei domain U e l'hardware fisico, e consente all'utente la creazione e la gestione dei domain U.

I domain U sono domini non privilegiati, cioè non hanno accesso diretto alle risorse hardware e vengono gestiti dal domain 0. Ciascuno di essi può presentare un sistema operativo con kernel parzialmente modificato (DomainU PV) oppure un sistema operativo con kernel non modificabile (DomainU HVM), che sfrutta un particolare hardware per la virtualizzazione detto Hardware Virtual Machine. I Domain U PV possono ospitare sistemi operativi quali Linux, Solaris, FreeBSD e gli altri principali open source. I Domain U HVM permettono invece il funzionamento di tutti i sistemi Microsoft.

La di sicurezza è garantita dal completo isolamento tra i domini (e tra questi e l'hypervisor).

KVM

Il kernel-based Virtual Machine (KVM) è il modulo di virtualizzazione per Linux. Ha fatto parte del kernel di Linux dalla versione 2.6.2, venendo così supportato nativamente da diverse distribuzioni. Inoltre, attività come la gestione della memoria e lo scheduling vengono svolte da features del kernel esistenti, rendendo KVM più semplice e più piccolo rispetto agli hypervisor che prendono il controllo dell'intera macchina.

KVM sfrutta la virtualizzazione hardware-assisted, che migliora le prestazioni e permette di supportare sistemi operativi ospiti inalterati; attualmente, supporta diverse versioni di Windows, Linux e UNIX.

VMWare

VMWare è un software in grado di creare sulla nostra macchina, computer virtuali dove installare ed eseguire diversi sistemi operativi contemporaneamente, in finestre separate o a schermo intero. Il sistema operativo della macchina fisica (su cui gira VMWare) è detto *host*, mentre il sistema operativo ospitato è detto *guest*. Il programma crea un'hard disk virtuale ed emula tutto l'hard disk di un PC dove è possibile installare qualsiasi sistema operativo lasciandovi la possibilità di scegliere quanta memoria RAM allocare alla macchina e le dimensioni dell'hard disk virtuale. Sulla macchina virtuale è possibile installare un qualsiasi sistema operativo, dos, Windows, Linux, ecc.

La macchina virtuale “parte” in una finestra del sistema operativo principale ed emula in tutto e per tutto un nuovo PC che dispone di un proprio processore e di un proprio BIOS.

ESXi è un VMM di VMWare. Si tratta di un hypervisor bare-metal, ciò significa che si installa direttamente sul server fisico, mentre altri potrebbero richiedere un sistema operativo host. Esso fornisce le tecniche avanzate di virtualizzazione del processore, della memoria e dell'I/O. In particolare, attraverso il memory ballooning e la condivisione delle pagine di memoria aumenta la densità di macchine virtuali all'interno di un singolo server fisico.

Per poter girare, VMWare ha bisogno di una piattaforma con un sistema operativo multitasking come Linux o Windows.

Xen, KVM e VMWare a confronto

KVM si basa su CPU ad alte prestazioni con un supporto limitato per la para-virtualizzazione ma fa del suo punto di forza l'utilizzo di istruzioni native per la virtualizzazione. È difficile ottenere massime prestazioni in un ambiente KVM virtualizzato senza aver sotto hardware potenti.

Al contrario Xen offre un forte sostegno per la para-virtualizzazione che consente di

eseguire un SO con un kernel modificato (per guests Windows e Linux) con prestazioni eccellenti.

Per adoperare KVM è necessario un computer con un processore compatibile mentre per adoperare Xen (e la para-virtualizzazione) bisogna usare un sistema operativo modificato.

Xen è un hypervisor esterno; assume il controllo della macchina e divide le risorse tra i guests. KVM è parte di Linux e utilizza lo scheduler e il memory management di Linux. Ciò significa che KVM è molto più piccolo e semplice da utilizzare ma è anche più ricco di funzioni.

KVM è eseguito sui processori x86 hvm, mentre Xen permette di essere eseguito anche su sistemi operativi modificati su processori x86 non-hvm utilizzando la tecnica della para-virtualizzazione. KVM non supporta la para-virtualizzazione per la CPU, ma può supportarla per i device driver per migliorare le prestazioni di I/O.

VMWare è un prodotto proprietario mentre KVM è un software open-source rilasciato sotto licenza GPL.
Mentre VMWare viene eseguito su un sistema operativo host completamente standard e non modificato, Xen utilizza un altro paradigma. Al gradino più basso, vicino all'hardware, ha un piccolo hypervisor “bare-metal”. Su questo primo strato viene eseguito il sistema operativo guest che ha i privilegi per “parlare” con l'hardware (utilizzando i driver hardware) e per farne partire altri senza privilegi.

Xen è supportato su distribuzioni GNU/Linux e il loro kernel; KVM virtualizza completamente molti sistemi operativi tra cui le distribuzioni GNU/Linux, *BSD, Windows, Solaris; VMWare è disponibile sia per Windows che per Linux.

4.2 Web service e architettura server-oriented

Il Web service e il Service Oriented Architecture (SOA) non sono concetti nuovi; tuttavia rappresentano le tecnologie base per il Cloud Computing.
I cloud service sono tipicamente designati come Web service, i quali seguono gli standard industriali includendo WSDL, SOAP, e UDDI.
Un SOA organizza e gestisce Web service dentro la cloud; inoltre include anche un set di cloud service, che sono disponibili su varie piattaforme distribuite.

4.3 Service flows e Workflows

Il concetto di service flow e workflow si riferisce a una visione integrata di servizi forniti nella cloud.
I Workflow sono una delle più importanti aree di ricerca nel campo dei database e information systems.

4.4 Web 2.0 e Mashup

Il Web 2.0 è un concetto che si riferisce all'uso della tecnologia web e al web design per migliorare la creatività, la condivisione di informazioni e la collaborazione tra utenti.
Mashup è un'applicazione web che combina dati da più di una sorgente in un singolo storage tool integrato. Entrambe le tecnologie sono molto utili per il Cloud Computing.
La Figura 4.5 mostra un'architettura Cloud Computing nella quale una applicazione riutilizza vari componenti.

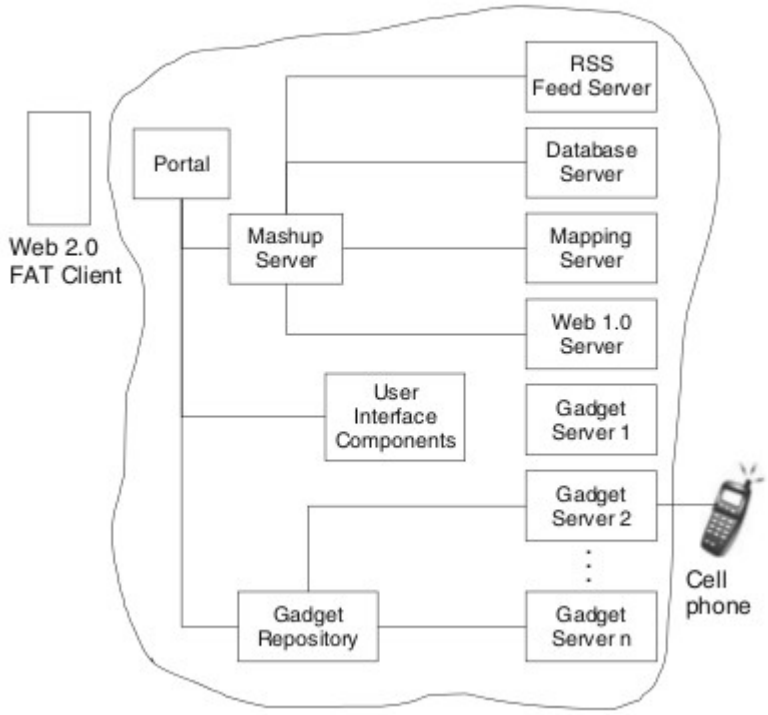

Figura 4.5 Mostra un'architettura basata sul Cloud Computing che utilizza vari componenti a differenti livelli.

I componenti in questa architettura sono dinamici, operanti in un modello SaaS, e con influenza SOA. I componenti più vicini all'utente sono più piccoli in natura e più riutilizzabili. I componenti nel centro contengono servizi aggregati ed estesi via server e portali mashup. I dati provenienti da un servizio (come l'indirizzo in un database) possono essere mashed up con una mappatura delle informazioni (come Yahoo o Google maps) per produrre una visione aggregata dell'informazione.

5 Caratteristiche del Cloud Computing

Vediamo qui di seguito le principali caratteristiche del Cloud Computing.

- *Scalabilità e servizi on-demand*: il Cloud Computing fornisce risorse e servizi per utenti su richiesta. Le risorse sono scalabili su diversi data center.
- *Interfaccia userc-centric*: le interfacce cloud sono indipendenti dalla locazione e possono essere accessibili da interfacce consolidate come Web Services e browser per internet.
- *Quality of service garantita (QoS)*: il Cloud Computing può garantire la qualità di servizio per gli utenti in termini di performance hardware/CPU, bandwidth e capacità di memoria.
- *Sistema autonomo*: i sistemi Cloud Computing sono sistemi autonomi gestiti trasparentemente dagli utenti. Tuttavia, software e dati nella cloud possono essere automaticamente riconfigurati e consolidati per una singola piattaforma dipendente dai bisogni dell'utente.
- *Prezzi*: i prezzi per le piattaforme e i servizi del Cloud Computing sono basati su tre aspetti: *storage* (in genere è misurato come quantità media giornaliera di dati memorizzati in GB per un periodo mensile), *bandwith* (è misurata dal calcolo dell'importo totale dei dati trasferiti in entrata e in uscita sulla piattaforma/servizio. Generalmente il trasferimento dati fra servizi all'interno della stessa piattaforma è gratuito), *compute* (è misurato come l'unità di tempo necessaria per eseguire un'istanza, o un'applicazione, o la manutenzione). I costi di implementazione di un'applicazione possono variare in base alla piattaforma scelta ma non richiederà mai preesistenti investimenti e gli utenti pagheranno per i servizi e le capacità come ne avranno bisogno.

5.1 La sicurezza nel Cloud Computing

Uno dei problemi critici nell'implementazione del Cloud Computing sta nelle virtual machine, che contengono applicazioni critiche e dati sensibili, di ambienti cloud pubblici e condivisibili. Utilizzare un servizio di Cloud Computing per memorizzare dei dati personali espone l'utente a potenziali problemi di violazione della privacy.
I dati risultano in possesso dell'azienda che quindi, se avesse un comportamento malevolo, potrebbe accedere ai dati degli utenti al fine di eseguire indagini di mercato o di profilazione dell'utente. Questi problemi possono essere parzialmente aggirati crittografando i dati sul server al fine di impedire alla società di accedere ai dati ma questa soluzione comunque non risolve del tutto il problema dato che il servizio di Cloud Computing potrebbe monitorare le attività degli utenti. Quindi, i potenziali utenti del Cloud Computing sono preoccupati per i suddetti problemi legati alla sicurezza.
Quel che ci si domanda è: “Gli utenti hanno ancora la stessa politica di controllo della sicurezza?” e “Si può dimostrare all'azienda che il sistema è ancora sicuro?”

Nei tradizionali data center, il comune approccio alla sicurezza include firewall, network segmentation, intrusion detection e prevention system.
I requisiti di sicurezza per i fornitori di Cloud Computing iniziano con gli stessi strumenti e tecniche come per i tradizionali data center; tuttavia la segmentazione fisica e la sicurezza basata su hardware non possono proteggere da attacchi tra macchine virtuali sullo stesso server.

Il rilevamento di intrusioni e sistemi preventivi devono essere in grado di rilevare le attività maliziose nel livello VM, indipendentemente dalla locazione della virtual machine all'interno dell'ambiente cloud virtualizzato. In sintesi, gli ambienti virtuali che distribuiscono i meccanismi di sicurezza sulle macchine virtuali includono firewall, intrusion detection and prevention (rilevamento e prevenzione), log inspection (registro di controllo), integrity monitoring, che rendono effettivamente una

VM cloud sicura e pronta da sviluppare.

Non da meno è il problema legato alla *continuità del servizio*, delegando a un servizio esterno la gestione dei dati e la loro elaborazione l'utente si trova fortemente limitato nel caso i suddetti servizi non siano operativi. Un eventuale malfunzionamento inoltre colpirebbe un numero molto elevato di persone contemporaneamente dato che questi sono servizi condivisi.

Infine bisogna considerare che tutto si basa sulla possibilità di avere una connessione internet ad alta velocità sia in download che in upload e che in caso di interruzione della connessione si ha la completa paralisi delle attività.

6 Cloud Computing sui sistemi Linux

Analizziamo qui di seguito OpenNebula, Haizea e Amazon EC2.

6.1 OpenNebula

OpenNebula è un gestore open-source di infrastrutture virtuali che orchestra tecnologie di storage, network e virtualizzazione per consentire il posizionamento dinamico di servizi multilivello su infrastrutture distribuite, combinando le risorse di data center e le risorse cloud remote in base alle politiche di assegnazione.
Inizialmente era stato concepito per gestire un'infrastruttura virtuale locale, ma, avendo al suo interno anche interfacce remote, è adatto anche per costruire cloud pubbliche (IaaS).
OpenNebula fornisce l'amministrazione interna alla cloud e le interfacce utente per la completa gestione della piattaforma cloud. La sua architettura è modulare e comprende diversi componenti specializzati collegabili tra loro.

Il modulo di base orchestra server fisici e i loro *hypervisor*, nodi di storage e network. Le operazioni di gestione vengono eseguite tramite i driver, che interagiscono con le API degli hypervisor, delle tecnologie di storage e di rete, e delle cloud pubbliche.
Il modulo *Scheduler*, che si occupa delle richieste di assegnamento di *macchine virtuali* in attesa agli host fisici, offre caratteristiche dinamiche di allocazione delle risorse. Gli amministratori possono scegliere tra differenti obiettivi di programmazione, quali collocare macchine virtuali in un minor numero di host o mantenere il load balance. Insieme ad Haizea anche OpenNebula supporta la prenotazione anticipata delle risorse e delle code di best-effort di leasing.

In sintesi, OpenNebula fornisce le seguenti funzionalità: Linux-based controller; CLI,

XML-RPC, EC2-compatible Query e OCA interface; Xen, KVM e VMWare backend; Interfaccia per ambienti cloud pubblici (Amazon EC2, ElasticHosts); virtual networks; allocazione dinamica delle risorse.

6.1.1 VM gestione dinamica con OpenNebula

OpenNebula è uno strumento aperto e flessibile che si inserisce in ambienti di data center esistenti per costruire qualsiasi tipo di distribuzione cloud. Può essere utilizzato principalmente come uno strumento di virtualizzazione per gestire l'infrastruttura virtuale, che viene solitamente indicato come una cloud privata. OpenNebula supporta una *cloud ibrida* per combinare l'infrastruttura locale con l'infrastruttura basata sulla cloud pubblica, consentendo ambienti di hosting altamente scalabili.

È possibile crearla sfruttando le risorse dei propri nodi e quelle di provider esterni come Amazon EC2 o Elastic Hosting. A livello utente questo tipo di integrazione risulta trasparente, sarà infatti il sistema che deciderà autonomamente se affidare una VM a un nodo interno o a provider esterni. Per implementare questa funzionalità si deve configurare appositamente un “Adaptor” per il provider desiderato, indicando le proprie credenziali di accesso e altri parametri, quali il numero massimo di VM instanziabili.

Supporta anche le *cloud pubbliche*, fornendo le interfacce della cloud per esporre la sua funzionalità per la macchina virtuale, l'archiviazione e gestione della rete.
OpenNebula è un'alternativa open-source agli strumenti commerciali per la gestione dinamica delle macchine virtuali su risorse distribuite.
Questo strumento sostiene diverse linee di ricerca, quali: prenotazione anticipata di capacità, controllo probabilistico, ottimizzazione del posizionamento, modelli di risorse per la gestione efficiente dei gruppi di macchine virtuali, elasticità di supporto, e così via. Queste linee di ricerca rispondono alle esigenze di entrambi i tipi di nuvole, vale a dire, pubbliche e private.

6.1.2 Architettura OpenNebula

L'architettura OpenNebula (vedi Figura 6) comprende diversi componenti specializzati in diversi aspetti della gestione VI.

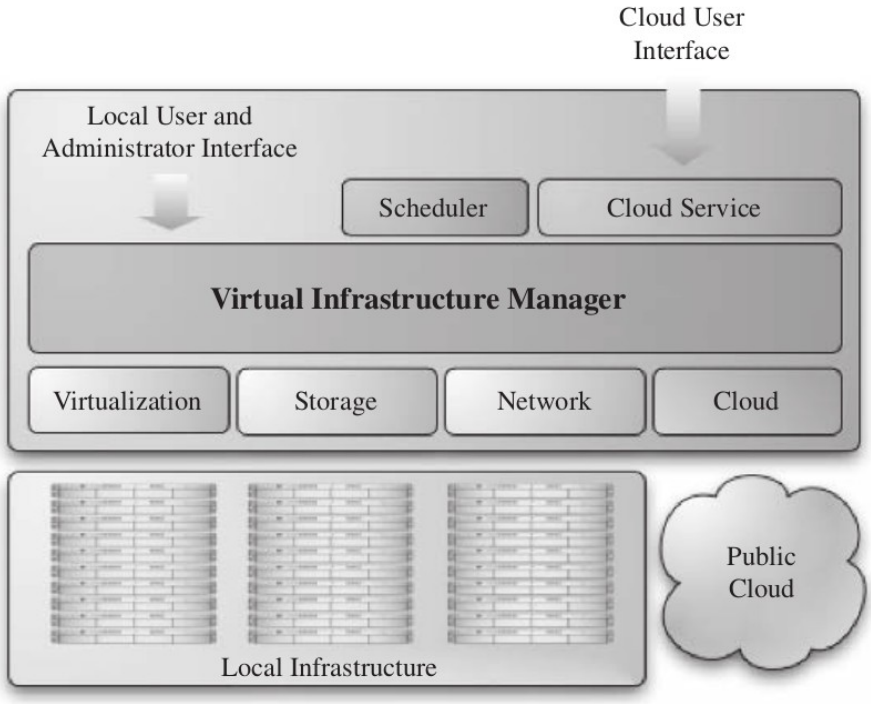

Figura 6 Mostra l'architettura di OpenNebula: l'esistenza di cloud pubbliche e private e anche le risorse gestite dal suo gestore virtuale.

Per controllare il ciclo di vita di una macchina virtuale, il nucleo OpenNebula orchestra tre diverse aree di gestione: *image and storage technologies* per la preparazione di immagini disco per le macchine virtuali; *network fabric* (come server, firewall o switches) per la fornitura di macchine virtuali con un ambiente di rete virtuale; *hypervisor* per la creazione e il controllo macchine virtuali (sono supportati KVM, Xen e VMWare).

Il nucleo esegue le operazioni di storage, di rete o la virtualizzazione tramite i driver pluggable. Così, OpenNebula non è legata ad un ambiente specifico, fornendo uno strato uniforme di gestione indipendentemente dall'infrastruttura sottostante.

Grazie a questo suo design altamente modulare, è quindi facilitata l'integrazione con

qualsiasi piattaforma di virtualizzazione e con qualsiasi componente di terze parti nella cloud (come toolkit, gestori di immagini virtuali, gestori di servizi, e scheduler VM). Ad esempio, specifica tutte le azioni relative alla creazione di un'immagine del disco VM (trasferendo l'immagine, l'installazione del software su di esso, e così via) in termini di hooks (ganci). Anche se OpenNebula include un "transfer manager" di default che utilizza questi hooks, può sfruttare anche i transfer manager esistenti o contestualizzare un'immagine VM semplicemente scrivendo il codice che interfaccia gli hooks e il software di terze parti.

Oltre a gestire individualmente il ciclo di vita delle macchine virtuali, vi è anche il nucleo per supportare la distribuzione di servizi; tali servizi includono tipicamente un insieme di componenti interconnessi (ad esempio, un server Web e database back end) che richiedono diverse macchine virtuali. In questo modo, si può trattare un gruppo di macchine virtuali collegate tra loro come un'entità di prima classe in OpenNebula. Oltre a gestire le macchine virtuali come un'unità, il nucleo gestisce anche l'erogazione di informazioni di contesto (come l'indirizzo IP del server Web, certificati digitali e licenze software) per le macchine virtuali.

Un componente *scheduler* separato prende le decisioni di collocamento VM. Più specificamente, lo scheduler ha accesso alle informazioni su tutte le richieste che OpenNebula riceve e, sulla base di queste richieste, tiene traccia degli stanziamenti attuali e futuri, creando e aggiornando un calendario delle risorse e inviando le istruzioni di distribuzione al nucleo OpenNebula. Lo scheduler di default fornisce una politica di pianificazione che pone macchine virtuali su risorse fisiche in base ad un algoritmo di ranking che l'amministratore può configurare. Si basa su dati in tempo reale da entrambe le macchine virtuali in esecuzione e dalle risorse fisiche disponibili.
Infine, OpenNebula supporta le cloud ibride utilizzando i driver per interfacciarsi con le cloud esterne.
Questo consente alle aziende di integrare l'infrastruttura locale con capacità di calcolo in grado di soddisfare le richieste di punta, servire meglio le richieste di accesso, o

implementare strategie per una maggiore disponibilità. OpenNebula attualmente include un driver EC2, che può presentare le richieste di Amazon EC2 e di Eucalyptus, così come un driver ElasticHosts.

6.1.3 Gestione distribuita di infrastrutture virtuali

La gestione delle macchine virtuali in un pool di risorse fisiche distribuite è un problema fondamentale nelle nuvole IaaS, che richiedono l'uso di un gestore dell'infrastruttura virtuale. Per affrontare alcune delle carenze delle attuali soluzioni ci viene incontro l'infrastruttura virtuale open source OpenNebula.
OpenNebula è in grado di gestire gruppi di macchine virtuali interconnesse (con supporto per Xen, KVM e piattaforme VMWare) all'interno dei data center e delle cloud private che coinvolgono una grande quantità di server fisici e virtuali. Può anche essere utilizzato per costruire cloud ibride interfacciandosi con cloud siti remoti.
Qui di seguito descriviamo come OpenNebula modella e gestisce le macchine virtuali in un'infrastruttura virtuale.

6.1.4 Modello VM e ciclo di vita

L'obiettivo primario di OpenNebula è quello di gestire macchine virtuali. All'interno di OpenNebula, una VM è modellata per avere i seguenti attributi:

- una capacità in termini di memoria e CPU.
- un insieme di schede di rete collegati a una o più reti virtuali.
- un insieme di immagini disco. In generale, potrebbe essere necessario trasferire alcuni di questi file immagine da/verso la macchina fisica la VM in esecuzione dentro
- un file di stato del file (opzionale) o di ripristino che contiene l'immagine della memoria di una VM in esecuzione, l'hypervisor più alcune informazioni specifiche.

Il ciclo di vita di una VM all'interno OpenNebula segue diverse fasi:

- *Selezione delle risorse*: una volta che una VM è richiesta a OpenNebula, viene

creato un piano di realizzazione la VM. Lo scheduler di default di OpenNebula fornisce un'implementazione di una politica di scheduler a livello, consentendo agli amministratori di configurare lo scheduler in modo da dare priorità alle risorse che sono più adatte per la macchina virtuale, utilizzando le informazioni dalle macchine virtuali e dagli host fisici.

- *Preparazione delle risorse*: le immagini disco della VM vengono spostate all'indirizzo delle risorse fisiche. Durante il processo di avvio, la VM è contestualizzata, un processo in cui le immagini disco sono specializzate per lavorare in un dato ambiente. Ad esempio, se la VM è parte di un gruppo di macchine virtuali che offrono un servizio (un cluster di calcolo, un'applicazione DB-based, ecc), la contestualizzazione potrebbe comportare la realizzazione della rete e dell'hostname della macchina, o la registrazione la nuova VM con un servizio (ad esempio, il nodo di testa in un cluster di calcolo). Sono disponibili diverse tecniche per contestualizzare un nodo di lavoro, compreso l'utilizzo di un sistema di installazione automatica (per esempio, Puppet o Quattor), un context server o l'accesso a un'immagine disco con i context dati per il nodo di lavoro.
- *Creazione della VM*: la VM viene avviata dalla risorsa hypervisor.
- *Migrazione della VM*: la VM potenzialmente viene spostata a risorse più adatte (ad esempio, per ottimizzare il consumo energetico delle risorse fisiche).
- *Terminazione della VM*: quando la VM sta per cessare l'attività, OpenNebula può trasferire nuovamente le sue immagini disco in una nota locazione. In questo modo, le modifiche della VM possono essere mantenute per un uso futuro.

6.1.5 Gestore di VM

OpenNebula gestisce il ciclo di vita delle macchine virtuali orchestrando tre diverse aree di gestione: *virtualizzazione,* interfacciandosi con un hypervisor di risorse fisiche, come Xen, KVM, o VMWare, per controllare (ad esempio, l'avvio, lo stop, o l'arresto), la VM; *gestione delle immagini* mediante il trasferimento le immagini delle macchine virtuali da un image repository alla risorsa selezionata e creando in-the-fly immagini

temporanee; e *networking* creando reti locali (LAN) per interconnettere le macchine virtuali e tenere traccia degli indirizzi MAC in ogni rete.

6.1.6 Virtualizzazione

OpenNebula gestisce le VM interfacciandosi con la tecnologia di virtualizzazione delle risorse fisiche (Xen, KVM o VMWare) utilizzando un set di driver collegabili tra loro che separano il processo di gestione dalla tecnologia sottostante.
Così, ogni volta che il nucleo deve gestire una macchina virtuale, utilizza comandi di alto livello come "start VM", "stop VM," e così via, che vengono tradotti dai driver in comandi che il gestore della macchina virtuale è in grado di capire. Con questo “sdoppiamento” del nucleo OpenNebula aggiunge supporto per ulteriori gestori di macchina virtuale semplicemente richiedendo la scrittura di un driver per esso.

6.1.7 Gestione delle immagini

Le macchine virtuali sono supportate da un insieme di dischi o immagini virtuali, che contengono il sistema operativo e qualunque altro software aggiuntivo necessario per la VM.
OpenNebula presuppone che esista una image repository, che può essere qualsiasi supporto di memorizzazione o servizio (locale o remoto), che contiene l'immagine di base delle macchine virtuali.
Ci sono un certo numero di diverse possibili configurazioni a seconda delle esigenze dell'utente. Ad esempio, gli utenti potrebbero voler tutte le loro immagini inserite in una locazione separata con solo accesso HTTP. In alternativa, le immagini possono essere condivise tramite NFS tra tutti gli host. OpenNebula mira ad essere sufficientemente flessibile per supportare le configurazioni di molte possibili differenti immagini.
OpenNebula utilizza i seguenti concetti per il suo modello di gestione delle immagini (Figura 6.1):

1) Image Repositories: si riferisce a qualsiasi supporto di memorizzazione, locale o remoto, che detiene le immagini di base delle macchine virtuali. Un image repository può essere un file server dedicato o un URL remoto da un provider, ma devono essere accessibili dal front-end di OpenNebula.

2) Virtual Machine Directory: è una directory sul nodo del cluster in cui una macchina virtuale è in esecuzione. Questa directory contiene tutti i file di distribuzione per l'hypervisor per avviare la macchina, i checkpoint e le immagini utilizzate o salvate (tutti specifici per quella VM). La directory deve essere condivisa per la maggior parte degli hypervisor per essere in grado di effettuare migrazioni dal vivo. Ogni immagine VM passa attraverso i seguenti passi lungo il suo ciclo di vita:

a) Preparation: implica tutte le modifiche necessarie da apportare per l'immagine della macchina in modo che sia pronta a offrire il servizio alla quale è destinata. OpenNebula presuppone che le immagini conformi a una particolare macchina virtuale siano preparate e messe in un archivio di immagini accessibile.

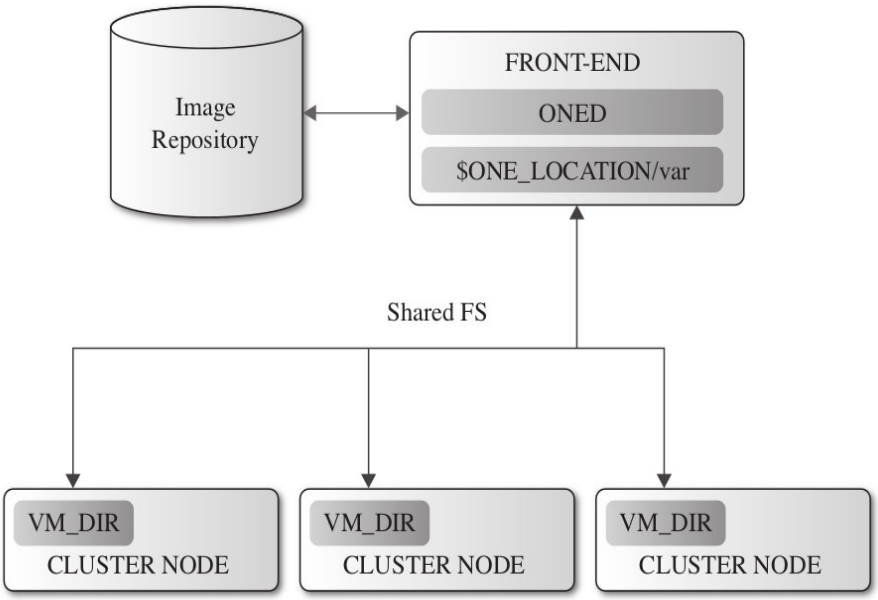

Figura 6.1 Mostra la gestione delle immagini in OpenNebula.

b) Cloning the image: significa prendere l'immagine dal repository e immetterla nella directory della VM nel nodo fisico in cui sta per essere eseguita prima che la VM sia effettivamente avviata. Se una immagine VM deve essere clonata, non verrà utilizzata l'immagine originale ma la copia. C'è un qualificatore (clone) per le immagini che può contrassegnarle come adatte per la clonazione o meno.

c) Save/remove: Se il salvataggio del qualificatore è disabilitato, una volta che la VM è stata arrestata, le immagini e tutte le relative modifiche andranno perse. Tuttavia, se il salvataggio è attivo, l'immagine verrà salvata per un successivo uso.

6.1.8 Networking

In generale, i servizi distribuiti su una nuvola richiedono diverse macchine virtuali interconnesse con una virtual application network (VAN) che è il collegamento primario tra di loro. OpenNebula crea dinamicamente queste VAN e traccia gli indirizzi MAC in rete per le macchine virtuali.

Gli host fisici che co-formano il “tessuto” delle nostre infrastrutture virtuali avranno bisogno di alcuni vincoli al fine di fornire, in modo efficace, i virtual network alle macchine virtuali. Pertanto, dal punto di vista della rete, possiamo definire il cluster fisico come un insieme di host, con una o più interfacce di rete, ciascuno dei quali collegato ad una rete fisica differente.

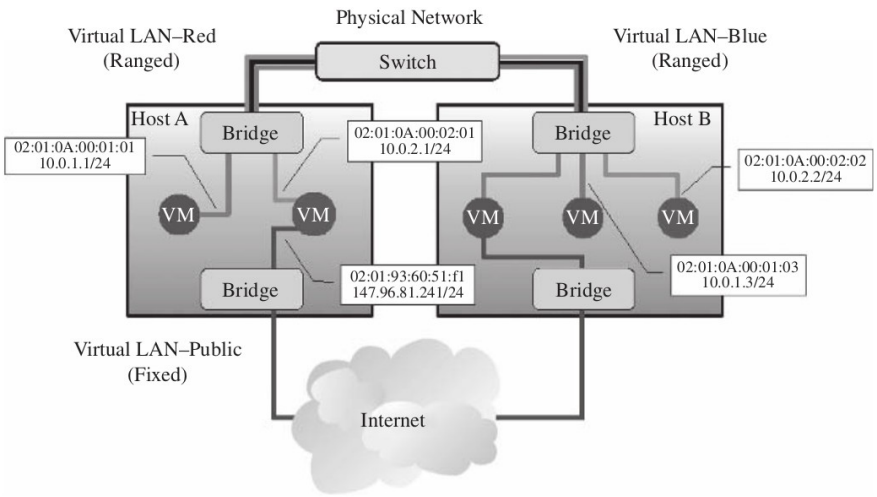

Figura 6.2 Mostra il modello di networking per OpenNebula.

Possiamo vedere nella Figura x due host fisici con due interfacce di rete ciascuno; quindi ci sono due reti fisiche diverse. C'è una rete fisica che collega i due host che usano uno switch, e c'è un altro che dà gli accessi host alla rete Internet. Questa è una possibile configurazione per il cluster, ed è quella che può essere utilizzata per fare sia VAN pubbliche che private per le macchine virtuali.

Muovendoci fino allo strato di virtualizzazione, possiamo distinguere tre VAN diverse. Una è mappata sulla parte superiore della rete Internet pubblica, e possiamo vedere un paio di macchine virtuali che ne traggono vantaggio. Queste due VM avranno accesso a Internet. Le altre due sono in cima alla rete fisica privata: la rossa e la blu. Le macchine virtuali collegate alla stessa VAN privata saranno in grado di comunicare tra loro, altrimenti verranno isolate e non saranno in grado di comunicare.

6.1.9 Vantaggi derivanti dall'utilizzo di OpenNebula

I benefici che OpenNebula è in grado di apportare al lavoro di utenti ed amministratori di risorse IT sono notevoli:

- *Gestione innovativa* di ambienti e servizi di Cloud Computing.
- *Architettura flessibile*, capace di creare e gestire combinazioni hardware e software di vario tipo.
- *Gestione centralizzata* di tutte le infrastrutture fisiche e virtuali create.
- *Capacità d'integrare* tra loro diversi tipi di servizi cloud.
- *Interfacce* semplici ed intuitive.
- *Elevata stabilità e scalabilità* anche con infrastrutture di grandi dimensioni.
- *Piattaforma completamente open source*, distribuita con licenza Apache.
- *Supporto di un'ampia community* formata da aziende, organizzazioni e professionisti del settore che collaborano costantemente per lo sviluppo di progetti di ricerca innovativi.

Da ciò che abbiamo visto possiamo quindi dire che OpenNebula offre un servizio

sicuro, scalabile, stabile ma soprattutto interoperabile.

6.2 OpenNebula e Haizea

OpenNebula è un gestore di VI che le organizzazioni possono utilizzare per implementare e gestire macchine virtuali, individualmente o in gruppi che devono essere co-schedulati sulle risorse locali o su nuvole pubbliche esterne.
Consente di automatizzare l'installazione di VM (la preparazione di immagini disco, la creazione di reti, e così via), indipendentemente dallo strato sottostante di virtualizzazione (Xen, KVM o VMware) o cloud esterne (EC2).

Haizea è un gestore di leasing delle risorse che può agire come uno scheduling back end per OpenNebula, fornendo funzionalità di leasing come ad esempio le prenotazioni anticipate (AR) e la precedenza delle risorse, che sono particolarmente rilevanti per cloud private.

Lo scopo di OpenNebula è quello di produrre una gestione di VI con un'architettura flessibile e aperta che le organizzazioni possano utilizzare per costruire cloud private/ibrido.
OpenNebula mira anche a superare le carenze delle VI:
• l'incapacità di adattarsi a nubi esterne;
• Le architetture monolitiche e chiuse che sono difficili da estendere o interfacciarsi con altri software, non permettendo una perfetta integrazione con le esistenti soluzioni di storage e di gestione della rete distribuite nei centri dati;
• una scelta limitata di politiche di collocamento preconfigurate (first fit, round robin, ecc.)
• una mancanza di supporto per la pianificazione, la distribuzione e la configurazione di gruppi di VM (per esempio, un gruppo di macchine virtuali che rappresentano un cluster, in cui tutti i nodi o sono distribuiti o non lo sono per niente, e dove alcune configurazioni di VM dipendono da altre).

Come già detto in precedenza, le tecnologie di virtualizzazione sono un fattore chiave del Cloud Computing. Le macchine virtuali sono utili anche per un'efficiente prenotazione delle risorse grazie alla possibilità di poter sospendere la loro attività e riprende senza modificare le applicazioni in esecuzione al loro interno. Tuttavia, con le macchine virtuali vi può essere il problema legato al loro sovraccarico di utilizzo:

- *Preparazione Overhead*: quando si utilizzano le macchine virtuali per implementare le prenotazioni, un'immagine disco VM deve essere preparata on-the-fly o trasferita al nodo fisico in cui è necessaria. Dal momento che un' immagine disco VM può avere una dimensione dell'ordine di gigabyte, questa “preparazione” può ritardare in modo significativo il tempo di avvio dei contratti di leasing. Questo ritardo può, in alcuni casi, essere inaccettabile per le prenotazioni che devono essere avviate in un momento specifico.
- *Runtime Overhead*: una volta che una VM è in esecuzione, le primitive di scheduling come il checkpoint e il resuming possono incorrere in un overhead significativo in quanto un intero spazio di memoria di una macchina virtuale deve essere salvato su disco, e poi letto da esso. La migrazione comporta il trasferimento di questa memoria salvata insieme all'immagine del disco VM. Questo overhead può comportare ritardi notevoli.

Il progetto Haizea è stato creato per sviluppare uno scheduler che possa efficacemente supportare le prenotazioni in anticipo in modo efficiente utilizzando il suspend/resume/migrate di macchine virtuali, ma riducendo al minimo l'overhead di utilizzo. La risorsa fondamentale di astrazione Haizea è il contratto di leasing, con tre tipi attualmente supportati:

- Advanced reservation leases: dove le risorse devono essere disponibili ad un tempo specifico.
- *Best-effort leases*: dove le risorse sono rilasciate non appena possibile e le richieste vengono immesse in una coda, se necessario.
- *Immediate leases*: dove le risorse vengono rilasciate quando richiesto o per

niente.

Il gestore di leasing Haizea, utilizzato come scheduling back-end per il gestore d'infrastruttura virtuale OpenNebula, permette di supportare questi tre tipi di contratti di leasing.

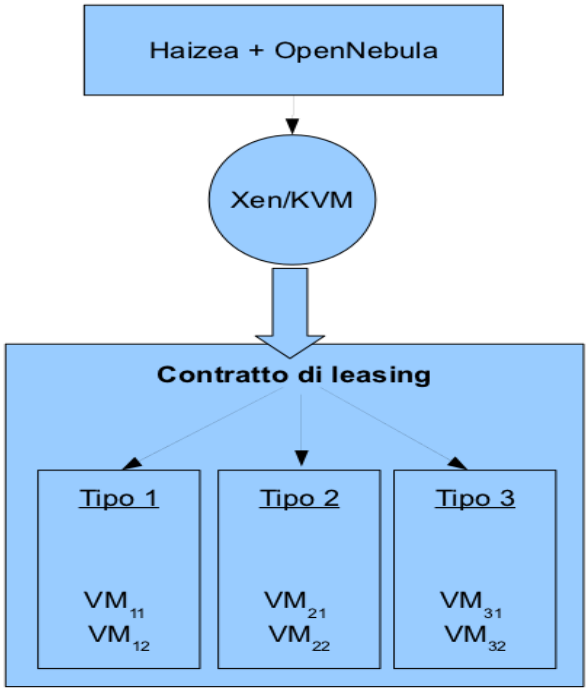

Figura 6.3 Mostra come Haizea e OpenNebula possano essere utilizzati insieme per la gestione di cluster attraverso l'utilizzo di Xen o KVM per distribuire i vari tipi di contratti di leasing istanziati come macchine virtuali.

Lavorando insieme possono quindi offrire contratti di leasing delle risorse come una fondamentale fornitura di astrazione e Haizea può funzionare con l'hardware reale attraverso OpenNebula.

Questa integrazione fornisce la soluzione di gestione di VI offrendo la capacità di prenotazione anticipata. Come mostra la tabella 6.4, altri gestori VI utilizzano la fornitura (leasing) immediata o il best-effort. Tuttavia, cloud private, in particolare quelle con risorse limitate, in cui non tutte le richieste sono immediatamente soddisfatte proprio a causa della mancanza di risorse, traggono beneficio dalle strategie più sofisticate di collocamento VM basate su code, priorità, e AR. Inoltre, il

servizio di fornitura delle cloud ha requisiti che non sono supportabili con solo un modello di fornitura immediato (per esempio, hanno bisogno di riserve di capacità in momenti specifici per soddisfare service-level agreement (SLA) o il picco di richieste di capacità).

Tool	Provisioning model	Default placement policies	Configurable placement policies	Support for hybrid cloud	Remote interface
Amazon EC2	Best effort	Proprietary	Proprietary	No	EC2 web services API
VMware	Immediate	Initial placement on CPU load and dynamic placement to balance average CPU or memory load and consolidate servers	No	Only when both local and external cloud use vSphere	vCloud API
OpenNebula	Best effort	Initial placement on requirement / rank policies to prioritize those resources more suitable for the virtual machine using dynamic information and dynamic placement to consolidate servers	Support for any static / dynamic placement policy	Driver-based architecture allow interfacing with multiple external clouds; supports EC2-compatible clouds and elastichosts	No
OpenNebula + Haizea	Immediate, best effort, and advance reservation (AR)	Dynamic placement to implement AR leases	VM placement strategies supporting queues and priorities	Driver-based architecture allow interfacing with multiple external clouds; supports EC2-compatible clouds and elastichosts.	No

Tabella 6.4 Mostra il confronto di strumenti virtuali che forniscono funzionalità di gestione delle infrastrutture.

Andiamo ora ad analizzare Haizea, i suoi modelli e algoritmi utilizzati per lo scheduling di questi contratti di leasing.

6.3 Haizea

Haizea è un'architettura di gestione leasing open-source basata su macchine virtuali. In poche parole, Haizea è una parte di software che, in combinazione con l'infrastruttura di gestione virtuale OpenNebula, può essere utilizzata per gestire Xen, KVM o un cluster VMWare, permettendo di distribuire diversi tipi di leasing che vengono istanziati come macchine virtuali.

Cos'è Haizea?

- *Haizea è un gestore di risorse*: Haizea è un componente software in grado di gestire un insieme di computer (tipicamente cluster), permettendo agli utenti di richiedere l'utilizzo esclusivo di tali risorse definite in vari modi, come ad esempio "Ho bisogno di 10 nodi, ognuno con 1 GB di memoria, proprio ora" oppure "Ho bisogno di 4 nodi, ognuno con 2 CPUs e 2GB di memoria, dalle ore 2pm alle ore 4pm di domani".
- *Haizea utilizza il leasing*: La risorsa fondamentale che fornisce l'astrazione in Haizea è il leasing. Intuitivamente, a contratto di leasing è come un contratto in cui una parte si impegna a fornire un insieme di risorse (come un appartamento, una macchina, ecc.) ad un'altra parte. Quando un utente vuole richiedere delle risorse computazionali da Haizea, lo fa sotto forma di contratto di leasing.
- *Haizea è VM-based*: Uno dei modi migliori di fornire contratti di leasing delle risorse è quello di utilizzare le macchine virtuali. Gli algoritmi di scheduling di Haizea sono orientati verso la gestione di macchine virtuali, factoring in tutte le operazioni extra richieste nella gestione delle macchine virtuali.
- Haizea è open source: Haizea è pubblicato sotto la licenza Apache 2.0, un BSD-like OSI-compatibile.

6.3.1 Cosa si può fare con Haizea

Haizea può essere utilizzato nei seguenti modi: come componente stand-alone, come uno scheduling backend per un gestore dell'infrastruttura virtuale come OpenNebula, o come “simulatore”.

- *Utilizzando Haizea con OpenNebula*: Haizea può essere usato come un sostituto per il demone di scheduling di OpenNebula. OpenNebula è un gestore di infrastrutture virtuali che consente la distribuzione dinamica e la ri-allocazione delle macchine virtuali su un pool di risorse fisiche. Haizea può essere utilizzato per estendere le capacità di pianificazione (scheduling) di OpenNebula, consentendo di supportare le prenotazioni avanzate delle risorse e le code di richieste best-effort. OpenNebula e Haizea si completano a vicenda, dal momento che OpenNebula fornisce tutti i “muscoli” della messa in atto (OpenNebula può gestire macchine virtuali come Xen e KVM su un cluster, con VMWare come supporto) mentre Haizea fornisce tutti i “cervelli” di scheduling.

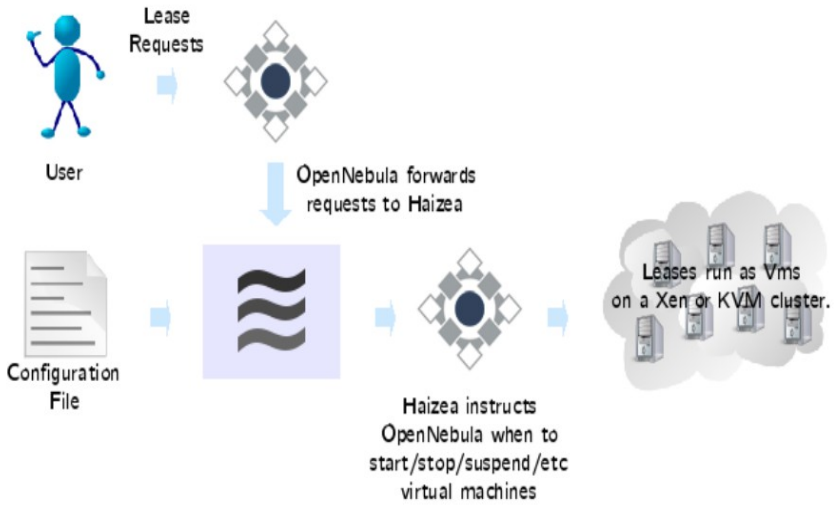

Figura 6.5 Mostra il funzionamento di Haizea con OpenNebula.

- *Utilizzando Haizea da solo*: Haizea è, principalmente, un componente di

gestione delle risorse VM che prendono le richieste di leasing e le decisioni di scheduling basate su queste richieste, ma in realtà non sa nulla di come mettere in atto tali decisioni. Per esempio, Haizea può determinare in quale momento una serie di macchine virtuali che rappresentano un contratto di leasing devono partire o arrestarsi, ma in realtà non sa come istruire una virtual machine manager (come Xen, KVM, ecc.) per fare queste azioni. Haizea può, comunque, delegare queste azioni a un componente esterno utilizzando delle API. Haizea può essere utile se si sta facendo una ricerca di pianificazione che coinvolge contratti di leasing o macchine virtuali.

- Può anche essere utilizzato per simulare l'esecuzione prolungata di carichi di lavoro (come settimane o mesi).

  In questa modalità, Haizea prende una lista di richieste di leasing (specificate in un tracefile) e un file che specifica le opzioni di simulazione e scheduling (come le caratteristiche dell'hardware da simulare), e li processa in un “tempo simulato”.

  In altre parole, l'obiettivo di questa modalità è quello di ottenere la programmazione finale per un set di leasing, senza attendere che tutti i leasing siano completati nel tempo reale (questo rende questa modalità particolarmente utilizzata per trovare che effetto può avere una certa scelta di programmazione in un periodo di settimane o mesi). Il risultato finale di una simulazione è un datafile con informazioni che possono essere utilizzate per creare resoconti (relazioni/rapporti) e grafici, come mostrato in Figura 6.6.

Figura 6.6 Mostra la simulazione di un'esecuzione prolungata di richieste specificate nel tracefile e le opzioni nel configuration file, li processa ottenendo un datafile utilizzabile per statistiche, grafici, ecc.

Haizea può essere utilizzato anche per una simulazione in tempo reale, ciò significa che, invece di dover fornire una lista di richieste di leasing in anticipo, è possibile utilizzare la riga di comando di Haizea per richiedere contratti di leasing in modo interattivo e verificare lo stato dello scheduling di Haizea.

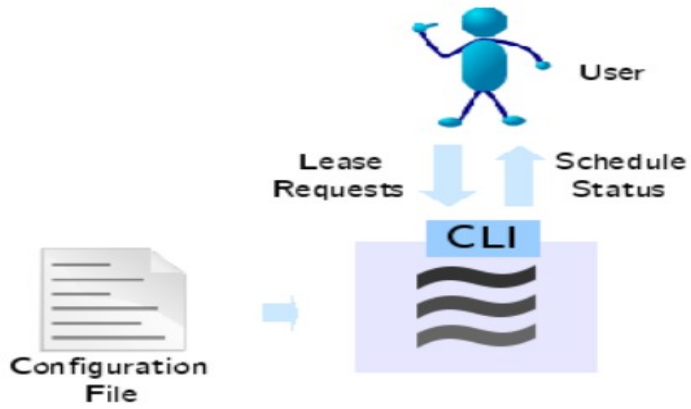

Figura 6.7 Mostra una simulazione in tempo reale.

6.3.2 Architettura

L'architettura di Haizea, mostrata in Figura 6.8, è divisa nei seguenti 3 livelli:

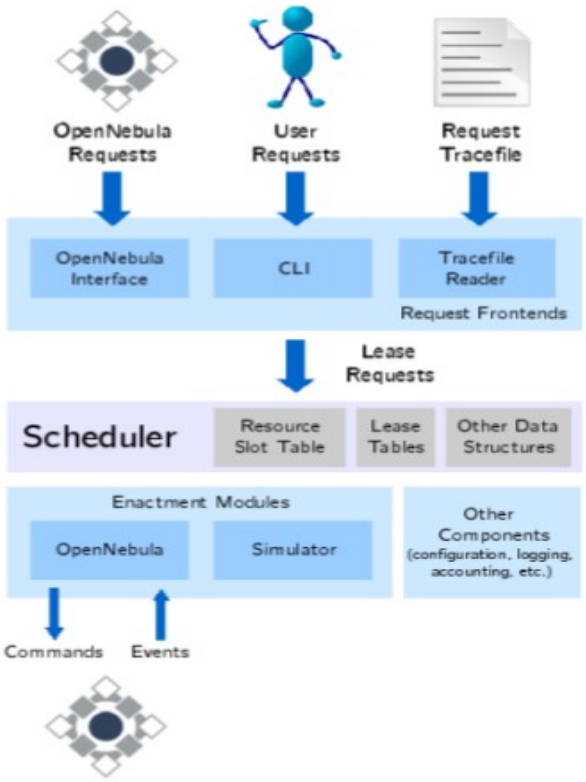

Figura 6.8. Mostra i livelli architetturali di Haizea (Request frontend, Scheduling Core, Enactment Modules)

1) *Request frontend*: è a questo livello che arrivano le richieste di leasing. Haizea può accettare le richieste da OpenNebula, attraverso riga di comando oppure leggendoli da un tracefile (in formato SWF o utilizzando il formato specifico di Haizea LWF).

2) *Scheduling core*: qui è dove le richieste di leasing vengono elaborate e programmate in azioni che si svolgono in specifici momenti (ad esempio, “start VM per leasing X al nodo Y al tempo T”, ecc).

3) *Enactment modules*: vengono promulgate le azioni generate dallo scheduler. Haizea può inviare le azioni a Open- Nebula, essendo così in grado di gestire cluster Xen e KVM, o simulare un cluster.

L'architettura di Haizea mantiene questi tre livelli completamente disaccoppiati, il che significa che aggiungendo il supporto per un enactment backend richiede la sola scrittura di un modulo enactment per tale backend.

6.3.3 Caratteristiche

I termini del leasing devono includere i seguenti grandezze:

- *hardware*: le risorse hardware (CPU, memorie, ecc) richieste dalla risorsa del consumatore.
- *software*: il software che deve essere installato in queste risorse.
- *disponibilità*: il periodo durante il quale le risorse hardware e software devono essere disponibili.

Per esempio, Amazon EC2 è molto bravo a fornire il software ambiente che si desidera, e abbastanza buono a fornire l'hardware, ma non così bravo a supportare una varietà di periodi di disponibilità. Così, Haizea mira a sostenere il leasing delle risorse lungo queste tre dimensioni e per ora supporta i seguenti tipi di disponibilità di leasing:

- quelli richiesti da una singola macchina virtuale o da un gruppo di macchine virtuali che devono essere eseguiti in parallelo.
- quelli che attendono in coda fino a quando le risorse diventano disponibili.
- quelli che devono iniziare in un momento specifico.
- quelli che devono iniziare immediatamente, o per niente.

Gli algoritmi di scheduling di Haizea possono:

- pianificare in modo esplicito l'overhead di macchine virtuali, invece di dedurlo da una assegnazione (stanziamento) dell'utente. Per esempio, se un contratto di leasing deve iniziare alle 2pm, Haizea programmerà il trasferimento delle

immagini necessarie delle macchine virtuali ai nodi fisici in cui le macchine virtuali saranno in esecuzione (e farà in modo che queste arrivino in tempo).

- sfruttare la capacità sospendi/riavvia delle macchine virtuali per sospendere anticipatamente il leasing quando uno con maggiore priorità ha bisogno delle risorse. Sfrutta anche la migrazione “ a freddo” (cold migration) delle macchine virtuali (la migrazione di una macchina virtuale sospesa a una differente macchina per farla riprendere (riattivarla) da quel punto in poi).

- pianificare al meglio le richieste utilizzando il principio di coda “First-Come-First-Serve”.

Haizea può essere utilizzato come programma finale per l'infrastruttura virtuale OpenNebula per fare tutto quanto sopra un cluster Xen o KVM.

6.3.4 Le politiche di Haizea

Haizea utilizza diversi algoritmi interni di scheduling per determinare le risorse da destinare a un leasing. Per la maggior parte, la modifica di questi algoritmi comporta una modifica in profondità nel codice di Haizea. Tuttavia, le diverse decisioni di scheduling che dipendono dalle proprie organizzazioni di politica di allocazione delle risorse sono prese fuori del codice principale di scheduling, nel modulo delle decisioni della politica.
In particolare, le seguenti decisioni sono prese fuori:

- *Lease admission*: in caso di richiesta di leasing, deve essere accettata o respinta? Tenendo conto che questa decisione ha luogo prima che Haizea determini se la richiesta è ancora fattibile o no. Per esempio, un'azienda può richiedere che tutti i contratti di leasing debbano essere richiesti con almeno un'ora di anticipo,

indipendentemente dal fatto che ci siano abbastanza risorse per soddisfare le richieste entro i tempi stabiliti. Tuttavia , il fatto che la richiesta di leasing venga accettata non significa che venga esaudita in quanto può essere che al momento richiesto le risorse non siano disponibili.

- *Lease preemptability*: non tutti i contratti di leasing sono stati creati uguali e, se lo scheduler determina che una richiesta possa essere soddisfatta solo contrastando (precludere) altre richieste, può dover determinare quali contratti di leasing sono i migliori candidati per la precedenza. Per esempio, dato un leasing in esecuzione da una settimana e uno che invece è in esecuzione da cinque minuti, si potrebbe preferire di non interrompere quello con lunga durata.

- *Host selection*: quando lo scheduler ha una vasta scelta tra host fisici su cui distribuire le macchine virtuali, si possono preferire alcuni host rispetto ad altri. Ad esempio, un'azienda potrebbe voler mettere nello stesso host più macchine virtuali, e chiudere quelli dove non vi sono macchine virtuali in esecuzione, mentre un'altra azienda potrebbe voler mettere le macchine virtuali su diversi host, lasciando alcune risorse disponibili in ogni host nel caso in cui le macchine virtuali abbiano bisogno di capacità aggiuntive.

Come si può notare, queste sono tutte decisioni “politiche” che dipendono dall'utilizzatore rispetto agli obiettivi per le proprie risorse. Queste politiche possono essere effettuate senza dover modificare il codice Haizea, semplicemente scrivendo un apposito modulo in Python (che dovrà poi essere “collegato” in Haizea attraverso un file di configurazione).

6.3.5 Tipi di leasing supportati da Haizea

Per illustrare meglio i tipi di leasing supportati da Haizea, assumiamo di avere 4 nodi cluster, e vogliamo in leasing le parti di quel cluster nel corso del tempo. Rappresentiamo i 4 nodi nel corso del tempo come in Figura 6.9:

Figura 6.9 Mostra 4 nodi cluster al passare del tempo.

caso1: il leasing deve iniziare e finire in uno specifico momento. Per esempio, il seguente inizia all'1pm e termina alle 2pm.

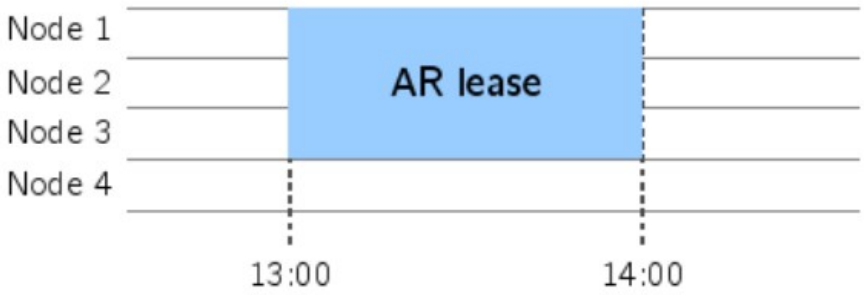

Figura 6.10 Mostra una richiesta di leasing che deve iniziare alle 13:00 e terminare alle 14:00.

caso2: vogliamo delle risorse ma non sappiamo di preciso quando ci serviranno e siamo disposti ad aspettare fino al momento in cui non ci saranno abbastanza risorse disponibili in quel momento.

Figura 6.11 Mostra una richiesta di risorse che deve essere soddisfatta non appena tali risorse saranno disponibili.

Quando si richiede un leasing di questo tipo, la richiesta viene messa in una coda, che verrà elaborata secondo il principio first-come-first-serve. Lo svantaggio di questo tipo di leasing è che potrebbe essere necessario attendere un po' fino a che le risorse ci vengano assegnate.

Figura 6.12 Mostra lo scenario in cui le richieste vengono soddisfatte dopo l'attesa delle risorse.

Inoltre è possibile mettere in pausa il nostro programma consentendo così a quelli con priorità più alta di essere eseguiti.

Figura 6.13 Mostra una risorsa messa in pausa per lasciar posto ad un'altra con maggior priorità.

caso3: caso in cui siamo disposti ad aspettare che le risorse siano disponibili, ma non vogliamo che vengano interrotte (ad esempio se desideriamo utilizzarle in modo interattivo). La richiesta vien fatta attraverso la coda.

caso4: quando abbiamo bisogno delle risorse immediatamente.

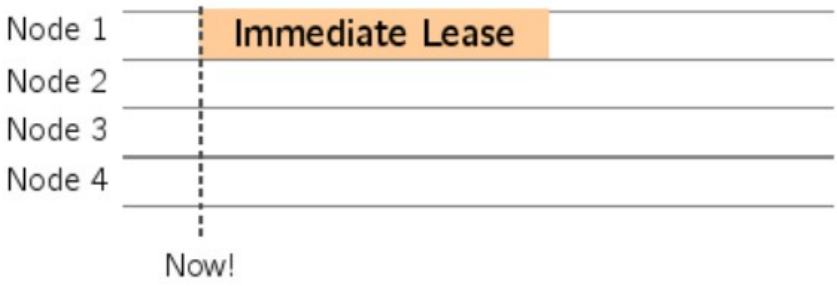

Figura 6.14 Mostra l'immediato bisogno delle risorse.

6.3.6 Haizea in futuro

In futuro Haizea sosterrà più tipi di contratti di leasing, come contratti con scadenze e leasing che richiedono una non banale trattativa prima che il contratto di leasing venga accettato.

Best-effort con scadenze

In alcuni casi, quando si dice “best-effort” significa “un best-effort ragionevole”, cioè siamo pur disposti ad aspettare per ottenere le nostre risorse, ma potremmo averne bisogno prima di una certa scadenza.

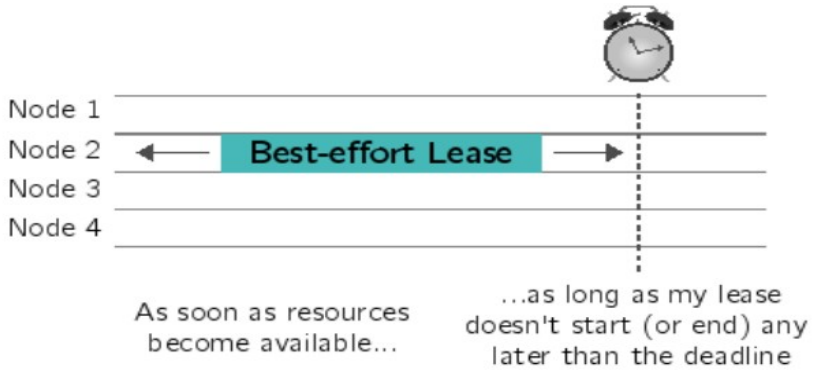

Figura 6.15 Mostra un caso di Best-Effort con scadenza.

Ad esempio, supponiamo di volere 16 nodi del cluster oggi per eseguire un programma di test. Non ci importa particolarmente di quando sarà disponibile il cluster, basta che sia in giornata e che ci sia dato un sufficiente avviso di quando ciò sarà disponibile. In futuro saremo in grado di dire ad Haizea che abbiamo una scadenza, e Haizea sarà in grado di darci le risorse chieste oppure ci dirà che ciò non è possibile.

Leasing negoziabili

In futuro potremo avere delle gestioni di leasing flessibili (esempio, vogliamo una risorsa per le 2pm, Haizea ci proporrà un intervallo di tempo più ampio come “il pomeriggio”) e saremo in grado di negoziare i contratti di leasing con Haizea.

Figura 6.16 Una possibile contrattazione di leasing con Haizea.

Analisi dei dati

Mentre Haizea è in esecuzione, raccoglie dati che possono essere analizzati offline (accettazione/rifiuto richieste leasing, tempi d'attesa, ecc). Questi dati sono salvati su disco quando Haizea finisce di operare, per ora, questa informazione è, in pratica, utile solo per simulare esperimenti. In futuro, Haizea sarà in grado di salvare i dati periodicamente in modo da poterli analizzare online.

6.3.7 Problemi e limitazioni

Qui di seguito elenchiamo alcuni problemi e limitazioni riguardo l'utilizzo congiunto di Haizea con OpenNebula:

- Haizea deve interrogare OpenNebula ogni minuto per chiedere se ci sono delle richieste. Sebbene OpenNebula abbia un meccanismo chiamato “hook mechanism” che consente che le azioni vengano portate a termine quando accadono certi eventi (come quando vengono mandate le notifiche di Haizea di una macchina virtuale fuori uso, oppure un'operazione sospesa è terminata prima del tempo previsto, ecc), Haizea attualmente non utilizza questo tipo di meccanismo.

- Haizea attualmente non può distribuire qualunque immagine con OpenNebula,

e le immagini VM sono presupposte essere predistribuite su nodi fisici, o disponibili su un NFS filesystem condiviso. Anche se OpenNebula include il supporto per l'interfacciamento con un transfer manager per gestire vari scenari di distribuzione di macchine virtuali, Haizea tuttora non accede a questa funzionalità.

- Haizea non può attuare cold migrations in OpenNebula (per esempio, la migrazione di una macchina virtuale sospesa a un differente nodo se le risorse diventano disponibili prima su un nodo diverso da quello in cui la macchina virtuale era stata sospesa). Haizea in realtà ha tutto il codice di programmazione disponibile per far ciò, manca solo la messa in atto vera e propria.

6.4 Amazon AWS

Amazon Web Service (AWS) è uno dei provider di servizi Cloud Computing (IaaS) più famoso per le sue funzionalità di base quali storage e capacità di calcolo on-demand.
I servizi di Amazon si possono dividere in due categorie:

- *Infrastructure web services*, che forniscono risorse fisiche a pagamento;
- *Application web services*, che forniscono applicazioni a supporto dei clienti, ad esempio per gestire i pagamenti in un portale di e-commerce.

6.4.1 Infrastructure Web Services

I servizi di questa categoria sono stati sviluppati principalmente per i software developer, o più in generale, per tutte quelle aziende che hanno la necessità di avere dei sistemi informatici efficienti e scalabili; gli AWS si propongono quindi come alternativa a basso costo, alla costruzione e al mantenimento di grandi data center aziendali.
I principali sono:

- *Simple Storage Service (S3)* fornisce una semplice interfaccia attraverso la quale l'utente può salvare i propri dati sui server di Amazon, avendo poi la possibilità di accedervi in qualsiasi momento via web; permette quindi di inserire dati nella cloud e di poterli recuperare in seguito, con la certezza di trovarli nello stato in cui li si ha inseriti. S3 non è un normale filesystem: prevede solo due livelli gerarchici. Al primo livello ci sono i bucket che possiamo pensare come a delle directory, visto che permettono di organizzare i dati da salvare in S3, ma a differenza delle normali directory, i bucket non possono contenere altri bucket. Lo spazio di nomi in cui vivono i bucket è condiviso da tutti i clienti Amazon; bisogna quindi utilizzare dei nomi che non possono collidere con nomi già in uso. Altro aspetto importante è che S3 è lento e l'accesso ad esso avviene tramite chiamate a dei web service, e non tramite filesystem

o al protocollo WebDAV. Le tariffe non comprendono costi fissi mensili, bensì la quota da pagare viene calcolata in base alla quantità di dati che si mantiene sui server e al traffico (sia upload che download) generato. Nel 2008 a questo servizio si è aggiunto Amazon Elastic Block Storage, un servizio che consente di creare dei veri e propri volumi virtuali (dischi) che possono essere utilizzati per il salvataggio dei dati; questa funzionalità risulta molto utile se integrata con EC2, uno dei più importanti servizi forniti da Amazon.

- *Elastic Compute Cloud (EC2)* il cuore della nuvola di Amazon. Fornisce agli sviluppatori le API necessarie per la creazione, la gestione e il rilascio di server virtuali, la capacità di calcolo on-demand, scalabile secondo le esigenze del singolo cliente; la principale potenzialità di questo servizio è la possibilità di scegliere di volta in volta la potenza necessaria per il calcolo, per questo motivo è stata denominata "elastic". Gli sviluppatori che utilizzano questo servizio devono pagare in base al tempo-macchina occupato, oltre a un corrispettivo simile ad S3 per i dati inviati e ricevuti. Sarà poi il cliente stesso che deciderà come utilizzare il servizio, avendo la possibilità di usufruire di molte risorse per un periodo breve di tempo, o poche risorse per un intervallo più lungo. La capacità di calcolo viene fornita dando all'utente il controllo di una o più macchine virtuali, che possono essere personalizzate secondo le singole esigenze, a partire dal sistema operativo utilizzato, fino ai programmi installati e alle configurazioni di rete dei dispositivi. Supporta i seguenti sistemi operativi: RedHat Linux, Windows Server, openSuSE Linux, Fedora, Debian, OpenSolaris, Cent OS, Gentoo Linux, and Oracle Linux; e supporta i seguenti formati di macchine virtuali: VMware ESX VMDK images, Citrix Xen VHD images, Microsoft Hyper-V VHD images and RAW images.
- *SimpleDB* fornisce agli sviluppatori un database remoto, affidabile ed efficiente; offre la possibilità di effettuare query e lookup su dati strutturati. Il servizio è stato sviluppato per essere in grado di interagire con S3 ed EC2, fornendo ai clienti una piattaforma più completa. Anche in questo caso le tariffe sono calcolate in base al traffico generato e allo spazio occupato. I vantaggi sono: non serve un amministratore di db; le API fornite son molto semplici per utilizzare i web service che effettuano le

query; la disponibilità di un db in un cluster; l'alta scalabilità in termini di quantità di dati memorizzabili.

- *Simple Queue Service (SQS)* permette di inviare messaggi attraverso applicazioni distribuite. Il mittente invia un messaggio alla coda e continua a fare quello che stava facendo. Il destinatario carica quindi il messaggio dalla coda (coda a cui si era registrato in precedenza) e lo processa. Il vantaggio è che il mittente non deve identificare direttamente il destinatario o effettuare controlli degli errori per gestire i problemi di comunicazione o di processamento. Non è nemmeno necessario che il destinatario sia attivo nel momento in cui il mittente invia il messaggio alla coda. Utilizzando questo servizio si rende molto più semplice la gestione del workflow di un'applicazione distribuita, in particolare quelle implementate su istanze di EC2. Anche in questo caso ogni cliente pagherà in base al traffico generato.

Uno dei maggiori punti di forza di Amazon AWS è sicuramente il modello innovativo di fatturazione a consumo, che ha consentito a questi servizi di crescere e di essere utilizzati anche da una parte di clientela nuova, costituita da tutti quegli sviluppatori che non hanno la possibilità o la necessità di firmare contratti annuali con gli hosting provider, come ad esempio molte startup IT.

6.5 Amazon Elastic Cloud Computing EC2

Amazon EC2 presenta un vero e proprio ambiente virtuale che consente di utilizzare le interfacce di servizi web per lanciare le istanze sotto una varietà di sistemi operativi, impiegando per ogni ambiente applicazioni specifiche.
EC2 fornisce agli utenti la possibilità di personalizzazione di tutte le risorse fornite, si possono decidere infatti non solo la quantità delle macchine virtuali da attivare, ma anche quale tipo di immagine utilizzare e decidere dove questa dovrà essere allocata, così che nuove istanze del server possano essere installate e avviate in pochi minuti, e la loro capacità possa essere scalata rapidamente attraverso una semplice interfaccia web service. Inoltre si possono definire tutte una serie di parametri legati alla gestione della sicurezza delle comunicazioni tra le diverse istanze attive. Amazon infatti dispone di data center dislocati in diverse zone del mondo, dando così la possibilità agli utenti di scegliere la collocazione migliore per le loro istanze.

Per utilizzare Amazon EC2, basta semplicemente:

- creare un Amazon Machine Image (AMI), che contenga le applicazioni, le librerie, i dati e le impostazioni di configurazione, o in alternativa, caricare una macchina virtuale già pre-configurata per ottenere una AMI che sia immediatamente funzionante.
- caricare l'AMI in Amazon S3. EC2 fornisce gli strumenti che rendono semplice la memorizzazione della stessa. Amazon S3 fornisce un sicuro, affidabile e veloce "magazzino" per memorizzare le "immagini". Utilizzare il servizio web di Amazon EC2 per la configurazione della sicurezza e l'accesso alla rete.
- Scegliere il tipo di istanze da utilizzare, il sistema operativo che si desidera, poi farle eseguire (lancio istanza – termino istanza); controllare il numero di istanze dell'AMI utilizzando, se necessario, il servizio Web API o altri strumenti di gestione.
- Stabilire se si desidera eseguire le istanze in locazioni multiple (IP statico

endpoints, o l’attach persistent block storage).

- Pagare solo le risorse che si consumano effettivamente, come ad esempio, il tempo o il trasferimento di dati.

Amazon EC2 offre ambienti virtuali basati su macchine di calcolo ed utilizza l'hypervisor Xen per gestire la propria istanza Amazon Machine Image (AMI). AMI è "un'immagine macchina crittografata che contiene tutte le informazioni necessarie per avviare le istanze del software". Utilizzando semplici interfacce di servizi Web, gli utenti possono avviare, eseguire, monitorare e terminare le loro istanze, come mostrato in Figura 6.17. Inoltre possono, aggiungere al volo qualunque delle caratteristiche di cui sopra per la loro configurazione come desiderano.

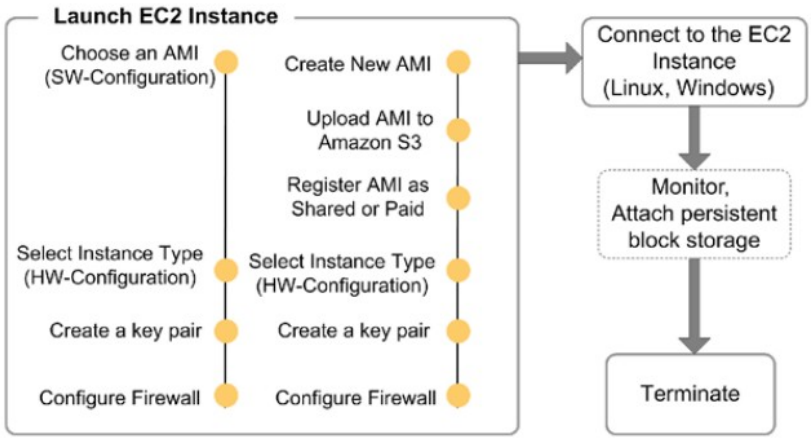

Figura 6.17. Lifecycle of Amazon machine image.

6.5.1 Come scegliere il tipo di istanze

L'utente deve decidere quale tipo di istanza vuole avviare per eseguire le proprie applicazioni; il tipo di istanza definisce la quantità di risorse che Amazon allocherà per la macchina virtuale, che determina anche la tariffa oraria che verrà fatturata all’utente. Amazon non permette di impostare direttamente i parametri fisici delle istanze, come la quantità di memoria o lo spazio su disco, ma fornisce una vasta gamma di modelli

tra cui scegliere.

I tipi di macchine proposti differiscono principalmente su quattro parametri: numero di CPU, quantità di memoria, spazio su disco e prestazioni di I/O.

Per poter adattarsi alle varie tipologie di applicazioni che si prestano ad essere eseguite da EC2, vengono proposte diverse tipologie di macchine divise per “famiglie” di applicazioni: standard, micro, High-CPU, High-Memory, Cluster Compute.

Type	CPU	Memory	Local Storage	Platform	I/O	Name
Small	1 EC2 Compute Unit (1 virtual core with 1 EC2 Compute Unit)	1.7 GB	160 GB instance storage (150 GB plus 10 GB root partition)	32-bit	Moderate	m1.small
Large	4 EC2 Compute Units (2 virtual cores with 2 EC2 Compute Units each)	7.5 GB	850 GB instance storage (2 x 420 GB plus 10 GB root partition)	64-bit	High	m1.large
Extra Large	8 EC2 Compute Units (4 virtual cores with 2 EC2 Compute Units each)	15 GB	1690 GB instance storage (4 x 420 GB plus 10 GB root partition)	64-bit	High	m1.xlarge
Micro	Up to 2 EC2 Compute Units (for short periodic bursts)	613 MB	None (use Amazon EBS volumes for storage)	32-bit or 64-bit	Low	t1.micro
High-CPU Medium	5 EC2 Compute Units (2 virtual cores with 2.5 EC2 Compute Units each)	1.7 GB	350 GB instance storage (340 GB plus 10 GB root partition)	32-bit	Moderate	c1.medium
High-CPU Extra Large	20 EC2 Compute Units (8 virtual cores with 2.5 EC2 Compute Units each)	7 GB	1690 GB instance storage (4 x 420 GB plus 10 GB root partition)	64-bit	High	c1.xlarge
High-Memory Extra Large	6.5 EC2 Compute Units (2 virtual cores with 3.25 EC2 Compute Units each)	17.1 GB	420 GB instance storage (1 x 420 GB)	64-bit	Moderate	m2.xlarge
High-Memory Double Extra Large	13 EC2 Compute Units (4 virtual cores with 3.25 EC2 Compute Units each)	34.2 GB	850 GB instance storage (1 x 840 GB plus 10 GB root partition)	64-bit	High	m2.2xlarge
High-Memory Quadruple Extra Large	26 EC2 Compute Units (8 virtual cores with 3.25 EC2 Compute Units each)	68.4 GB	1690 GB instance storage (2 x 840 GB plus 10 GB root partition)	64-bit	High	m2.4xlarge
Cluster Compute	33.5 EC2 Compute Units (2 x Intel Xeon X5570, quad-core "Nehalem" architecture)	23 GB	1690 GB instance 64-bit storage (2 x 840 GB plus 10 GB root partition)	64-bit	Very high (10 Gbps Ethernet)	cc1.4xlarge

Tabella 6.18 Confronto tra le diverse tipologie di macchine divise per famiglie di applicazioni.

Ognuna di esse è dedicata a un determinato tipo di applicazione, ad esempio le macchine della famiglia High-CPU sono dotate di una grande potenza di calcolo, dedicata appunto a tutte le applicazioni CPU-intensive, che magari richiedono una

quantità di ram non troppo elevata; in questo modo Amazon EC2 permette ai suoi utenti di pagare solamente per le risorse che gli sono effettivamente necessarie, senza dover spendere ulteriore denaro per altre non utilizzate.
La scelta della corretta configurazione della macchine risulta fondamentale, soprattutto nel caso in cui si debba utilizzare un cluster costituito da un numero di istanze abbastanza grande; effettuando una scelta appropriata si può risparmiare una quantità considerevole di denaro, senza dover sacrificare le prestazioni della propria applicazione.

Bisogna sottolineare anche che non è necessario fare una scelta cercando di mantenere una certa percentuale di risorse libere per eventuali utilizzi futuri, come ad esempio si è soliti fare quando si acquista una macchina fisica. Se le prestazioni di un'istanza diventassero insufficienti, basterà avviare una nuova macchina, utilizzando un modello con un maggior numero di risorse, e sostituirla a quella avviata precedentemente.
Questo tipo di soluzione è reso possibile dal fatto che per avviare una nuova istanza è sufficiente inviare una richiesta su server di EC2, che in pochi minuti la renderanno attiva e funzionante, cosa impensabile se si utilizzasse un servizio di hosting tradizionale, in cui è necessario fare delle richieste specifiche, che possono essere eseguite con tempi dell'ordine di qualche giorno.

6.5.2 Come scegliere quale AMI utilizzare

Come scegliere quale immagine utilizzare per l'esecuzione della macchina virtuale?
Una Amazon Machine Image contiene tutte le informazioni che servono ad una virtual machine per essere avviata, come il sistema operativo e tutto il software necessario all'esecuzione dell'applicazione scelta. Amazon rende disponibili molte immagini pronte all'uso (su cui sono installati i sistemi operativi, come Linux e Windows; o dotate di programmi preinstallati pronti all'uso, come server MySQL, Oracle, o web server apache).
Gli utenti possono scegliere se utilizzare tali immagini aggiungendo semplicemente i

propri dati, oppure modificarle, creando delle proprie AMI personalizzate; tali immagini possono anche essere rese pubbliche, consentendo così ad altri utenti di utilizzarle per le loro attività. Qualora le immagini offerte nel catalogo di Amazon non fossero adatte alle applicazioni che si vogliono utilizzare, è possibile creare delle proprie immagini a partire da host reali o da immagini di macchine virtuali, che poi possono essere caricate sui server Amazon.
Nel caso in cui si debbano eseguire delle immagini su cui è installato del software protetto da copyright (come nel caso di Windows), si possono utilizzare delle AMI a pagamento orario maggiorato, in modo da poter corrispondere i diritti della casa produttrice.

6.5.3 Modalità d'esecuzione delle macchine virtuali

Per quanto riguarda la gestione della persistenza dei dati delle immagini in esecuzione si può scegliere di eseguire le proprie macchine virtuali in due modalità differenti: EBS-backed e S3-backed.

EBS-Backed
Un’istanza EBS-backed utilizza un disco di Elastic Block Storage come partizione di root.
Il funzionamento di questo tipo di istanze è molto simile a quello che si verifica quando si utilizza una macchina virtuale su un normale PC desktop, infatti tutti i dati dell’immagine della macchina sono salvati su un supporto persistente, di conseguenza non vi è nessun pericolo di perdita di dati dopo lo spegnimento della macchina virtuale. Le principali caratteristiche delle istanze di questa tipologia sono:

- *Allocare una maggior quantità di spazio* per la partizione di root del sistema, limitata a 10GB per le istanze S3-backed; utilizzando un EBS invece si può montare una partizione fino a 1TB di capacità.
- *Maggior velocità in fase di boot*, visto che i dati del disco non devono essere completamente copiati prima di avviare l’istanza, ma basta accedere ai file necessari

per il boot della macchina. In fase di runtime questo tipo di istanza risulta meno performante, in quanto i dischi volatili delle altre istanze sono creati localmente sulla macchina host, mentre i volumi EBS vengono acceduti tramite connessioni a servizi esterni.

- *Possibilità di spegnere un'istanza* e riavviarla in un secondo momento, mantenendo lo stato del sistema e senza dover pagare le ore-macchina quando questa è spenta.
- *Creazione delle EBS-backed AMI* tramite delle API a partire da istanze in esecuzione, il processo risulta molto semplice, in quando la procedura consiste semplicemente nel creare un nuovo volume EBS, associarlo ad una configurazione di boot e registrarla.

S3 Backed

Le istanze lanciate da immagini S3-backed sono in realtà una copia dell'immagine che si è scelto di utilizzare: il sistema infatti, quando riceve una richiesta di avvio per una nuova istanza, crea una copia di tutti i dati necessari e li salva su un disco locale (virtuale) che verrà utilizzato come partizione di root.

Da questo momento in poi, tutte le modifiche che verranno effettuate sull'immagine dell'istanza in esecuzione, come modifica dei dati, delle impostazioni del software ecc. saranno solo memorizzate localmente sul disco virtuale associato alla macchina. Nessuna modifica può essere fatta sull'immagine originale, che ha solamente il ruolo di stato di partenza, da cui vengono copiati i dati. Quando viene spenta un'istanza, tutti i dati contenuti sul disco vengono eliminati, così come lo stato del sistema; per mantenere in memoria i dati prodotti da una certa istanza si devono utilizzare tecniche di salvataggio dei dati diverse, come ad esempio collegare al sistema un disco di EBS, o utilizzare S3.

EBS vs S3

La differenza sostanziale tra queste due modalità riguarda come vengono gestiti i dati prodotti durante l'esecuzione dell'istanza: in un caso questi restano memorizzati su un

disco di EBS, mentre nell'altro vengono eliminati una volta che la macchina virtuale ha terminato la propria esecuzione.
Il vantaggio maggiore delle istanze EBS rispetto a quelle S3-backed è sicuramente il poter essere sospese e poi riattivate in momenti successivi, senza perdere lo stato del sistema; macchine di questo tipo si comportano, di fatto, come macchine reali.

A livello di prestazioni vi è una differenza sensibile tra l'accesso ai dati su un EBS e su un disco locale: al primo infatti vi si accede in maniera remota all'interno della rete di Amazon, quindi le operazioni di I/O effettuate su dischi EBS sono meno veloci rispetto a quelle effettuate su dischi locali. Conviene perciò utilizzare soluzioni S3-backed se si richiede che gli accessi su disco siano molto veloci e frequenti.

Parlando di costi, utilizzare un'istanza di tipo S3-backed richiede solo di pagare il tempo effettivo nel quale è stata eseguita la macchina virtuale, senza nessun costo aggiuntivo per gli accessi sul disco locale. Se invece si utilizza un'istanza EBS-backed si devono considerare anche i costi dello storage dei dati su EBS, oltre che agli accessi effettuati durante l'esecuzione dell'istanza.

Per quanto riguarda la sicurezza dei dati, se si richiede di avviare una serie di istanze identiche, per poterle poi raggruppare in un cluster facendole collaborare tra loro, si può trarre vantaggio dal sistema proposto da Amazon, cioè da una sola immagine "base" si possono attivare quante macchine virtuali si desidera. Uno dei vantaggi associato a questo tipo di implementazione riguarda la possibilità di separare il software, ossia il sistema operativo, con annesse applicazioni e impostazioni, dai dati utilizzati e prodotti dal software in esecuzione sulle istanze. Inoltre utilizzare un'organizzazione sistemistica del tipo appena descritto consente di ottenere dei livelli di fault-tolerance e di disaster recovery più alti, infatti se il volume EBS associato ad un'istanza EBS-backed dovesse corrompersi, tutti i dati dell'immagine verrebbero persi, mentre se si dovessero avere problemi con i dischi locali di un'istanza S3-backed si richiederebbe solamente un riavvio della macchina virtuale per rendere tutto

di nuovo operativo.

6.5.4 Networking con la rete EC2

Ad ogni istanza EC2 vengono assegnati, in fase di avvio, due indirizzi IP, uno privato e uno pubblico, collegati tra loro tramite un sistema di NAT; gli indirizzi privati sono raggiungibili solamente dall'interno della rete Amazon, mentre quelli pubblici sono accessibili dalla rete Internet.

6.5.5 Politiche di sicurezza e security groups

Per gestire le policy di sicurezza della rete tra le istanze, si possono assegnare le macchine virtuali a dei security groups definiti dall'utente, il quale può personalizzare le regole del firewall da applicare in ogni gruppo. Tali regole non sono relative ad una singola istanza, ma vengono applicate all'intero insieme di istanze: si facilitano così tutte le operazioni di modifica e aggiunta di nuove regole, che verranno automaticamente applicate a tutte le istanze appartenenti a quel security group. Le regole che si possono definire in un security group sono molto simili a quelle disponibili per l'applicazione iptables nei sistemi Linux. La divisione in gruppi delle istanze consente inoltre di creare degli insiemi di macchine virtuali totalmente isolati tra loro, garantendo quindi un ottimo livello di sicurezza.

6.5.6 Chiavi di accesso

Prima di avviare un'istanza, EC2 può modificare l'immagine originale per inserire una parte di una coppia di chiavi, che consentirà poi all'utente di accedere alla macchina virtuale tramite ssh. Ogni utente può definire diverse keypair, ad ognuna delle quali si deve assegnare un nome, che deve essere passato come parametro durante la richiesta di startup di una nuova istanza. L'utilizzo di questo sistema di autenticazione permette di evitare di accedere alle macchine tramite password, metodo sconsigliato in modo da massimizzare il livello di sicurezza degli accessi; inoltre in questo modo si possono

utilizzare tranquillamente immagini pubbliche, senza correre nessun rischio di intrusioni non consentite, infatti in tale tipologia di immagini l'accesso via password è disabilitato, di conseguenza si potrà accedere alla macchina solamente utilizzando una delle coppie di chiavi che l'utente stesso ha definito.

6.5.7 Storage

I server fisici che Amazon utilizza per fornire i propri servizi di Cloud Computing sono basati su una versione altamente personalizzata dell'hypervisor open-source Xen che utilizza la para-virtualizzazione.
L'ambiente Xen permette la creazione e il rilascio dinamico di server virtuali e tutte le altre funzionalità necessarie per offrire ambienti di lavoro completamente isolati l'uno dall'altro.
EC2 prevede due modalità di storage:

- *temporaneo* (ephemeral storage), legato ad ogni nodo e che quindi sparisce insieme all'istanza;
- *persistente* nel tempo (storage a blocchi).

Quest'ultima è tra le principali problematiche che si devono risolvere in un sistema EC2 vi è la persistenza dei dati, che deve essere gestita in maniera appropriata, utilizzando i vari strumenti offerti da Amazon.

Il mantenimento della persistenza dei dati in sistemi IaaS strutturati, come Amazon EC2, deve essere gestito in maniera particolare, dato che di default le istanze sono semplicemente delle copie delle immagini originarie, e tutti i dati che producono vengono persi una volta che terminano la loro esecuzione. Per salvare i propri dati, gli utenti devono utilizzare i servizi complementari di Amazon, o altri servizi analoghi se si utilizzano altri fornitori IaaS; si può utilizzare S3, che consente di mantenere i propri dati in una sorta di repository remoto (permettendo così di ottimizzare l'efficienza di accesso ai dati), oppure utilizzare EBS, salvando i propri dati su volumi di storage che

devono essere collegati alle istanze in esecuzione. La scelta di questo modello di gestione dei dati può sembrare troppo complesso ad una prima analisi, così facendo però è possibile disaccoppiare completamente la macchina dai dati che essa elabora, consentendo in questo modo di creare dei cluster largamente scalabili, gestibili in maniera più semplice e veloce.
Supponendo che si voglia creare un cluster di web server, nel momento in cui si volesse aggiungere una nuova macchina affinché lavori in parallelo alle altre, è sufficiente avviare una nuova istanza EC2 con gli stessi parametri degli altri web server, senza dover clonare l'intera macchina come invece si richiede nei sistemi gestiti in maniera tradizionale. Si ha anche la possibilità di non sapere a priori di quante macchine sarà composto un cluster, infatti si può avviare la stessa immagine di una macchina più volte, senza doverne creare di nuove ogni volta.

Il modello architetturale proposto da Amazon, costituito da istanze dotate di storage volatile, consente anche di facilitare le operazioni di manutenzione e di aggiornamento dei sistemi installati, grazie alla separazione che vi è tra il software, i dati e la macchina. La separazione tra macchina e software permette di poter effettuare degli "upgrade" hardware delle istanze semplicemente riavviandole scegliendo una tipologia di istanza diversa, dotata di un maggior numero di risorse; per eseguire questa operazione infatti non è necessario modificare in alcun modo l'immagine utilizzata, le impostazioni del sistema operativo o del software installato.
Avere la possibilità di effettuare rapidi aggiornamenti delle risorse allocate permette di dimensionare le proprie macchine per utilizzarle al massimo evitando comunque fenomeni di underprovisioning, che possono essere risolti riavviando l'istanza con una quantità maggiore di risorse.
Ogni utente può pagare solamente per le risorse di cui effettivamente ha bisogno e, qualora fosse necessario, ne può richiedere di aggiuntive nel tempo di un riavvio. La separazione tra software e dati permette di effettuare operazioni di manutenzione e aggiornamento sul software dell'istanza, senza doversi preoccupare della gestione dei dati e del mantenimento della disponibilità del servizio. Le operazioni di

aggiornamento possono essere fatte su un’istanza avviata separatamente rispetto a quelle utilizzate per fornire il servizio, che verrà poi salvata e memorizzata nel repository di immagini; per rendere effettivi gli aggiornamenti basta anche in questo caso avviare una nuova istanza per sostituire quella obsoleta, gestendo la migrazione dei dati semplicemente spostando il volume EBS sul quale sono memorizzati da un’istanza all’altra, oppure sincronizzandoli con il bucket S3 che le contiene.

6.5.8 Soluzioni

Una possibile soluzione per mantenere la persistenza dei dati è quella di utilizzare un volume EBS e di salvare su di esso i dati, ad esempio montandolo al posto della cartella in cui sono contenute le tabelle di un database relazionale.
Questo sistema è molto semplice da implementare, anche se risulta poco adatto nel caso in cui si volessero utilizzare più istanze della stessa applicazione, per rendere il sistema più tollerante ai guasti; infatti ogni macchina dovrà essere dotata di un volume diverso dalle altre, il che comporta un aumento della difficoltà nella sincronizzazione dei dati.
Un'altra tecnica che può essere utilizzata sfrutta la possibilità di creare degli snapshot dei volumi EBS e di poter creare dei nuovi dischi a partire da queste immagini. In questo modo si può utilizzare uno snapshot “master” dai quali si andranno a creare nuovi volumi ogni volta che una macchina verrà avviata.
Questo approccio consente di separare il sistema dell’istanza dall’applicazione e dai dati che essa elabora, anche se è una soluzione poco funzionale, soprattutto perché si devono ogni volta creare nuovi volumi a partire da uno o più snapshot. Utilizzare i volumi per la persistenza dei dati risulta comunque molto valido per applicazioni quali i database.
Si può infine utilizzare il servizio S3 di Amazon per mantenere un repository aggiornato di tutti i dati che le nuove istanze dovranno utilizzare, basta configurarle in modo che dopo l’avvio contattino S3 per ricevere i dati necessari. Questo sistema è molto utile per la gestione di tutti i file di configurazione delle applicazioni, in quanto

basta aggiornare il file su S3 per renderlo utilizzabile immediatamente.

6.5.9 Punti di forza di EC2

- *Elasticità*: Amazon EC2 consente di aumentare o diminuire la capacità di calcolo nel giro di pochi minuti, non in ore o in giorni. È possibile utilizzare centinaia o addirittura migliaia di server contemporaneamente. Questo perché tutto è sotto controllo mediante le API del servizio Web. Le applicazione degli utenti si possono scalare in maniera del tutto automatica, sia verso l'alto che verso il basso, a seconda delle esigenze.
- *Controllo*: l'utente ha il completo controllo delle sue istanze. Ha accesso come root per ciascuna di esse, e può interagire con loro come si farebbe con qualsiasi macchina. Le istanze possono essere riavviate in remoto usando le API del servizio web. Si può anche ottenere un accesso alla console di uscita delle proprie istanze.
- *Flessibilità*: l'utente può scegliere tra varie tipologie di istanze, ha a disposizione vari sistemi operativi e pacchetti software. Amazon EC2 consente di selezionare una configurazione di memoria, di CPU, e della migliore istanza di archiviazione possibile, in funzione della scelta del sistema operativo e delle applicazioni. Ad esempio, la scelta di sistemi operativi comprende numerose distribuzioni di Linux, di Microsoft Windows Server e di OpenSolaris.
- *Integrazione*: Amazon EC2 è progettato anche per l'utilizzo con altri Amazon Web Services. Esso, infatti, lavora in collaborazione con Amazon Simple Storage Service (Amazon S3), Amazon SimpleDB e Amazon Simple Queue Service (Amazon SQS) per fornire una soluzione completa per il calcolo, l'elaborazione e l'archiviazione di ricerca in una vasta gamma di applicazioni.
- *Affidabilità*: Amazon EC2 è altamente affidabile ed offre un ambiente in cui le istanze possono essere comodamente previste, quindi, facilmente commissionate, ma anche rapidamente sostituite (database di istanze pre-configurate). I vari servizi vengono eseguiti all'interno dell'infrastruttura di rete e dei data center di Amazon.

7 Conclusioni e sviluppi futuri

In questa tesi sono state presentate le caratteristiche e le funzionalità dei servizi di Cloud Computing evidenziando quali vantaggi si possono ottenere grazie al suo utilizzo e quali problematiche si devono affrontare.
Nella prima parte si è visto cos'è il Cloud Computing, in particolare la sua architettura, concentrando l'attenzione sulle principali caratteristiche dei tre livelli SaaS, PaaS e IaaS, e sulla tecnica di virtualizzazione che contribuisce in maniera considerevole al funzionamento del Cloud Computing.
Nella seconda parte sono stati analizzati i prodotti open source Haizea, OpenNebula e Amazon EC2, descrivendone l'architettura, il funzionamento, i vantaggi e gli svantaggi.

I servizi di Cloud Computing nei prossimi anni svolgeranno sicuramente un ruolo importante nel settore dell'Information Technology, sia per quanto riguarda le offerte Software-as-a-Service, grazie alla sempre più avanzata tecnologia dei client e alla disponibilità di collegamenti veloci, sia per i servizi Infrastructure-as-a-Service, che risultano molto interessanti per le aziende e gli sviluppatori di applicazioni.
Le risorse cloud di tipo infrastrutturale possono essere vantaggiose dal punto di vista economico per molte attività, in particolare per tutte quelle che operano in settori strettamente collegati al Web. I principali vantaggi che un'azienda può trarre dall'utilizzo di servizi cloud sono una riduzione dei costi operativi e un annullamento degli investimenti iniziali, ma anche la garanzia di dover pagare esattamente in base alla quantità di risorse effettivamente utilizzate. Oltre ai vantaggi economici, vi sono anche delle agevolazioni dal punto di vista del mantenimento dei sistemi, infatti tutta la gestione "fisica", con le problematiche del caso, viene delegata a terzi, come la collocazione, l'alimentazione e la manutenzione delle macchine.

Un altro aspetto importante è costituito dalla possibile interazione tra i servizi cloud pubblici, forniti dai service provider, e le risorse private presenti all'interno dei data center aziendali. Questo nuovo sistema di integrazione viene detto Hybrid Cloud .
Così come crescerà l'interesse nelle cloud private e ibride IaaS, così ci sarà la necessità di un diverso ecosistema di strumenti e tecnologie che potranno essere utilizzati come blocchi per creare e gestire le cloud. Anche se alcune soluzioni sono emerse in tre grandi categorie - cloud, VI, e gestione VM - la sfida del futuro si occuperà di integrare più componenti per creare soluzioni complete per la costruzione di cloud IaaS.
Cloud private e ibride dovranno affrontare anche la sfida di gestire in modo efficiente risorse limitate. Tuttavia, gli attuali gestori VI contano su una fornitura delle risorse immediate che assume implicitamente che la capacità è praticamente infinita. Anche se questo è ragionevole per i fornitori di cloud grandi dimensioni, come Amazon EC2, non è applicabile ai piccoli provider, dove la probabilità di sovraccarico è maggiore.
OpenNebula e Haizea affrontano queste due sfide. Facendo affidamento su un'architettura flessibile, aperta, OpenNebula è stato progettato fin dall'inizio per essere facile da integrare con altri componenti, come il gestore di leasing Haizea. Quando vengono utilizzati insieme, OpenNebula e Haizea sono la soluzione di gestione VI che fornisce funzionalità di fornitura di leasing immediati, tra cui contratti di leasing best-effort e prenotazione anticipata di capacità.

Una delle problematiche che emerge quando si sceglie di utilizzare i servizi cloud riguarda le garanzie che vengono rilasciate sulla disponibilità di servizio. Il Cloud Computing ha ancora qualche problema per quanto riguarda l'affidabilità del servizio 24h su 24h, nonostante vengano assicurate percentuali di uptime molto elevate, i cloud provider purtroppo non sono ancora in grado di garantire una disponibilità totale. Ci son stati casi in cui i servizi di Cloud Computing hanno subito un'interruzione anche se di poche ore. Per questo motivo i servizi IaaS non possono essere utilizzati per applicazioni che non ammettono periodi di downtime, come quelle che operano in

ambito governativo o finanziario. È da sottolineare però che nella maggior parte dei casi le garanzie offerte dai cloud provider soddisfano pienamente le richieste degli sviluppatori: di solito viene assicurato un uptime del 99.9% del servizio durante il corso dell'anno, riuscendo così a rendere i servizi IaaS un'ottima alternativa a soluzioni di hosting tradizionali.
In futuro ci saranno sempre più fornitori di Cloud Computing, servizi più ricchi, norme e standard stabiliti.
Un futuro scenario di sviluppo potrà essere che l'impresa potrà utilizzare una cloud ibrida distribuita come illustrato in Figura 7.

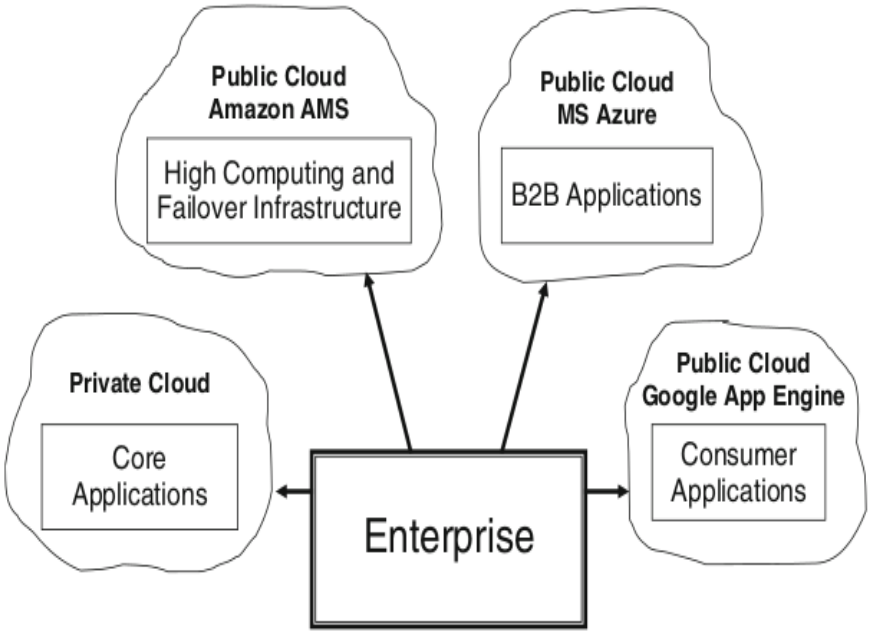

Figura 7. Mostra un'ipotetica situazione in cui un'azienda utilizza una cloud ibrida

Secondo questo scenario, l'impresa utilizzerà le applicazioni di base sulla sua cloud privata, mentre alcune altre applicazioni saranno distribuite su diverse cloud private, che saranno ottimizzate per specifiche applicazioni.

Infine un altro aspetto che devono considerare gli utenti è rappresentato dai livelli di sicurezza garantiti dal provider, che devono essere verificati e confrontati con le necessità dell'applicazione; oltre a questa valutazione tecnica si deve anche effettuare

un’analisi di come venga percepito dai propri clienti il fatto di affidare a terzi l’esecuzione delle applicazioni e i dati ad esse relativi .

Ad oggi non è ancora stato definito uno standard condiviso per l’utilizzo dei servizi di Cloud Computing, che favorirebbe sicuramente l’adozione di questo tipo di soluzioni da parte di potenziali clienti. Si ridurrebbero infatti gli effetti di lock-in, causati dalla complicazione delle procedure di migrazione da un provider all’altro.

Bibliografia

- “Cloud Application Architectures” di George Reese, 2010
- “Cloud Computing: A Practical Approach” di Anthony T.Velte, Toby J.Velte e Robert Elsenpeter, 2009
- “Handbook of Cloud Computing” di Borko Furht e Armando Escalante, 2010
- “Virtual Infrastructure Management in Private and Hybrid Clouds” di Borja Sotomayor, Rubén S.Montero, Ignacio M.Llorente e Ian Foster, 2009
- “Cloud Computing: Principles and Paradigms” di Rajkumar Buyya, James Broberg e Andrzej Goscinski, 2011
- “Algorithms and Architectures for Parallel Processing” di Yang Xiang, Alfredo Cuzzocrea, Michael Hobbs, Wanlei Zhou, 2011
- “Cloud Computing with Linux” di M.Tim Jones, 2012
- “Above the Clouds: A Berkeley View of Cloud Computing” di Michael Armbrust, 2009.
- “Cost Model for Planning, Development and Operation of a Data Center” di D. Patel Chandrakant e Shah Amip J, 2005.
- “Quantitative comparison of Xen and KVM” di T. Deshane, 2008.
- “The business of clouds” di Guy Rosen, 2010.
- “Resource Leasing and the Art of Suspending Virtual Machines ” di Borja Sotomayor , Rubén Santiago Montero, Ignacio Martìn Llorente e Ian Foster, 2009
- “Capacity Leasing in Cloud Systems using the OpenNebula Engine ” di Borja Sotomayor , Rubén Santiago Montero, Ignacio Martìn Llorente e Ian Foster, 2009
- Amazon Web Services. [Online]. http://aws.amazon.com/
- IBM Cloud. [Online]. http://www.ibm.com/ibm/cloud/
- Haizea [Online]. http://haizea.cs.uchicago.edu/
- OpenNebula [Online]. http://www.opennebula.org/

- Xen [Online]. http://xen.org/
- KVM [Online]. http://www.linux-kvm.org/page/Main_Page
- VMWare [Online]. http://www.vmware.com/it/
- Persistence Strategies for Amazon EC2. [Online]. Gerardo Viedema. (2010, Luglio) http://www.theserverlabs.com/blog/2010/07/08/ec2-persistence-strategies/