

# Rapporti tecnici INGV

Una infrastruttura di virtualizzazione  
e primi passi per un *private cloud*  
per l'INGV

# 231



## **Direttore**

Enzo Boschi

## **Editorial Board**

Raffaele Azzaro (CT)

Sara Barsotti (PI)

Mario Castellano (NA)

Viviana Castelli (BO)

Rosa Anna Corsaro (CT)

Luigi Cucci (RM1)

Mauro Di Vito (NA)

Marcello Liotta (PA)

Simona Masina (BO)

Mario Mattia (CT)

Nicola Pagliuca (RM1)

Umberto Sciacca (RM1)

Salvatore Stramondo (CNT)

Andrea Tertulliani - Editor in Chief (RM1)

Aldo Winkler (RM2)

Gaetano Zonno (MI)

## **Segreteria di Redazione**

Francesca Di Stefano - coordinatore

Tel. +39 06 51860068

Fax +39 06 36915617

Rossella Celi

Tel. +39 06 51860055

Fax +39 06 36915617

[redazionecen@ingv.it](mailto:redazionecen@ingv.it)



# Rapporti tecnici INGV

**UNA INFRASTRUTTURA DI VIRTUALIZZAZIONE E PRIMI  
PASSI PER UN *PRIVATE CLOUD* PER L'INGV**

Stefano Chiappini

INGV (Istituto Nazionale di Geofisica e Vulcanologia, Sezione Geomagnetismo, Aeronomia e Geofisica Ambientale)

# 231



## Indice

Introduzione	5
1. La strada verso la virtualizzazione	5
2. La virtualizzazione: che cosa è e come funziona	6
2.1 Concetti generali	6
2.2 Tipi di virtualizzazione: <i>hosted e bare metal</i>	7
2.3 Come funziona la virtualizzazione	8
2.3.1 CPU	8
2.3.2 Memoria	9
2.3.3 I/O	11
2.4 Principali vantaggi: <i>High Availability e live motion</i>	11
3. Il <i>private cloud</i>	12
3.1 La classificazione dei servizi	13
3.2 Dalla virtualizzazione al <i>cloud</i>	14
4. L'infrastruttura realizzata a Roma2	14
5. Applicazioni virtualizzate	18
5.1 Analisi del risparmio	19
6. Risorse disponibili e sviluppi futuri	21
Bibliografia	23



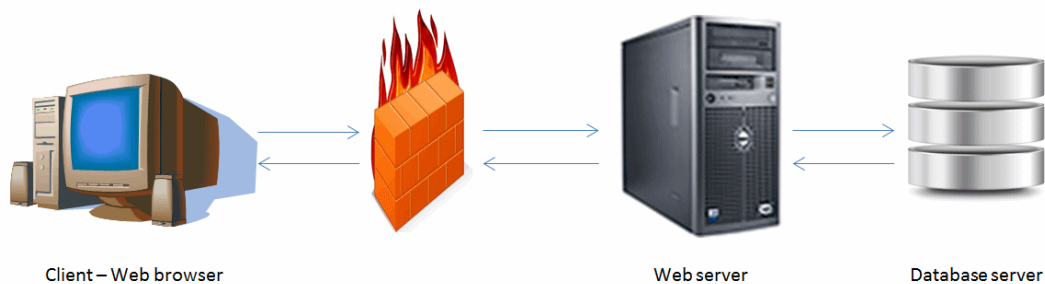
## Introduzione

Con il progressivo incremento, anno dopo anno, della potenza elaborativa dei nuovi calcolatori immessi sul mercato, la virtualizzazione si è dimostrata essere una tecnologia utile e versatile per migliorare la gestione dei *datacenter*, incrementare la disponibilità dei servizi, nonché semplificare il processo di sviluppo, test e distribuzione del *software*. Tutto ciò è possibile in quanto il disaccoppiamento totale del sistema operativo e delle applicazioni (*software*) dalle risorse fisiche sottostanti (*hardware*), con l'incapsulamento del tutto in una macchina virtuale facilmente trasportabile, ha reso fattibili configurazioni prima difficilmente realizzabili, se non quasi impossibili. Come, ad esempio, porre un servizio, che non prevede la sua configurazione in *cluster*, in altissima affidabilità (24x7x365) senza che sia necessario pianificare indisponibilità per eventuali *backup* e manutenzione *hardware*. La virtualizzazione è spesso confusa con la "simulazione", o anche l'"emulazione", tuttavia non è nessuna di queste due cose: la virtualizzazione è la possibilità di far girare più sistemi operativi contemporaneamente all'interno di una stessa macchina fisica.

### 1. La strada verso la virtualizzazione

Fino agli inizi degli anni '90, i centri di calcolo che fornivano servizi a grandi quantità di utenti erano dotati principalmente di elaboratori centrali monolitici (*mainframe*), dotati di elevata capacità di calcolo e grandi quantitativi di memoria per poter gestire le numerose sessioni degli utenti, i quali si collegavano tipicamente mediante terminali passivi, ovvero che non apportavano alcun contributo alla potenza elaborativa del sistema. Questa soluzione aveva come vantaggio una gestione centralizzata dell'intero sistema, a partire dalle politiche di accesso e sicurezza, alla gestione delle quote su disco, al calcolo delle risorse utilizzate e relativa contabilizzazione. Come controindicazione, i costi si rivelavano estremamente elevati per questo tipo di soluzione, per altro difficilmente scalabile: potenziare il sistema molto spesso significava sostituirlo con un altro completamente nuovo di livello superiore, con ovvie implicazioni sugli investimenti economici. Il fatto poi che queste piattaforme fossero assai poco *standard*, anzi praticamente chiuse, costituiva il secondo grande limite. Una volta scelto un fornitore di *hardware* (che fosse IBM, o Digital, o Sun Microsystems, o Bull, ecc.) ci si legava completamente ad esso sia per l'acquisto di eventuali periferiche aggiuntive, sia per il sistema operativo, che per gli stessi contratti di assistenza. Questo fatto incideva di nuovo negativamente sui costi al cliente finale.

Gli anni alla fine di quel decennio ed all'inizio del 2000 portarono profonde innovazioni tecnologiche nel settore della microelettronica. Microprocessori sempre più potenti e miniaturizzati furono messi in commercio su vasta scala, e questo rese possibile la diffusione di una nuova generazione di macchine da calcolo di dimensioni ridotte, ma comunque dotate di buona potenza elaborativa, e soprattutto economiche. Contemporaneamente, la comparsa delle prime versioni di Linux dava la possibilità di installare senza spese un sistema operativo alternativo alle versioni Unix fornite con *hardware* proprietario (Sun Solaris, IBM Aix, Digital True64 UNIX, ecc.). Questa evoluzione portò a profondi cambiamenti nella realizzazione dei centri di calcolo, il paradigma dei sistemi cosiddetti *legacy*, e più in generale delle architetture proprietarie, visse un rapido declino, mentre proliferarono i più compatti *server Intel-based*, termine derivante dal nome del produttore Intel Corporation, che con le proprie CPU ridusse rapidamente le quote di mercato di processori come Sun Sparc, MIPS e Digital Alpha. Inoltre, avere come destinazione una sola piattaforma *hardware* di riferimento per cui sviluppare *software* applicativo, sia a livello commerciale che *open-source*, costituiva indubbiamente una situazione più favorevole per la diffusione delle applicazioni rispetto a dover fronteggiare molteplici architetture differenti. Poiché si rivelò economicamente conveniente acquistare più macchine di medio livello piuttosto che una al top della gamma, il precedente concetto di centralizzazione cedette il passo a due nuove idee: calcolo distribuito e scalabilità. Si diffusero così i sistemi *client-server* a più livelli, dove ogni servizio girava su una macchina fisica ad esso dedicata, sia per ragioni di prestazioni, che sicurezza ed affidabilità (figura 1), ed i *cluster*, ovvero sistemi a più macchine interconnesse fra loro, utilizzati per realizzare l'alta affidabilità dei servizi (se un nodo subisce un guasto il servizio continua ad essere disponibile tramite i nodi superstiti) ed il bilanciamento di carico sui nodi.



**Figura 1.** Esempio di sistema client-server a 3 livelli.

Questo proliferare di *server*, innescato inizialmente da un risparmio economico sull'acquisto di *hardware*, causò tuttavia in pochi anni enormi incrementi nei cosiddetti costi "nascosti", ovvero di gestione dei *datacenter*, a partire dallo spazio fisico occupato dai numerosi rack ed eventuali spese di affitto, alle grosse problematiche di raffreddamento di locali con sempre più elevata densità di macchine per unità di volume e di conseguenza di calore prodotto, fino all'importante costo per l'energia elettrica necessaria per far funzionare sia i computer stessi che i condizionatori dedicati al raffreddamento dell'intero CED. Non meno importante infine il costo del personale dedicato alla gestione del centro: amministratori di reti, di sistemi e di applicativi. Nella seconda metà del decennio scorso, la concomitanza con i periodi di crisi economica con cui la maggioranza delle aziende mondiali ha dovuto confrontarsi, ha determinato la necessità di iniziare un processo di razionalizzazione delle risorse informatiche e dei costi associati, il quale si è concretizzato nel cosiddetto "consolidamento" dei servizi erogati, ovvero riduzione del numero di macchine fisiche presenti nel *datacenter* mediante virtualizzazione dei *server*, al fine di aumentarne l'efficienza.

## 2. La virtualizzazione: che cosa è e come funziona

### 2.1 Concetti generali

Il concetto alla base del "consolidamento" si può spiegare in modo molto semplice. La pratica sopra descritta di separare ogni servizio (ad esempio: *web server*, *database server*, *application server*, *domain controller*, *mail server*, acquisizione dati, DNS, ecc.) su di una macchina fisica specializzata introduce una maggiore robustezza, sia in termini di sicurezza di accesso ai sistemi, sia perché si evitano probabili conflitti ed incompatibilità software dovute a più installazioni sulla stessa macchina, ma anche grandi inefficienze nell'utilizzo delle risorse della macchina. Mentre infatti un programma di calcolo scientifico o simulazione riesce magari a sfruttare a pieno la potenza elaborativa di una macchina, altre applicazioni come un programma di acquisizione dati, oppure un DNS, utilizzano solo una piccola percentuale della CPU o della memoria RAM, altre invece effettuano prevalentemente I/O su disco, altre ancora generano sostanzialmente traffico di rete. Ma un computer la cui CPU è sfruttata prevalentemente per una piccola percentuale (ad esempio al 5%) purtroppo non consuma in modo proporzionale, bensì attesta il proprio fabbisogno energetico solo poco al di sotto del suo valore massimo, quindi il risparmio risulta trascurabile.



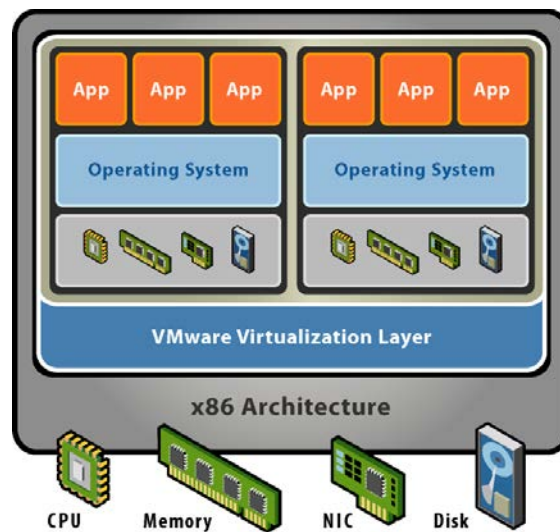
**Figura 2.** Schematizzazione di una macchina virtuale.

Ipotizziamo invece, a titolo di esempio, uno scenario teorico composto di 10 *server*, ognuno dei quali dotato di 10 GB di memoria e su cui gira un programma che usa mediamente solo il 10% di CPU ed 1 GB di RAM. Se si riuscisse a spostare queste 10 applicazioni su un'unica macchina, questa potrebbe idealmente lavorare al massimo delle proprie capacità e le altre 9 potrebbero rimanere spente, ottenendo in tal caso un risparmio teorico del 90% in termini di alimentazione elettrica e condizionamento.

La virtualizzazione aiuta in tal senso, in quanto permette essenzialmente di incapsulare un intero ambiente software, completo di sistema operativo ed applicativi, all'interno di un contenitore virtuale (nella pratica composto da un insieme di *files* su disco), dotato di proprie risorse e paragonabile a tutti gli effetti ad una macchina fisica (figura 2). Proprio per il fatto di essere isolati, più sistemi



virtuali possono girare contemporaneamente su una medesima macchina fisica, condividendone tutte le risorse *hardware*, quali processore, memoria, disco, rete, ecc. (figura 3).



**Figura 3.** Più macchine virtuali attive sulla medesima macchina fisica.

La tecnologia della virtualizzazione è abbastanza consolidata e risale agli anni '60, tuttavia essa si è affermata verso la metà degli anni '90 sulle architetture Intel a 32 e 64 bit (x86 e x64). Le prime soluzioni sono state presentate da VMware con il rilascio prima del prodotto *desktop* VMWare Workstation, poi le successive implementazioni per *server*. Xen, nato come progetto di ricerca presso l'Università di Cambridge e poi rilasciato in ambito *open-source*, è diventato un software commerciale con XenSource, che ha rilasciato la prima release ufficiale di Xenserver nel 2004. XenSource è stata poi acquisita da Citrix nel 2007. Oramai molti nomi noti appaiono nel mondo dei prodotti per la virtualizzazione, tra cui Oracle e Microsoft.

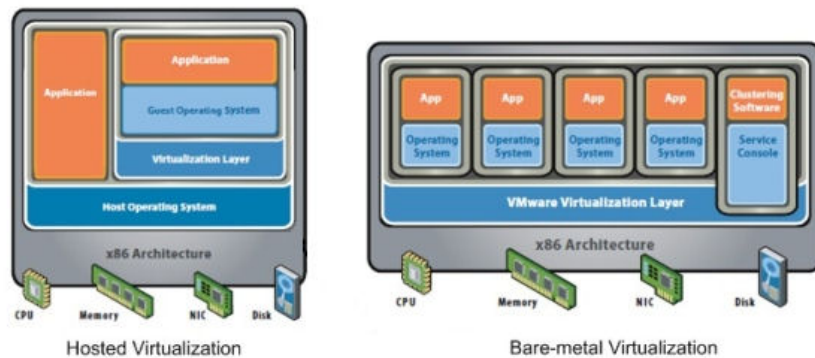
## 2.2 Tipi di virtualizzazione: *hosted* e *bare metal*

Con il termine “virtualizzazione” si intende in senso più generale l'operazione di separazione dei servizi *software* dalle componenti fisiche di una macchina, le quali, mediante questo processo di astrazione, vengono poi rese disponibili sotto forma di *hardware* virtuale. L'intera operazione è svolta da uno strato *software*, chiamato *virtualization layer* o *hypervisor*, il quale si prende carico di tradurre le chiamate alle risorse virtuali in accessi al sottostante dispositivo fisico, nonché di gestire dinamicamente il partizionamento e condivisione delle risorse risolvendo le eventuali contese quando più macchine virtuali sono in funzione contemporaneamente.

I tipi di approccio alla virtualizzazione sono fondamentalmente due, chiamati in gergo *hosted* e *bare metal*. Nel tipo *hosted*, lo strato di virtualizzazione è una applicazione che si installa in una macchina tradizionale dotata di proprio sistema operativo (come ad esempio MS Windows, Mac OSx o Linux). In questo caso l'accesso all'*hardware* non è esclusivo, bensì mediato dal sistema operativo installato, e le risorse sono condivise con quest'ultimo. Il sistema ospitante (*host*) e quello nella macchina virtuale (*guest*) sono isolati in termini di processi e memoria, ma la maggiore controindicazione è che un *crash* del sistema *host* fa automaticamente bloccare anche la macchina virtuale, in quanto l'*hypervisor* poggia comunque sul sistema operativo *host*. Alcuni esempi attuali di software di virtualizzazione di tipo *hosted* sono: VMWare Workstation, VMWare Player ([http://www.vmware.com/products/desktop\\_virtualization/](http://www.vmware.com/products/desktop_virtualization/)), Virtualbox (<https://www.virtualbox.org/>), Virtual PC (<http://www.microsoft.com/windows/virtual-pc/default.aspx>), Qemu ([http://wiki.qemu.org/Main\\_Page](http://wiki.qemu.org/Main_Page)). Questi software di virtualizzazione supportano la maggior parte dei sistemi operativi attualmente disponibili.

Nel tipo *bare metal*, invece, l'*hypervisor* si installa direttamente sull'*hardware* della macchina e ne prende pieno possesso, senza l'intermediazione di un sistema operativo (figura 4). Poiché l'accesso alle risorse è diretto, questa soluzione è più efficiente della precedente e presenta vantaggi come maggiore robustezza, scalabilità e prestazioni. Per contro, la base di compatibilità è molto più ristretta rispetto al tipo

*hosted*, perché i requisiti hardware sono molto più stringenti. Gli esempi più significativi di questa categoria sono: VMWare ESX e VMWare vSphere (<http://www.vmware.com/products/vsphere/mid-size-and-enterprise-business/overview.html>), Citrix Xenserver ([http://www.citrix.it/Prodotti\\_e\\_Soluzioni/Prodotti/XenServer/](http://www.citrix.it/Prodotti_e_Soluzioni/Prodotti/XenServer/)), Microsoft Hyper-V (<http://www.microsoft.com/en-us/server-cloud/hyper-v-server/default.aspx>)



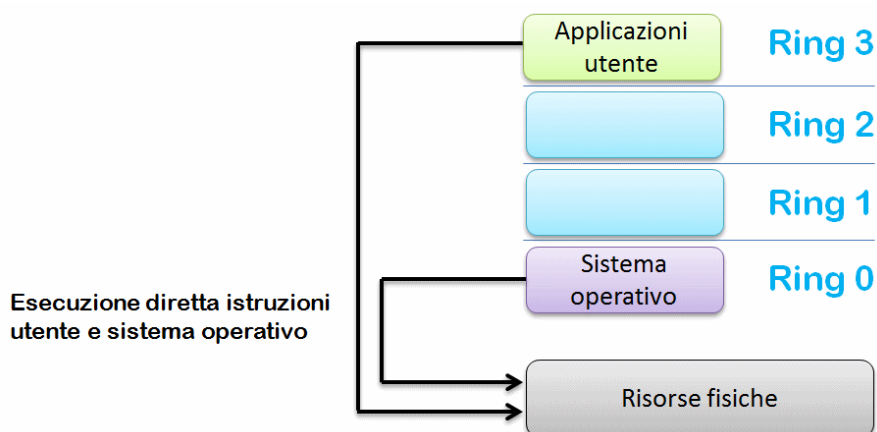
**Figura 4.** Schemi dei due tipi di virtualizzazione.

Tra tutti i vari fornitori oggi sul mercato, VMWare è quella con la maggiore esperienza, in quanto è stata la prima a sviluppare soluzioni in entrambe le tipologie di virtualizzazione. In particolare, il prodotto di classe *enterprise*, attualmente denominato vSphere, ha raggiunto un tale livello di affidabilità e maturità da posizionarsi avanti alla concorrenza presso i principali clienti sul panorama mondiale.

## 2.3 Come funziona la virtualizzazione

### 2.3.1 CPU

La prima grande sfida della virtualizzazione sull'architettura x86 si è avuta sul funzionamento della CPU. Su questa architettura, i sistemi operativi sono progettati per funzionare direttamente sulla macchina fisica e pertanto assumono di avere il pieno controllo dell'*hardware*. L'architettura x86 offre al sistema operativo ed alle applicazioni 4 livelli di privilegi per la gestione dell'accesso all'*hardware*, i quali vengono identificati come Ring 0, 1, 2 e 3 (figura 5).

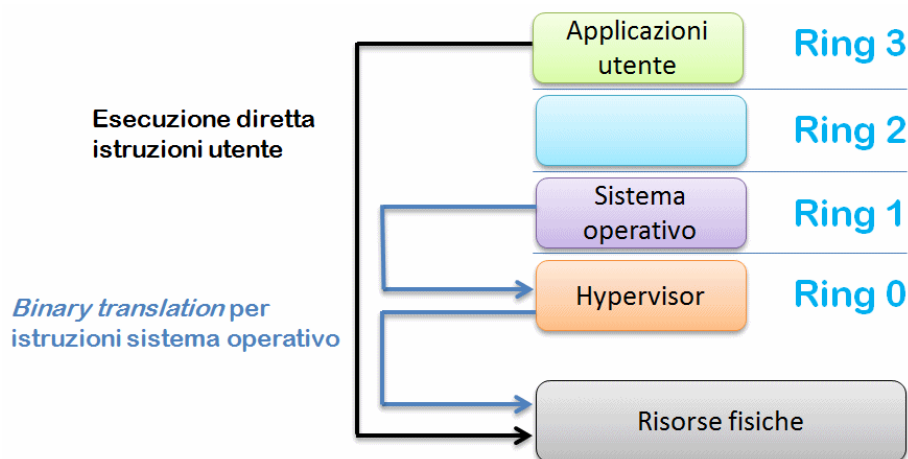


**Figura 5.** Livelli di privilegi dell'architettura x86 senza virtualizzazione.

Mentre le applicazioni utente generalmente girano sul livello meno privilegiato (Ring 3), il sistema operativo necessita del massimo dei privilegi per poter avere accesso diretto alla memoria e alle altre risorse fisiche, pertanto gira al livello Ring 0. Virtualizzare tale architettura implica inserire l'*hypervisor* al di sotto del sistema operativo, al Ring 0 in quanto è quest'ultimo che ora media gli accessi alle risorse fisiche e presenta ai livelli superiori quelle virtuali. Questo implica traslare il sistema operativo ad un livello

superiore, ma la situazione a questo punto si complica per il fatto che alcune istruzioni cambiano completamente semantica quando non vengono eseguite al livello di Ring 0. Proprio la difficoltà nel gestire e modificare queste istruzioni in fase di esecuzione costituì agli inizi un grosso ostacolo, tanto da far ritenere la virtualizzazione su questa architettura non fattibile.

La soluzione fu trovata nel 1998 da VMWare mediante le tecniche della *binary translation* e *direct execution*. Il sistema operativo viene spostato sul Ring 1 (figura 6) e l'*hypervisor* ne traduce tutte le istruzioni non virtualizzabili in una diversa serie di chiamate che mantiene però lo stesso effetto sull'*hardware* virtuale. Le applicazioni utente nel Ring 3 invece continuano ad essere eseguite direttamente dal processore per mantenere elevate prestazioni.



**Figura 6.** Livelli di privilegi dell'architettura x86 con virtualizzazione.

Questa soluzione viene anche chiamata *full virtualization*, in quanto il sistema operativo ospite (*guest*) è completamente disaccoppiato dall'*hardware* sottostante per mezzo del *virtualization layer*, quindi non si accorge di trovarsi in un sistema virtualizzato piuttosto che in uno fisico. Questo comporta il grande vantaggio che il sistema operativo non necessita di alcuna modifica per poter essere installato all'interno di una macchina virtuale, semplificando così migrazioni dal fisico al virtuale e la portabilità di macchine virtuali; inoltre non è richiesto alcun *hardware* particolare dal momento che l'*hypervisor* si prende carico di tradurre in tempo reale le istruzioni e mantenere in una *cache* i risultati per velocizzare analoghe operazioni future. La *full virtualization* offre inoltre il miglior grado di sicurezza ed isolamento fra le macchine virtuali.

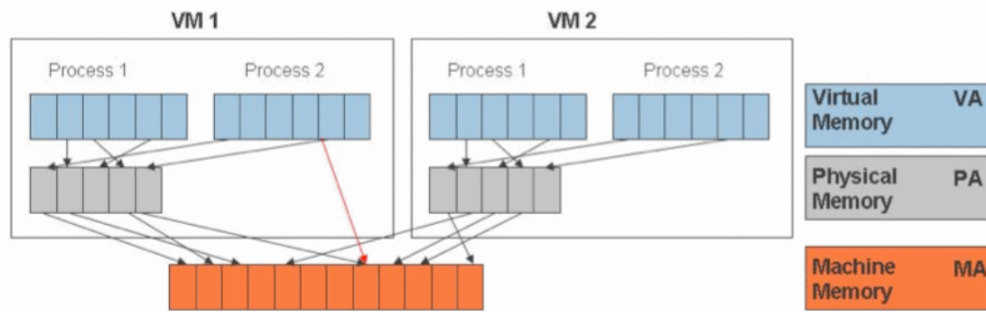
La *paravirtualization* si propone come un'alternativa alla *full virtualization*, la differenza principale consiste nella necessità di un sistema operativo con il *kernel* modificato, questo per evitare la totale traduzione delle istruzioni (*binary translation*) e ridurre così il carico di lavoro extra introdotto da quest'ultima. Lo svantaggio principale è l'impossibilità di virtualizzare sistemi operativi non modificati, come ad esempio Windows 2000 e Windows XP.

La terza alternativa è costituita dalla *hardware assisted virtualization*, introdotta con le tecnologie VT-x di Intel ed AMD-V di AMD. I nuovi processori dotati di tali tecnologie presentano un ulteriore livello al di sotto del Ring 0, chiamato *root mode level*, in cui gira il *virtualization layer* il quale intercetta le chiamate provenienti dal soprastante Ring 0, senza più la necessità di effettuare la *binary translation*. Questa è la tecnologia di virtualizzazione della CPU più recente, introdotta dal 2006, e probabilmente in futuro diventerà la più utilizzata.

### 2.3.2 Memoria

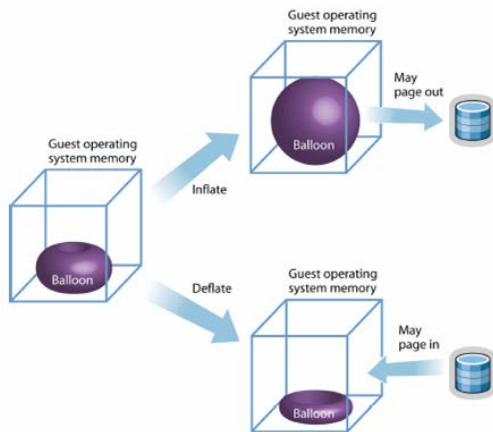
Dopo il processore, la memoria è l'altro componente fondamentale da virtualizzare, ovvero da suddividere ed allocare dinamicamente alle varie macchine virtuali in esecuzione. Il meccanismo è molto simile alla gestione della memoria virtuale dei moderni sistemi operativi: le applicazioni vedono uno spazio di indirizzamento contiguo, non necessariamente strutturato allo stesso modo sulla memoria fisica, in quanto il sistema operativo si occupa di tenere traccia della mappa tra le pagine virtuali e le risorse reali. Le moderne CPU contengono due componenti, la *memory management unit* (MMU) ed il *translation lookaside buffer* (TLB), proprio per ottimizzare la gestione della memoria virtuale. Per eseguire più macchine virtuali

su un singolo sistema fisico, si rende quindi necessario un altro livello di virtualizzazione della memoria (figura 7) [Waldspurger, 2002].



**Figura 7.** Virtualizzazione della memoria.

Il sistema operativo *guest* continua nella sua funzione di mappatura delle pagine di memoria virtuale, con la differenza che esso non ha più accesso diretto alla memoria fisica della macchina, ma questo compito ora è eseguito dall'*hypervisor*, il quale usa il TLB per mappare la memoria virtuale direttamente su quella reale (*machine memory*), evitando così la doppia traduzione ad ogni accesso (figura 7).

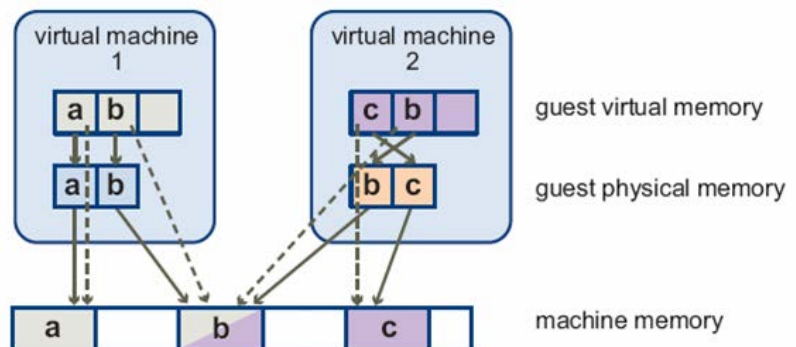


**Figura 8.** La tecnica del *ballooning*.

Analogamente a quanto accade per la CPU, la quale viene riservata alle varie macchine virtuali a seconda delle necessità di calcolo, così anche per la memoria sono implementate varie tecniche per ottimizzarne l'utilizzo. La *memory overcommitment* consente di allocare per le macchine virtuali complessivamente più memoria di quella realmente disponibile sull'*host* fisico (ad esempio assegnare a 3 macchine virtuali 2 GB di RAM ognuna su un *host* con soli 4 GB di RAM). Questa tecnica si basa sul fatto che, in media, le macchine virtuali non utilizzino il 100% delle risorse assegnate, pertanto quelle al momento non utilizzate vengono reclamate dall'*hypervisor* e temporaneamente riassegnate in tempo reale sulla base delle esigenze. La tecnica con cui ciò avviene è detta *ballooning* (figura 8) e si rende necessaria proprio perché la macchina virtuale è isolata, il sistema operativo *guest* è convinto di trovarsi

su un server fisico e non è consapevole dell'esistenza di eventuali altre macchine virtuali in funzione contemporaneamente ad esso, né in grado di rilevare il grado di utilizzo delle risorse reali. Pertanto l'*hypervisor* comunica, mediante un *driver* installato nel sistema *guest*, con uno pseudo-dispositivo che quando si espande assegna più memoria alla macchina virtuale, quando invece si contrae ne rilascia una parte che può venire riassegnata ad altre VM tramite lo stesso meccanismo.

Un'altra tecnica interessante è il *memory sharing*, che risulta efficace quando sullo stesso *host* sono in funzione più VM con il medesimo sistema operativo *guest*, ancora meglio se fanno girare le stesse applicazioni oppure contengono dati comuni. In



**Figura 9.** Il *memory sharing*.

questa situazione, l'*hypervisor* elimina dalla memoria fisica le eventuali pagine di memoria duplicate, guadagnando spazio per un ulteriore *overcommitment*. Chiaramente le pagine comuni sono in sola lettura, qualora dovesse rendersi necessaria una scrittura, tale pagina verrebbe prima copiata e poi sovrascritta, al fine di mantenere l'isolamento fra le macchine virtuali.

### 2.3.3 I/O

Anche per quanto riguarda le periferiche di *Input/Output*, l'*hypervisor* offre al sistema *guest* un insieme di dispositivi standard (figura 10), i quali emulano altrettanti dispositivi noti (scheda di rete, scheda video, controller disco, ecc.) e traducono le richieste all'*hardware* di sistema. Questa standardizzazione aumenta la portabilità delle macchine virtuali, le quali possono girare indipendentemente dal particolare hardware presente sulla macchina fisica.

Nel caso della virtualizzazione delle risorse di rete, l'introduzione di schede (*virtual NIC*) e *switch* virtuali (*virtual switch*) semplifica enormemente la creazione di una infrastruttura di rete virtuale. Si possono realizzare topologie anche abbastanza complesse senza che sia necessario acquistare un singolo componente reale aggiuntivo. Inoltre, tutto il traffico di rete che transita all'interno dei *virtual switch* di un medesimo *host* non impatta minimamente sul consumo di banda della rete fisica, in quanto i pacchetti vengono effettivamente immessi nella rete solo quando è necessario che escano realmente dall'*host*.

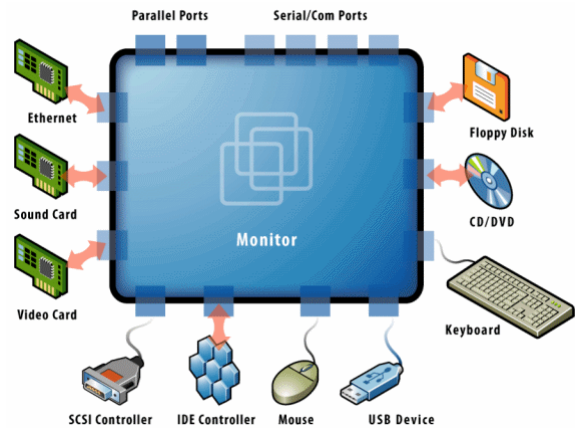


Figura 10. I dispositivi di I/O.

## 2.4 Principali vantaggi: *High Availability* e *live motion*

Come già accennato nell'introduzione, la virtualizzazione di tipo *bare metal* permette di disaccoppiare completamente il contenitore virtuale dall'*hardware* sottostante, questo apre delle possibilità molto interessanti. Si ipotizzi ad esempio una configurazione con un numero  $n$  di macchine fisiche dotate di un *hypervisor* e collegate ad una rete locale (*Local Area Network*). Inoltre, i file delle macchine virtuali sono memorizzati su uno spazio disco condiviso (*Storage Area Network*) ed accessibile da tutti gli *host* (figura 11).

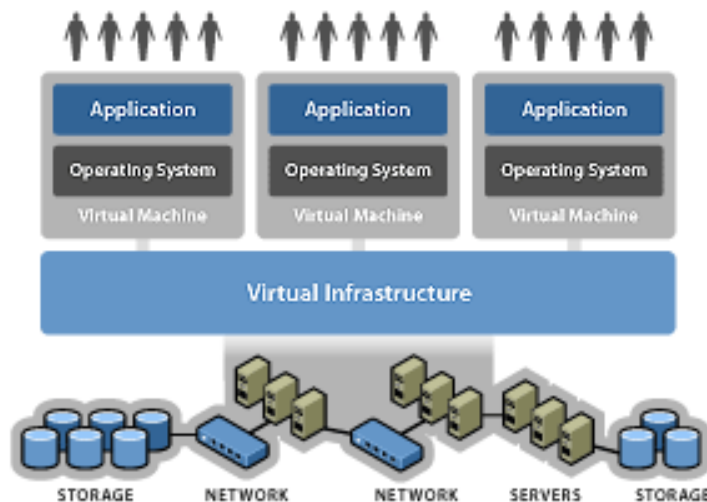
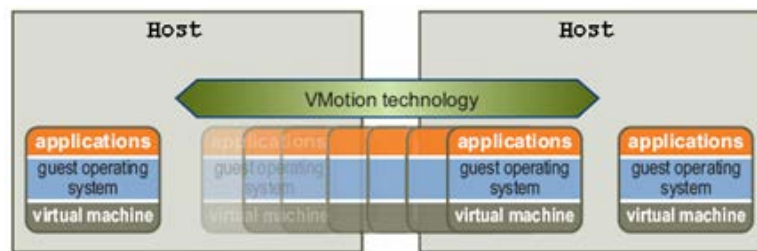


Figura 11. Infrastruttura di virtualizzazione.

Quello che si può realizzare è la cosiddetta "alta affidabilità". Il fatto di memorizzare i file di una VM su uno storage condiviso, consente tecnicamente a qualsiasi nodo appartenente all'infrastruttura di avviare

quella VM. Questo implica che, in caso di guasto hardware improvviso del nodo su cui al momento sta funzionando quella VM, uno qualsiasi dei nodi rimanenti può riavviare la VM che si è spenta insieme al nodo guasto, recuperando tutto ciò che è stato scritto su disco, e perdendo solamente lo stato della memoria all'istante del *crash*. Questa funzionalità era già presente nei *cluster*, tuttavia la loro realizzazione comporta molto spesso configurazioni abbastanza complesse del sistema operativo, ed a volte la necessità di costose licenze aggiuntive (vedi ad esempio Oracle Real Application Cluster). In questo modo invece qualsiasi macchina virtuale, e quindi qualsiasi applicazione che gira in esse, può beneficiare del *cluster* di *host* (nodi con *hypervisor*), cosicché si possono configurare in alta affidabilità anche servizi che non forniscono in modo nativo questa funzione.

Una evoluzione tecnica dell'alta affidabilità è il *live motion*, o anche *live migration*. Sempre grazie al fatto che lo storage è centralizzato e condiviso da tutti, una VM può essere facilmente trasferita da un *host* ad un altro senza che si renda necessario spegnerla sul nodo di partenza e riaccenderla su quello di destinazione, bensì mentre questa è funzionante e continua ad erogare i propri servizi (da cui il termine *live*). Questo è possibile facendo una fotografia dell'area di memoria in uso alla VM ad un certo istante, successivamente questa memoria viene copiata sul nodo di destinazione. Contemporaneamente alla copia, il nodo origine tiene traccia di eventuali modifiche alla memoria che intercorrono durante il processo di copia (figura 12). Terminata la copia, la VM viene congelata per pochissimi istanti, necessari al rilascio della stessa da parte del nodo origine e presa in carico da parte del nodo destinazione, nonché dell'applicazione delle eventuali differenze accumulate durante la prima copia dell'area di memoria. A questo punto la VM è completamente funzionante sul nuovo nodo e nessun utente ha avvertito disservizi, se non al massimo qualche istante di blocco, assimilabile ad uno dei possibili ritardi di rete.



**Figura 12.** *Live migration.*

Questa funzionalità è estremamente interessante, in quanto consente di spostare a piacimento le macchine virtuali all'interno della propria infrastruttura senza che questo abbia alcun impatto sulla produttività del sistema. La prima applicazione utile è la manutenzione hardware degli *host*: se un server necessita di intervento di riparazione o di aggiornamento (aggiunta RAM, sostituzione scheda guasta, ecc.), si spostano prima tutte le VM attive su quel nodo e poi lo si spegne senza inficiare sulla disponibilità dei servizi. Senza la virtualizzazione, la chiusura del servizio attivo su quel *server* per il tempo necessario alla manutenzione sarebbe stata inevitabile. Il secondo campo di applicazione della *live migration* è il bilanciamento di carico degli *host*. Se ad esempio ci sono alcune macchine virtuali che consumano molte risorse e sono concentrate su un unico host, questo risulta sovraccaricato con conseguente degrado delle prestazioni globali, mentre altri magari sono fortemente sottoutilizzati. Ebbene, con la *live migration* si possono spostare “a caldo” le VM particolarmente critiche, bilanciando in modo più uniforme il carico su tutti i *server*, nuovamente senza il minimo impatto negativo sulla funzionalità del sistema.

### 3. Il *private cloud*

La definizione di *cloud computing* è ancora abbastanza controversa, in quanto il termine è piuttosto vago e viene utilizzato in molti contesti con significati spesso diversi fra loro. Volendo adottare la definizione data da Gartner Inc. [Gartner, 2008], una delle maggiori società al mondo di consulenza ed analisi di mercato nel settore della tecnologia e dell'informatica, si può considerare il *cloud computing* come una nuova modalità di utilizzo dell'infrastruttura tecnologica, in cui quantità altamente scalabili di risorse informatiche vengono fornite pacchettizzate in servizi (“*as a service*”) ai vari clienti esterni, i quali ne usufruiscono in remoto tramite le moderne tecnologie *internet*. Si parla di *cloud* “pubblico” (*public cloud*)

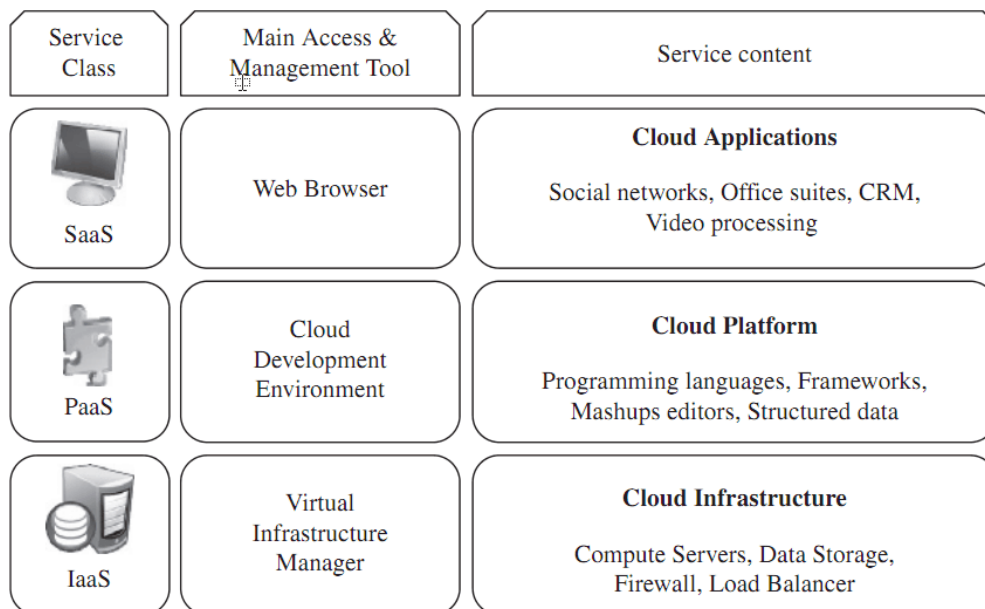
quando sia l'infrastruttura *hardware* che quella applicativa sono messe a disposizione dal fornitore di servizi, il quale ne detiene il pieno controllo, mentre l'utente finale in genere accede semplicemente ad un portale e paga un canone sulla base delle risorse effettivamente utilizzate. Questa soluzione presenta da un lato il grande vantaggio di semplificare enormemente la gestione dell'infrastruttura IT (*Information Technology*), nel caso in cui questa non rappresenti un centro fondamentale nell'attività dell'azienda o dell'ente; d'altro canto invece la completa delocalizzazione delle risorse di calcolo e spazio per l'archiviazione dei dati incide sensibilmente sulle problematiche di sicurezza e riservatezza delle informazioni, tanto da rendere questa soluzione completamente impraticabile laddove esistano stringenti normative di tipo legale.

All'estremo opposto si colloca il *cloud* "privato" (*private cloud*), ovvero quando l'intera infrastruttura è comunque gestita all'interno dell'organizzazione e solitamente anche di proprietà della stessa. Sono state mosse alcune critiche da esperti del settore a questo tipo di modello, in quanto in questo caso le aziende devono comunque sobbarcarsi i costi di acquisto e gestione delle risorse, vanificando in un certo senso il concetto di risparmio economico che rende interessante il *cloud computing*. Tuttavia nei casi in cui non si può prescindere dal possedere un'infrastruttura tecnologica in casa, il *cloud* privato può rappresentare comunque un modo per ottimizzare lo sfruttamento delle risorse e massimizzare il ritorno degli investimenti.

In modo abbastanza intuitivo, il *cloud* "ibrido" (*hybrid cloud*) è una composizione dei due modelli precedenti, in cui i servizi più critici risiedono sulla parte interna, mentre quelli meno strategici sono allocati su fornitori esterni, con la possibilità di migrare da infrastruttura privata a quella pubblica e viceversa a seconda delle necessità di risorse oppure dei rapporti costi/benefici.

### 3.1 La classificazione dei servizi

I servizi erogabili mediante il *cloud computing* vengono classificati in tre tipologie principali, sulla base del livello di astrazione di ciò che viene fornito. Queste classi si possono considerare come i livelli di un'architettura dove i servizi di una classe possono usufruire dei contributi dei servizi delle classi sottostanti (figura 13) [Voorsluys et al., 2011].



**Figura 13.** Le tre classi di servizi erogabili.

- *Infrastructure as a Service (IaaS)*: il livello base di questa architettura offre interi *server*, tipicamente sotto forma di macchine virtuali, così come altre risorse di tipo infrastrutturale, quali *storage*, *firewalls*, *load balancers*, *Local Area Networks*, spazi di indirizzamento IP, *Virtual Private Networks*. In questo livello l'utente si configura le risorse secondo necessità, esattamente come farebbe con una macchina fisica, scegliendo il sistema operativo, avviando o spegnendo la macchina, installando applicazioni, aggiungendo dischi virtuali, configurando le politiche di accesso ed autorizzazioni.

- Platform as a Service (PaaS): si pone ad un livello di astrazione maggiore rispetto al precedente, in quanto offre un intero ambiente di sviluppo in cui l'utente può realizzare le proprie applicazioni, senza necessariamente doversi preoccupare della configurazione o del dimensionamento delle risorse quali potenza di CPU o memoria o altro. Un esempio tipico è la soluzione per sviluppare siti web, comprendente sistema operativo, uno o più linguaggi di programmazione (php, java, python, ecc.), *server web* e *database* per l'archiviazione dati, così come l'offerta per la connettività, generalmente regolamentata sulla base del traffico dati generato.
- Software as a Service (SaaS): è il livello più alto, in cui viene fornito direttamente il prodotto *software* e gli utenti accedono ad esso via rete mediante i *client*, costituiti sia dal classico *web browser* che da applicazioni molto leggere installate sui dispositivi utente, tipicamente quelli portatili. Anche in questo modello, l'utente non si deve preoccupare di configurare o dimensionare le risorse sottostanti, quali memoria, rete, *storage*, ecc., né dello sviluppo o manutenzione dell'applicazione.

### 3.2 Dalla virtualizzazione al *cloud*

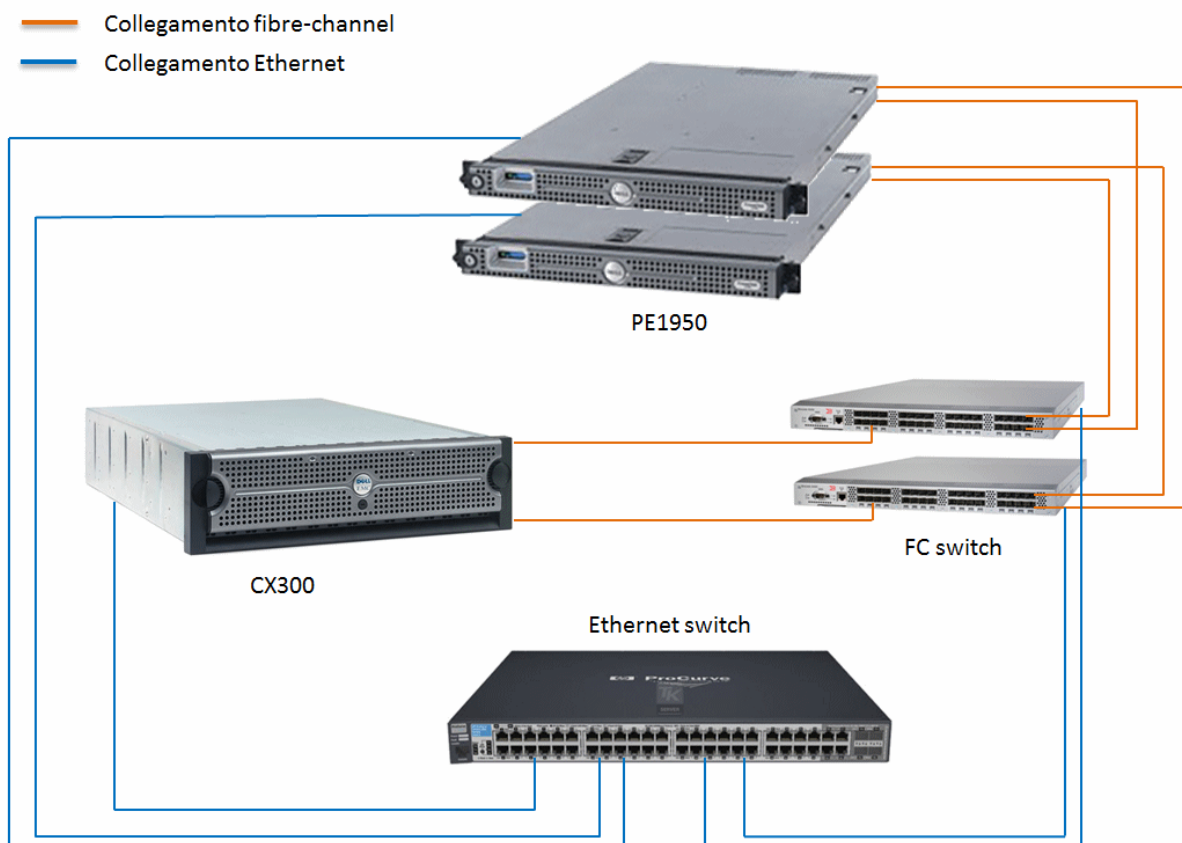
Una piattaforma *cloud* inizia sempre con una infrastruttura di virtualizzazione, tramite cui è possibile, secondo quanto illustrato in precedenza, aggregare risorse, clonare e creare rapidamente nuovi sistemi, migrare servizi attivi. Secondo la definizione data dal NIST [Mell and Grance, 2011], vi sono alcune caratteristiche essenziali che devono essere soddisfatte affinché una infrastruttura *cloud* possa essere considerata tale.

- Automatismo e autonomia: gli utenti devono essere in grado di allocarsi risorse, come potenza di calcolo e spazio su disco, sulla base delle proprie esigenze e senza intervento manuale da parte del fornitore.
- Aggregazione: l'infrastruttura *cloud* deve essere costruita su gruppi di risorse virtualizzate aggregate e condivise (CPU, memoria, *storage*, *networking*, ecc.) ed assegnate dinamicamente ai differenti utenti. L'architettura *software* deve essere *multi-tenant*, ovvero progettata per partizionare virtualmente i dati e le configurazioni, così che ogni differente utente lavora con un'istanza personalizzata dell'applicazione virtuale di interesse.
- Standard aperti ed interoperabilità: l'accesso alle risorse del *cloud* deve avvenire tramite protocolli standard, senza la necessità di alcuna procedura personalizzata oppure *client* dedicati.
- Flessibilità: il *cloud* deve fornire e rilasciare dinamicamente le risorse in modo automatico in base al carico di lavoro, ed essere in grado di scalare in tempi sufficientemente rapidi, tali che l'utente percepisca che le risorse utilizzate sono in ogni momento quelle più appropriate.
- Misurabilità: l'utilizzo di tutti i tipi di risorse deve essere calcolabile in modo trasparente, sia per l'utilizzatore che per il fornitore di servizi, il costo addebitato all'utente dipende solo dalla quantità di risorse effettivamente utilizzate o che egli dichiara di volersi riservare.

## 4. L'infrastruttura realizzata a Roma2

Presso la sede distaccata della Sezione Roma2 dell'INGV, è presente un centro dati costituito alla luce dall'esigenza di adempiere agli obblighi istituzionali che l'Istituto ha nei confronti di alcuni Dicasteri, obblighi regolamentati da appositi accordi e convenzioni. Tra i requisiti riportati in tali accordi, vi era quello di garantire determinati livelli di confidenzialità e di sicurezza delle informazioni. All'interno del centro è stata realizzata una infrastruttura di virtualizzazione completa, il cui scopo iniziale era il consolidamento dei servizi erogati per far fronte all'obsolescenza delle macchine, nonché fornire una maggiore flessibilità nella gestione dell'esistente e delle installazioni future.



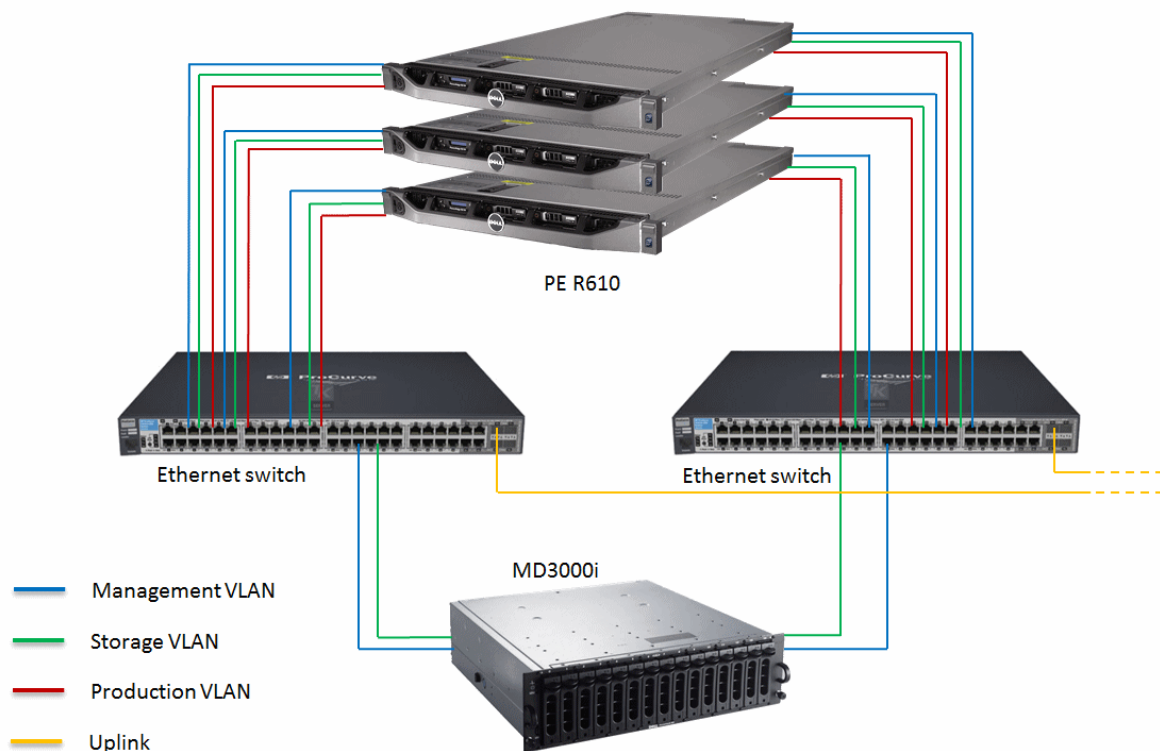


**Figura 14.** Cluster di sviluppo, test e backup.

Il processo è iniziato nella seconda metà del 2007, con una fase di *test* dei due prodotti di maggior rilievo disponibili al momento: Xenserver 4.0 e VMWare ESX 3. Dopo circa 6 mesi di prove condotte in parallelo sui due sistemi, la scelta finale è caduta sulla soluzione VMWare, per la sua maggiore affidabilità generale e completezza di funzionalità.

L'intera infrastruttura si articola su due *cluster*, di cui il più longevo (CLUSTER\_PE1950, figura 14), destinato ora ad ambiente di sviluppo, *test* e *backup*, è composto da 2 *server* da rack Dell PowerEdge 1950, ciascuno dei quali equipaggiato con 2 processori Xeon *Quad-core*, 16 GByte di memoria RAM, 6 porte di rete Ethernet da 1 Gbit/s e doppia scheda Qlogic per connettività *fibre channel*. Lo spazio su disco è fornito da uno *storage array* in tecnologia *fibre channel* Dell-EMC CX300, che pilota due box di espansione DAE con ulteriori dischi. Sia i *server* che lo *storage* sono connessi a due switch McData per connessioni in fibra, per realizzare la ridondanza dei *path*. Completa il quadro uno *switch* Ethernet HP Procurve 2650, che fornisce la connettività di rete a tutti i dispositivi del rack. Le risorse globali a disposizione del *cluster* ammontano quindi a 37 GHz di CPU, 32 GB di RAM e quasi 3 TB di spazio disco suddiviso in 8 *datastore*.

Il secondo e più performante *cluster* (CLUSTER\_R610, figura 15), il quale ospita tutti i servizi attualmente in produzione, è composto da 3 *server* da rack Dell PowerEdge R610, ciascuno dei quali equipaggiato con 2 processori Xeon *Esa-core*, 96 GByte di memoria RAM, 8 porte di rete Ethernet da 1 Gbit/s più una per la gestione remota. Mancano le schede per connettività *fibre channel*, in quanto lo spazio disco viene messo a disposizione da uno *storage array* iSCSI Dell MD3000i. La gestione della rete, compreso il partizionamento in VLAN di gestione, produzione e *storage*, è demandata ai due *switch* Ethernet ridondati HP Procurve 2910-48, dotati di 48 porte da 1 Gbit/s ed un modulo mini-GBIC ciascuno per l'*uplink* in fibra ottica verso il centro stella. Le risorse globali a disposizione del *cluster* ammontano quindi a 100 GHz di CPU, 288 GB di RAM e circa 6 TB di spazio disco suddiviso in 6 *datastore*.



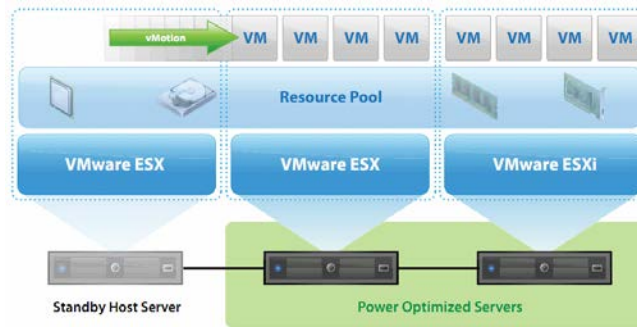
**Figura 15.** Cluster di produzione.

Sui 3 *server* del *cluster* di produzione è installato l' *hypervisor* VMWare vSphere 4.1 update 2, su scheda di memoria Secure Digital da 4 GB, dal momento che le macchine sono prive di *hard disk* proprio per minimizzare i rischi di guasti meccanici ai dischi, oltre che ridurre leggermente consumi e calore prodotto.

L'intero *cluster* è gestito da un VMWare VirtualCenter Server, affinché sia possibile usufruire di tutte le componenti avanzate disponibili con una licenza "Enterprise", tra cui si elencano le più significative:

- VMWare HA (High Availability): alta affidabilità dei servizi, ovvero se uno dei nodi del *cluster* si guasta, tutte le macchine virtuali attive al momento su quel nodo vengono automaticamente riavviate sui nodi superstiti; il periodo di disservizio è pari al tempo che impiega il VirtualCenter a decretare un nodo inattivo, più il tempo di riavvio della macchina virtuale.
- VMWare vMotion: migrazione a caldo senza interruzione di servizio (*live migration*). Una macchina virtuale accesa e funzionante, su cui possono essere al momento anche connessi eventuali *client*, può essere spostata da un nodo ad un altro del *cluster* senza la necessità che questa venga spenta, bensì continuando invece ad erogare il servizio ai *client* collegati, i quali non si accorgono dell'operazione in quanto non avvertono alcun disservizio. Come già spiegato precedentemente, questa funzionalità è estremamente utile in caso di manutenzione di un nodo, oppure per bilanciare l'utilizzo quando alcuni *host* sono particolarmente sovraccarichi rispetto ad altri.
- VMWare DRS (Distributed Resource Scheduler): le risorse del *cluster* vengono aggregate all'interno di contenitori logici (*resource pool*) e bilanciate dinamicamente sulla base di continui controlli sul loro utilizzo. Quando in una macchina virtuale si verifica un incremento di carico elaborativo, VMWare DRS alloca automaticamente risorse aggiuntive redistribuendo, tramite vMotion, le altre macchine virtuali tra i server fisici appartenenti a quel *resource pool* [VMWare, 2009].
- VMWare DPM (Distributed Power Management): è un componente del VMWare DRS, che controlla continuamente l'utilizzo delle risorse del *cluster*. Quando la richiesta di risorse diminuisce, nei periodi di minor carico di lavoro, VMWare DPM sposta le macchine virtuali consolidandole in un numero minore di nodi, spegnendo quelli completamente inutilizzati, così da risparmiare energia (figura 16).

Quando poi la richiesta di risorse ritorna a valori più alti, i nodi spenti vengono automaticamente riaccesi ed il carico nuovamente bilanciato fra di essi.



**Figura 16.** Funzionamento del DPM.

- **VMware FT (Fault Tolerance):** questa funzione è un'ulteriore evoluzione dell'alta affidabilità. Se una macchina virtuale è particolarmente critica, attivando su di essa la *fault tolerance* viene creata una copia identica (stesso indirizzo IP, stesso *MAC address*) della VM su un nodo differente, e questa copia viene tenuta perfettamente sincronizzata in tempo reale, replicando su di essa tutte le operazioni che interessano la VM primaria. Se la VM primaria dovesse subire un *crash* a causa di un problema sull'*host* dove essa risiede, immediatamente la VM secondaria prenderebbe il suo posto continuando a svolgere il lavoro in esecuzione al momento del guasto, senza alcun disservizio in quanto non viene perso lo stato della memoria e vi è continuità nell'esecuzione delle applicazioni.
- **VMWare Vmsafe:** questa nuova tecnologia è dedicata alla sicurezza, in quanto fornisce una serie di strumenti *software* (API) per ispezionare determinati aspetti di una VM, come gli accessi alla memoria, al disco, rete dati ed identificare così eventuale *software* malevolo attivo nella VM. Tale controllo granulare sull'utilizzo delle risorse virtuali non viene effettuato per mezzo di *agent* all'interno dell'ambiente virtuale, in quanto l'*agent* stesso potrebbe venire compromesso da un eventuale *malware*, bensì allo stesso livello dell'*hypervisor*, e ciò aumenta il livello di sicurezza.
- **Storage vMotion:** tecnologia concettualmente identica alla *live migration*, ma dedicata alla migrazione a caldo dei *file* su disco della macchina virtuale fra *storage array* anche eterogenei, a patto che essi siano visibili e condivisi tra tutti gli *host*. La migrazione viene finalizzata previa verifica sull'integrità totale della transazione e senza interruzione di servizio. Anche in questo caso, lo spostamento "a caldo" è molto utile in caso di manutenzioni programmate.

Alla luce di quanto discusso nel paragrafo 3.2, non si può ancora affermare in termini rigorosi di aver realizzato una infrastruttura *cloud*. Le funzionalità da implementare sono: a) la parte di automazione completa ed il portale relativo al *self-provisioning*, creare e distribuire gli ambienti virtuali richiede ancora l'intervento manuale dell'amministratore; b) occorre definire le metriche per il calcolo dell'utilizzo delle risorse e la relativa contabilizzazione di eventuali addebiti; c) manca la parte di gestione delle migrazioni degli ambienti virtuali verso altri *cloud* privati oppure verso eventuali *cloud* pubblici.

Il tassello mancante necessario per ottenere tutte le funzionalità richieste è l'acquisto e l'installazione del prodotto *software* VMWare vCloud, specificatamente creato per trasformare infrastrutture di virtualizzazione basate sull'*hypervisor* VMWare vSphere in veri e propri *cloud*. Esso si compone di vari moduli (figura 17), ognuno dei quali copre un aspetto ben determinato (ad esempio, "vShield" riguarda la sicurezza, "Chargeback" la contabilizzazione dell'utilizzo delle risorse, "Orchestrator" si occupa dell'automatizzazione dei processi ripetitivi, "Service Manager" gestisce tramite un portale personalizzabile il flusso "richiesta-notifica-approvazione" e la fase di configurazione e successive modifiche da parte dell'utente).

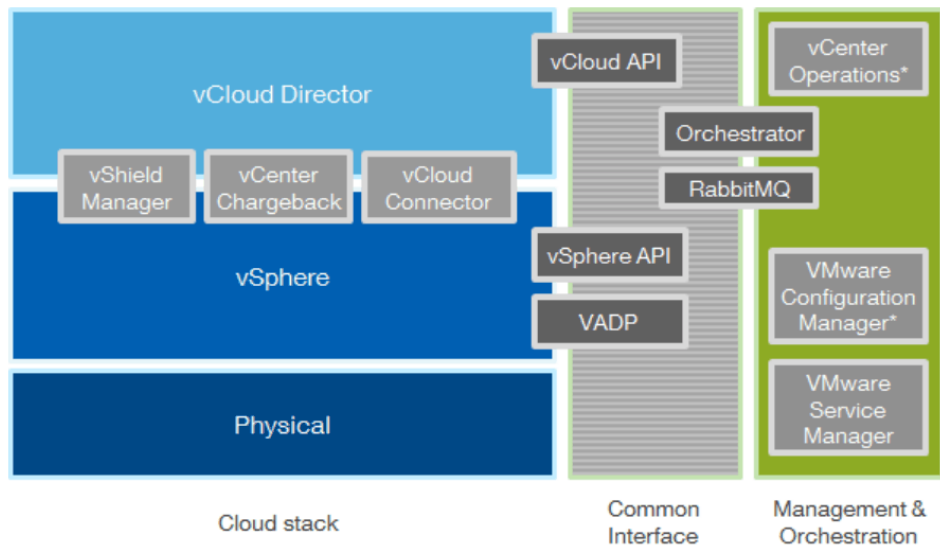


Figura 17. Componenti del prodotto software VMWare vCloud.

## 5. Applicazioni virtualizzate

Il primo obiettivo, dopo aver realizzato questa infrastruttura, è stato il consolidamento del centro di calcolo mediante virtualizzazione dei server più datati, su cui giravano servizi poco esigenti dal punto di vista delle risorse di calcolo, ma critici relativamente alla disponibilità. Le prime macchine ad essere migrate sono stati i *domain controller* del dominio *Active Directory*, i quali sono responsabili dell'autenticazione centralizzata su tutti i computer *desktop* e *server* della sede, nonché sulle risorse di rete e NAS (*Network Attached Storage*). Poiché i *controller* di dominio gestiscono sia tutte le credenziali di accesso degli utenti che le politiche di autorizzazione (*group policy*), è particolarmente importante che si tratti di *server* dedicati esclusivamente a quel ruolo, senza altri applicativi installati che possano introdurre eventuali falle nella sicurezza. Questa caratteristica rende i *domain controller* dei candidati ideali alla virtualizzazione, in quanto il loro utilizzo non richiede molte risorse, ma è estremamente importante che al momento del bisogno (tipicamente una autenticazione su un sistema appartenente al dominio) essi siano disponibili, pena la negazione dell'accesso. In figura 18 sono riportati i grafici di un mese di utilizzo di memoria e CPU relativi ad uno dei *domain controller* virtualizzati.

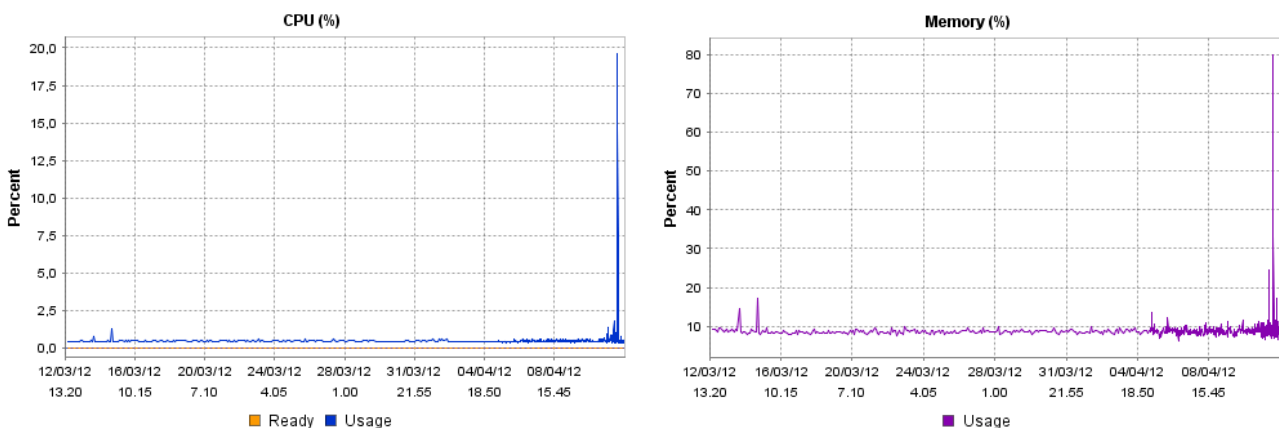


Figura 18. Percentuali di uso di memoria e CPU relative ad un *Active Directory domain controller*. I picchi visibili sono relativi ad operazioni di manutenzione ed aggiornamento *patch* di sicurezza.

Come si può notare, l'impatto sull'*host* fisico che ospita la macchina virtuale è trascurabile; quest'ultima inoltre ora può beneficiare di tutti i vantaggi discussi in precedenza, quali alta affidabilità, *live migration*, ecc.. Discorso analogo è stato fatto per i DNS di dominio, anch'essi parte integrante di *Active*

*Directory* e con i medesimi requisiti di disponibilità ed affidabilità dei *domain controller*, così come per le curve di utilizzo delle risorse virtuali. La CA (*Certification Authority*) non è invece un servizio critico in termini di disponibilità, ma richiede particolare attenzione riguardo la sicurezza della macchina che lo ospita, pertanto anch'esso è stato inserito nel processo di virtualizzazione, così come il *server* di licenze centralizzate di alcuni pacchetti *software* in dotazione alla sezione, come anche diversi server *web*, sia ad uso interno che disponibili su rete pubblica (ad esempio il sito che abbiamo creato per la diffusione del software scientifico prodotto dal nostro gruppo: <http://geosoftware.sci.ingv.it>), ed il *server* interno di aggiornamento dell'antivirus, al quale si collegano i *client* in rete locale evitando di duplicare il traffico su rete pubblica per lo scarico dei medesimi dati.

Emblematico il caso dei due server denominati "Onda" e "Aria" che, sebbene con compiti molto diversi fra loro, hanno beneficiato entrambi del processo di consolidamento. Aria era dedicato al calcolo di mappe giornaliere di diffusione atmosferica (*backtracing*), basate sui campi di vento forniti ogni 6 ore dal NOAA (<http://www.noaa.gov>) e calcolate con origine nei siti di stazioni radionuclidiche a particolare dislocate su tutto il globo. I calcoli intensi duravano mediamente tra le 4 e le 6 ore, per il tempo rimanente la macchina era praticamente inattiva. La virtualizzazione ha così permesso di recuperare il 75% o anche più delle risorse inutilizzate nell'arco delle 24 ore. Onda invece era dedicata alla ricezione di segnali continui in tempo reale, sempre da stazioni site in punti remoti del pianeta, tramite un canale di comunicazione satellitare dedicato. L'utilizzo della macchina era quindi costante, senza interruzioni, ma fondamentalmente solo I/O e con scarso utilizzo di processore e memoria, quindi altro esempio tipico di consolidamento efficiente.

Completano il quadro svariate macchine virtuali dedicate allo sviluppo di applicazioni personalizzate, ai *test* di vari pacchetti *software* in ambienti sia Linux che MS Windows, al monitoraggio e controllo remoto del centro dati, ma anche al calcolo scientifico con simulazioni ed analisi dati, particolarmente utili nei periodi di picco delle attività, in prossimità di convegni e/o pubblicazioni. Due esempi relativi a quest'ultima categoria possono essere individuati da: a) i vari *software* applicativi, molto onerosi dal punto di vista computazionale, prodotti all'interno della nostra unità per l'interpretazione dei campi di potenziale basata su tecniche di inversione [Nicolosi, 2006] [Pignatelli, 2007]; b) il programma GaMField per la generazione di campi magnetici e gravimetrici sintetici, molto utile nella fase di modellazione delle anomalie dei campi di potenziale [Pignatelli, 2007].

Questi sono i casi in cui la versatilità di una infrastruttura di virtualizzazione si rivela particolarmente utile. Durante alcune elaborazioni onerose, può capitare di avere necessità, magari per brevi periodi, di maggiori risorse. Con delle macchine fisiche, questo significa dover procedere all'acquisto dell'*hardware* necessario (RAM, *hard disk*), se non direttamente una nuova macchina più potente, quindi installare il sistema operativo ed il *software* necessario, con ovvie implicazioni sui costi ed i tempi per completare l'operazione. Riallocare invece la risorse ad una macchina virtuale è un'operazione molto semplice e veloce, che si traduce in poche modifiche al file di configurazione associato. Tecniche come la *memory overcommitment*, spiegata nel paragrafo 2.3.2, permettono poi di gestire efficientemente situazioni in cui occorrono più risorse fisiche, ma senza necessariamente dover sostenere delle spese di potenziamento dell'*hardware*.

## 5.1 Analisi del risparmio

Può essere interessante effettuare un confronto fra la situazione del CED prima e dopo il processo di virtualizzazione. A tale scopo si è utilizzato lo strumento software *Dell Data Center Capacity Planner*, disponibile sul sito del fornitore di hardware ([http://www.dell.com/content/topics/topic.aspx/global/products/pedge/topics/en/config\\_calculator?c=us&l=en&s=corp&cs=](http://www.dell.com/content/topics/topic.aspx/global/products/pedge/topics/en/config_calculator?c=us&l=en&s=corp&cs=)) per ottenere stime attendibili dei consumi di corrente, potenza dissipata e quindi calore prodotto, flusso di aria necessaria al raffreddamento, poiché non erano state effettuate misure sperimentali nella fase precedente la virtualizzazione. In figura 19 è riportato il risultato del pianificatore per la configurazione di partenza, mentre in figura 20 è illustrato il medesimo calcolo relativo alla fase in cui i *cluster* VMWare hanno sostituito i vecchi server fisici.

### Rack Power

Name	Location (RU)	Weight (kg)	Thermal (btu/h)	Amps @ 230 volts	Flow Rate (CFM)
McData 4700 @ 40	40	7.20	597.20	0.76	
McData 4700 @ 39	39	7.20	597.20	0.76	
PE750 @ 38	38	13.10	652.00	0.83	30.50
PE750 @ 37	37	13.10	652.00	0.83	30.50
PE750 @ 36	36	13.10	798.60	1.02	30.50
PE750 @ 35	35	13.10	798.60	1.02	30.50
PE750 @ 34	34	13.10	962.20	1.23	30.50
PE750 @ 33	33	13.10	962.20	1.23	30.50
PE1750 @ 32	32	17.50	922.70	1.18	42.40
PE1750 @ 31	31	17.50	922.70	1.18	42.40
PE1950- 3.5 in. drives @ 30	30	20.50	1318.70	1.68	54.30
PE1950- 3.5 in. drives @ 29	29	20.50	1318.70	1.68	54.30
PowerVault 114 @ U27 - U28	U27 - U28	29.00	290.10	0.37	
PE2650 @ U25 - U26	U25 - U26	30.30	1172.10	1.49	50.90
PE2650 @ U23 - U24	U23 - U24	30.30	1125.20	1.43	50.90
PE2850 @ U21 - U22	U21 - U22	30.80	1042.10	1.33	45.80
PE2850 @ U19 - U20	U19 - U20	31.10	1699.20	2.16	45.80
DAE2 @ U16 - U18	U16 - U18	43.20	1284.30	1.64	93.40
DAE2 @ U13 - U15	U13 - U15	43.20	1284.30	1.64	93.40
PE7250 @ U9 - U12	U9 - U12	50.10	1700.00	2.17	152.50
PE7250 @ U5 - U8	U5 - U8	50.10	1700.00	2.17	152.50
CX300 w/UPS @ U1 - U4	U1 - U4	70.50	1769.20	2.25	169.80

Cooling	1.9 Tons Sensible Cooling
SoundPower	10.3 bels A-weighted sound
Power	23570.5 btu/h
Weight	718.0 kg
Voltage	

Configuration Totals: 23570.5 btu/h 718.0 kg 30.03 amps 230 volts 1.9 Tons Sensible Cooling



Figura 19. Stima dei consumi del CED prima del consolidamento.

### Rack Power

Name	Location (RU)	Weight (kg)	Thermal (btu/h)	Amps @ 230 volts	Flow Rate (CFM)
McData 4700 @ 20	20	7.20	597.20	0.76	
McData 4700 @ 19	19	7.20	597.20	0.76	
PE1950- 3.5 in. drives @ 18	18	20.50	1318.70	1.68	54.30
PE1950- 3.5 in. drives @ 17	17	20.50	1318.70	1.68	54.30
DAE2 @ U14 - U16	U14 - U16	43.20	1284.30	1.64	93.40
DAE2 @ U11 - U13	U11 - U13	43.20	1284.30	1.64	93.40
CX300 w/UPS @ U7 - U10	U7 - U10	70.50	1769.20	2.25	169.80
PE R610 @ 6	6	25.90	1105.40	1.41	26.10
PE R610 @ 5	5	25.90	1105.40	1.41	26.10
PE R610 @ 4	4	25.90	1105.40	1.41	26.10
MD3000i @ U1 - U3	U1 - U3	40.30	1405.70	1.79	76.40

Cooling	1.0 Tons Sensible Cooling
SoundPower	8.0 bels A-weighted sound
Power	12892.0 btu/h
Weight	439.4 kg
Voltage	

Configuration Totals: 12892.0 btu/h 439.4 kg 16.42 amps 230 volts 1.0 Tons Sensible Cooling



Figura 20. Stima dei consumi del CED dopo il consolidamento.

Come si può notare dal confronto delle due tabelle, i requisiti di corrente si sono ridotti a circa il 55% del valore iniziale. Di conseguenza anche il calore prodotto, il quale è direttamente proporzionale alla potenza dissipata, che a sua volta è direttamente proporzionale alla corrente assorbita, sperimenta una analoga percentuale di riduzione. Ai valori di corrente sopra citati, in realtà va aggiunto anche il contributo dei condizionatori dedicati al raffreddamento dell'aria nella sala CED, ma anch'essi ora producono un minore consumo in quanto la quantità di calore da smaltire è approssimativamente la metà. Nelle tabelle in figura 19 e 20 la potenza viene riportata in unità del sistema di misura britannico, *BTU/h* (*BTU*=British Thermal Unit), probabilmente in quanto la potenza refrigerante dei condizionatori in commercio viene generalmente espressa in tali unità. Per passare ad unità del Sistema Internazionale si può utilizzare il seguente fattore di conversione: 1 KWatt ~ 3412 *BTU/h*. Nelle due schematizzazioni presenti unitamente alle tabelle, è evidente anche il decremento di numero di *server* utilizzati e spazio fisico associato in termini di unità *rack*. Originariamente tutte le macchine erano alloggiata in un armadio *rack* da 42 unità di capacità totale, di cui 40 occupate (>95%); dopo il consolidamento solamente 20 unità sono occupate (<47%), pertanto sarebbe anche sufficiente un armadio di metà altezza, da 24 unità.

In realtà, ad una analisi più approfondita, i risparmi reali risulterebbero nettamente superiori. Infatti, secondo quanto illustrato nelle figure 14 e 15, i *cluster* attualmente in funzione sono due, ma quello che ospita tutti i servizi virtualizzati descritti nel paragrafo precedente è solo quello di produzione, quindi il confronto andrebbe fatto senza considerare il *cluster* di *test* e sviluppo, né la parte di *storage fibre channel*, bensì solo quanto schematizzato in figura 15 (i consumi degli *switch ethernet* si possono considerare trascurabili rispetto agli altri dispositivi esaminati). In tale scenario, il consumo di corrente si abbatterebbe a circa il 20% del valore iniziale (6.02 A), così come la potenza dissipata (~4724 *BTU/h*). Inoltre, dismettendo l'intero *cluster* più vecchio, si libererebbero ben 14 unità nell'armadio *rack*. Ovviamente, trattandosi di dispositivi ancora perfettamente funzionanti, essi verranno utilizzati proficuamente fino alla fine naturale del loro ciclo di vita.

## 6. Risorse disponibili e sviluppi futuri

Da quanto discusso nei paragrafi precedenti, si può concludere che l'infrastruttura realizzata a Roma2, pur non potendosi ancora definire rigorosamente un *cloud* completo, costituisce uno stadio propedeutico al suo completamento. Ciononostante, qualora se ne presentasse l'esigenza e fatti salvi i criteri di confidenzialità accennati nel capitolo 4, già da ora alcuni servizi tipici del *cloud* (*IaaS* e *PaaS*) potrebbero essere messi a disposizione di altri utenti della sezione, o anche di altre sezioni dell'Ente, anche se in modalità assistita (con intervento manuale dell'amministratore) e non *self-service* ed automatizzata tramite portale.

Il caso più semplice è l'*IaaS*, ovvero fornire intere macchine virtuali, basate sui più comuni sistemi operativi disponibili sul mercato, alle quali l'utente accede in remoto con la facoltà di personalizzazione dell'ambiente ed eventuale installazione autonoma di applicazioni desiderate. Si possono prevedere classi predefinite di macchine virtuali in base alla destinazione (sviluppo, *test*, calcolo), oppure concordare la configurazione di *hardware* virtuale caso per caso.

Come già accennato in precedenza, un esempio di *PaaS* potrebbe essere invece una piattaforma *LAMP* (Linux, Apache, Mysql, Php) già pronta per lo sviluppo di applicazioni *web*, ovvero una macchina basata su sistema operativo *open source* (Linux), con uno dei *web server* più diffusi (Apache), un motore di database relazionale (Mysql) ed un linguaggio di *scripting* lato *server* (Php) per la programmazione delle pagine dinamiche del sito.

Un discorso a parte merita una risorsa molto particolare attualmente disponibile: una istanza di *database* Oracle versione 11g. L'utilità di usufruire di una base dati relazionale per l'archiviazione dati è ampiamente discussa in letteratura, ed esula comunque dallo scopo del presente documento. Ciò che spesso frena l'utente finale dall'adottare tale soluzione è generalmente la complessità di installazione, configurazione e manutenzione di un *RDMBS* (*Relational DataBase Management System*) come Oracle. Usfruendo di questo servizio già pronto, all'utente verrebbero assegnate credenziali di accesso ed uno spazio in cui creare le proprie tabelle per caricare e manipolare dati a piacimento, senza doversi preoccupare di tutta la complessa parte amministrativa del *database*.

Come possibili sviluppi futuri, si potrebbe completare la realizzazione dell'infrastruttura *cloud*, contemplando così anche l'automatizzazione dei processi e la contabilizzazione delle risorse usate. Questo passaggio potrebbe porre le basi per una nuova concezione di utilizzo delle infrastrutture informatiche,

eventualmente estendibile in seguito anche all'intero Istituto. Come esempio interessante si può considerare quello di alcune grandi realtà del mondo dell'impresa, le quali già oggi stanno cambiando il loro modo di concepire il ruolo del dipartimento IT interno, trasformando quest'ultimo da centro di costo a fornitore di servizi informatici per le altre strutture aziendali, alla stessa stregua di un *provider* esterno, che viene finanziato sulla base dei servizi offerti. Il *private cloud* si adatta bene a questa nuova idea, perché permette di usufruire di tutti i benefici fin qui discussi (ottimizzazione delle risorse, maggiore economicità, niente carico di gestione delle infrastrutture per gli utenti finali, possibilità di rendicontare il livello utilizzo dei sistemi, ecc.), mitigando così il frazionamento o la duplicazione delle risorse.



## **Bibliografia**

Carl A. Waldspurger, (2002). Memory Resource Management in VMware ESX Server. Proceeding of the fifth Symposium on Operating System Design and Implementation, Boston, Dec 2002.

Gartner (2008). Press Release (<http://www.gartner.com/it/page.jsp?id=707508>), June 2008.

Mell P., Grance T., (2011). The NIST Definition of Cloud Computing. Retrieved 24 July 2011. National Institute of Standards and Technology. (<http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf>)

Nicolosi I., Blanco-Montenegro I., Pignatelli A., Chiappini M., (2006). Estimating the magnetization direction of crustal structures by means of an equivalent source algorithm. *Physics of the Earth and Planetary Interiors*, 155, 163-169, 2006.

Pignatelli A., Nicolosi I., Chiappini M., (2007). An alternative 3-D source inversion of magnetic anomalies method with depth resolution. *Annals of Geophysics*, 169, 1021-1027, 2007.

Pignatelli A., Nicolosi I., Carluccio R., Chiappini M., Von Frese R., (2011). Graphical interactive generation of gravity and magnetic fields. *Computers & Geosciences*, 37, 567-572, 2011.

VMWare Inc., (2009). Resource Management Guide, Revision: 20090116, Item: VI-ENG-Q407-438

Voorsluys, William; Broberg, James; Buyya, Rajkumar, (2011). Introduction to Cloud Computing. In R. Buyya, J. Broberg, A.Goscinski. *Cloud Computing: Principles and Paradigms*. New York, USA: Wiley Press. pp. 1-44. ISBN 978-0470887998.

([http://media.johnwiley.com.au/product\\_data/excerpt/90/04708879/0470887990-180.pdf](http://media.johnwiley.com.au/product_data/excerpt/90/04708879/0470887990-180.pdf))



**Coordinamento editoriale e impaginazione**

Centro Editoriale Nazionale | INGV

**Progetto grafico e redazionale**

Daniela Riposati | Laboratorio Grafica e Immagini | INGV

© 2012 INGV Istituto Nazionale di Geofisica e Vulcanologia

Via di Vigna Murata, 605

00143 Roma

Tel. +39 06518601 Fax +39 065041181

**<http://www.ingv.it>**



**Istituto Nazionale di Geofisica e Vulcanologia**