Università degli Studi di Padova

Facoltà di Ingegneria

Corso di Laurea in Ingegneria delle Telecomunicazioni

Tesi di Laurea

SVILUPPO E CONSOLIDAMENTO DELL'INFRASTRUTTURA DEI SERVER APPLICATIVI AZIENDALI TRAMITE VIRTUALIZZAZIONE DELLE RISORSE

Candidato:
Alessandro Rossi

Relatore:
Prof. Nevio Benvenuto

Tutor aziendale:
Ing. Mirco Tamborra

a.a. 2009/2010

Indice

INTRODUZIONE

I – L'AZIENDA

DBA LAB S.p.A. è Internet Content Provider di seconda generazione e opera nel settore dell'Innovation & Comunication Technology (ICT). Sviluppa software e applicativi web based per il Project e il Facility Management, soluzioni di Document Management e sistemi di IT Security , portali verticali (Vortal) e contenuti digitali.

DBA LAB S.p.A. nasce su iniziativa della Holding DBA Group S.p.A. di Villorba (TV), attiva in Italia e all'Estero nel mercato dell'Ingegneria e del Project Management attraverso le proprie società operative; in particolare la DBA Progetti S.p.A., con sede principale a Villorba ma con altre cinque sezioni operative in altrettante sedi dislocate sul territorio nazionale: Milano, Roma, L'Aquila e Santo Stefano di Cadore (BL).

Nel 2005 DBA LAB S.p.A. acquisisce l'intero capitale della società padovana VENICEPLAZA S.p.A., presente dal 2000 nelle regioni del Nord Est in qualità di Internet Service Provider. Con l'incorporazione di VENICEPLAZA S.p.A., DBA LAB S.p.A. ha perfezionato il proprio *know-how* nello sviluppo di applicazioni software basate su protocolli internet, ha disposto di un marchio già noto e di un parco clienti attivo e si è posizionata rapidamente tra le principali Software House venete operanti nello sviluppo di servizi innovativi forniti via web.

DBA LAB S.p.A. fornisce, oggi, attraverso il marchio Veniceplaza, servizi e prodotti web based per le Piccole e Medie Imprese (PMI) e per gli Enti Locali.

DBA Progetti S.p.A., dal canto suo, si occupa di attività ingegneristiche e architetturali in svariati settori che spaziano dalle reti di comunicazioni fisse e mobili, alle reti di distribuzione di carburanti e all'urbanistica, la mobilità e l'ambiente.

II – PREMESSA ALLO SVOLGIMENTO DEL TIROCINIO

Ciò che accomuna queste società, all'apparenza così diverse tra loro, è l'esigenza di avere un luogo sicuro, protetto e accessibile solo alle persone autorizzate per salvare e all'occorrenza poter utilizzare i propri dati e le proprie attività, siano essi progetti di Autocad o siti internet HTML, ormai nella quasi totalità dei casi di natura telematica.

Risulta così possibile svolgere un'attività dal proprio ufficio lavorando su dei progetti in tempo reale con altre persone, siano esse locate nell'ufficio a fianco o in una sede remota, distante anche migliaia di chilometri dal proprio posto di lavoro. E tutto questo con due garanzie basilari: la sicurezza di non perdere il proprio lavoro (per guasti, manomissioni, etc.) e la sicurezza di non essere spiati da persone non autorizzate o, peggio, malintenzionate.

Per svolgere tutto questo nel migliore dei modi è stato creato un datacenter, il cuore pulsante della DBA LAB S.p.A. e della DBA PROGETTI S.p.A. dislocato presso la loro sede principale di Villorba.

Nel datacenter ci sono sistemi informatici di varia natura: i server, con il compito di ospitare fisicamente le risorse, sia quelle interne dell'azienda che quelle relative ai propri clienti, le SAN (Storage Area Network), unità di memoria atte ad ospitare grandi moli di dati, utilizzate soprattutto per creare backup del lavoro svolto e presente nei server, gli switch e i router, con il compito di mettere in collegamento tutte le risorse presenti nei server con la rete aziendale, un modem MPLS che mette in collegamento la sede di Villorba con le altre sedi dislocate nel territorio nazionale per mezzo di un canale dedicato e infine, protetto da firewall, il collegamento verso la rete internet, attuata per mezzo di un media converter ASCO TLC.

Questa tesi nasce dal tirocinio formativo svoltosi nella divisione Tecnico-Amministrativa di DBA LAB S.p.A., e consiste essenzialmente in una relazione di qualità che studia i pregi e difetti di una tecnologia che l'azienda, dopo dovute e

attente riflessioni, ha intenzione di instaurare all'interno del proprio datacenter: la virtualizzazione dei server.

III – IL LAVORO SVOLTO

Il mio percorso in DBA LAB S.p.A. è iniziato con la creazione di un database in Access con il quale ho catalogato tutti i server presenti nel datacenter, con funzionalità aggiuntive quali la creazione di report ad hoc per monitorare la situazione in base a parametri prestabiliti, come ad esempio le risorse CPU di cui dispongono, lo spazio di memorizzazione ancora utilizzabile al loro interno o la funzione che rivestono all'interno della rete.

Successivamente ho creato un documento, inserito nell'Appendice A di questo testo, denominato "Struttura Datacenter", con il diagramma della rete DBA e la composizione del layout e dei rack del datacenter stesso, utile da un lato ai tecnici e ai nuovi amministratori di rete che devono intervenire al suo interno, e dall'altro ai clienti in visita per dare una visione d'insieme della rete e delle macchine a disposizione dell'azienda.

Infine, come ultimo e più importante compito, ho redatto questo documento, per esporre ai responsabili tecnici cosa è di fatto la virtualizzazione dei server, come lavora e che vantaggi si possono trarre da una sua implementazione.

Lo scopo principale di un azienda del settore ICT che fornisce, tra le altre cose, un servizio di gestione di siti web sui propri server per poi renderli accessibili alla rete internet, è raggiungere standard qualitativi elevati cercando nel contempo di contenere i costi. Si deve quindi fare in modo che gli apparati di rete siano:

- **Sempre disponibili**. Obiettivo degli amministratori di rete è puntare ad un *uptime* dei server prossimo al 100%, che siano cioè sempre in funzione.
- **Sicuri per i dati**. La sicurezza dei dati è un requisito fondamentale per ogni server. L'unità di storage deve essere in grado di conservare, organizzare e rendere disponibili i dati che contiene; ad essa deve essere poi collegato un sistema di backup, che effettua la memorizzazione periodica dei dati in una unità di storage separata e, se possibile, un

disaster recovery, strumento di ripristino dei dati in caso di malfunzionamenti improvvisi.

- **Sicuri per le applicazioni**. Le applicazioni che forniscono servizi devono essere indipendenti tra loro. Ciò non toglie il fatto che debbano cooperare, ma il funzionamento di una o più di esse non deve influire sulle altre.
- **Sicuri per la rete**. Il sistema di sicurezza installato deve fare in modo che solo gli utenti accreditati, stabiliti dall'amministratore di rete, possano usufruire dei servizi, evitando quindi non solo accessi accidentali ad aree riservate, ma veri e propri attacchi informatici.
- **Ottimizzati per l'esecuzione dei servizi**. I servizi devono poter disporre delle risorse massime consentitegli. Si deve evitare di contro che ci siano momenti di picco non supportati dai server uniti a momenti di inattività, ma si deve invece puntare ad una distribuzione omogenea dei carichi di lavoro.
- **Utilizzo ottimale delle risorse**. Lo sfruttamento dell'hardware sottostante ottimizza gli investimenti effettuati e consente di risparmiare sull'acquisto di ulteriori componenti.

Con questo documento si vuole stabilire se la virtualizzazione dei server è in grado di assolvere questi punti fondamentali.

IV – COMPOSIZIONE DEL DOCUMENTO

Nel capitolo 1 si vedrà la virtualizzazione nel senso più generale del termine, facendo un lungo excursus storico partendo dalla sua nascita e giungendo fino ai giorni nostri, per scoprire in dettaglio le varie fasi dello sviluppo tecnologico che l'hanno portata a come la conosciamo noi oggi. Il capitolo si concluderà poi con una definizione formale del termine, in modo da avere una prima visione generale dei prodotti che possono o meno far parte di questa categoria.

Nel capitolo 2 si entrerà in dettaglio nel tema della virtualizzazione dei server, iniziando col capire più in profondità i reali benefici dell'utilizzo di questa tecnologia per poi passare ad argomentazioni tecniche che spiegheranno come

funzionano le moderne tecnologie di virtualizzazione basate sul software, la relazione che intercorre tra il processore e il sistema operativo e si capirà come lavora e come si progetta la VMM (Virtual Machine Monitor), il gestore delle macchine virtuali. Infine si terminerà il capitolo facendo un riassunto dei tipi di virtualizzazione dei server esistenti, vedendone i pregi e i difetti.

CAPITOLO 1

INTRODUZIONE ALLA VIRTUALIZZAZIONE

La virtualizzazione è una termine che negli ultimi anni ha guadagnato grande popolarità tra i professionisti e responsabili IT. Promettendo di ridurre la sempre crescente infrastruttura all'interno dei datacenter, le tecnologie di virtualizzazione sono sorte in decine di società di hardware e software.

La virtualizzazione è utilizzata ormai da più di tre decenni. Una volta accessibile solo dalla grande, ricca e prosperosa impresa, la sua tecnologia è ora disponibile in ogni aspetto dell'informatica a costi molto più contenuti. In numerosi casi è disponibile addirittura gratuitamente (grazie a iniziative *open-source*) o inclusa nel prezzo dei prodotti che la implementano, quali ad esempio software di sistemi operativi o hardware di memorizzazione/archiviazione e di telecomunicazioni.

Comprendere la tecnologia e il carico di lavoro che è possibile eseguire in un ambiente virtualizzato è fondamentale per ogni amministratore e architetto di sistemi che vuole offrire tutti i vantaggi che ne derivano alla propria organizzazione o ai propri clienti.

In particolare, a questa categoria fanno parte tutti quei prodotti e strumenti che rispondono, in parte o integralmente, ai seguenti punti chiave:

- Aggiungere un livello di astrazione tra le applicazioni e l'hardware;
- Attivare una riduzione dei costi e della complessità;

- Garantire l'isolamento delle risorse informatiche per una migliore affidabilità e sicurezza;
- Migliorare i livelli e la qualità del servizio;
- Allineare in modo più efficace i processi IT agli obiettivi aziendali;
- Eliminare le ridondanze e massimizzare l'utilizzo delle infrastrutture IT.

Anche se la forma più comune di virtualizzazione, e quella a cui faremo riferimento in questo documento, è focalizzata sull'hardware delle piattaforme server, questi obiettivi, e le tecnologie che li supportano, si sono sviluppati anche in altri componenti critici (e costosi) dei moderni datacenter, incluse le unità di storage e le infrastrutture di rete.

Ma per capire in modo approfondito cos'è la virtualizzazione è necessario fare un excursus sulla sua storia e le sue origini.

1.1 – LA STORIA DELLA VIRTUALIZZAZIONE

La prima forma di virtualizzazione, concepita negli anni '60 del secolo scorso, prese piede col nome di *time sharing* ("condivisione di tempo"). Christopher Strachey, primo professore di Calcolo all'Università di Oxford e leader del Programming Research Group, portò alla luce questo termine nel suo scritto *Time Sharing in Large Fast Computers*. Strachey lo usò quale estensione della "multiprogrammazione", l'esecuzione di più processi sullo stesso sistema contemporaneamente, sfruttando i momenti di inattività del processore. In un sistema multiutente avrebbe permesso a più programmatori di compilare ed eseguire in modo contemporaneo i loro programmi interagendo con il sistema centralizzato ciascuno con il proprio terminale. Dal momento che i primi computer mainframe erano estremamente costosi, non era possibile garantirne l'accesso esclusivo ad un singolo utilizzatore, ma con questa soluzione di gestione delle richieste multiutente da parte della CPU (detta *context switch*) si dava l'impressione ad ognuno di avere a disposizione il computer centrale interamente per sé.

La multiprogrammazione, come diverse altre idee rivoluzionarie, cominciarono così a guidare l'innovazione, scaturendo in una serie di computer che irruppero con forza sulla scena. Due in particolare sono considerati parte integrante della linea evolutiva di virtualizzazione come la conosciamo noi oggi, l'Atlas e l'IBM M44/44X.

1.1.1 – I PRIMI PROGETTI

Il primo di essi, progettato e gestito dal Dipartimento di Ingegneria Elettrica dell'Università di Manchester, il supercomputer Atlas, approfittò dei concetti di *time-sharing*, multiprogrammazione e controllo periferico condiviso per fornire ai suoi utilizzatori una velocità che nessun altro mainframe di quegli anni era in grado di dare. La gestione dei processi venne suddivisa tra due componenti in base alla loro natura: un componente si occupava dell'esecuzione dei programmi utente mentre l'altro, il *supervisor*, gestiva i processi del sistema operativo. Quest'ultimo gestiva anche le risorse chiave, come ad esempio il tempo di elaborazione del computer, e furono implementate istruzioni speciali, dette *extracodes*, per aiutarlo a disporre e a gestire l'ambiente di calcolo per le istruzioni dei programmi utente. In sostanza, questa fu la nascita dell'*hypervisor*, il controllore delle macchine virtuali.

Atlas introdusse anche il concetto di memoria virtuale, che in origine venne chiamata *one-level store* (memoria ad un livello) a causa del fatto che, sebbene la memoria in uso si collocava su unità a più livelli all'interno della gerarchia di memoria, l'utente poteva vedere tutta la memoria di cui disponeva ad un unico livello di accessibilità. Ed infine, ma non ultime di importanza, le tecniche di *paging* per la memoria di sistema, creando un nucleo di memoria separato logicamente dalla memoria usata dai programmi utente, sebbene le due parti fossero integrate nello stesso supporto fisico. Sotto molti aspetti fu il primo passo verso la creazione di uno strato di astrazione che hanno in comune tutte le tecnologie di virtualizzazione.

La risposta di IBM, che intendeva mantenere il proprio titolo di principale innovatore di computer, fu il Progetto M44/44X, sviluppato presso il Centro di

Ricerca Thomas J. Watson di Yorktown, New York. Venne creata una architettura simile a quella dell'Atlas, nella quale venne usato per la prima volta il termine *virtual machines* e diventò il contributo principale di IBM agli emergenti sistemi di time sharing. La macchina principale era un computer scientifico IBM 7044 (M44) con diverse macchine virtuali 7044, o 44X, simulate utilizzando sia hardware (la memoria virtuale) che software (la multiprogrammazione).

Negli anni successivi perfezionò le tecnologie dei suoi sistemi, introducendo nell'IBM 7094 il *Compatible Time Sharing System* (CTSS), sistema di time sharing compatibile con lo standard di elaborazione *batch* del sistema operativo utilizzato dalla macchina, il *Fortran Monitor System* (FMS), che consentiva di eseguire grossi carichi di lavoro tipicamente non interattivi (i jobs) in modo pianificato. CTSS non solo eseguì una copia di FMS nel computer main 7094, utilizzato come strumento primario per il flusso dello standard batch, ma ne avviò una copia in ogni macchina virtuale collegata ad esso. In questo modo i *jobs* in *background* avrebbero potuto accedere a tutte le periferiche quali nastri, stampanti, lettori di schede perforate e display grafici al pari dei jobs FMS in *foreground*, purché non interferissero con l'esecuzione di quest'ultimi o con qualsiasi altra risorsa collegata ad essi.

Alla fine del 1960 un gruppo di ricercatori IBM del Cambridge Scientific Center (CSC), guidati da Norm Rassmussen e Bob Creasy, svilupparono con successo il primo sistema operativo per macchina virtuale basato su hardware completamente virtualizzato, il CP-40, precursore del popolare VM/370; quest'ultimo, rilasciato nel 1972, grazie al componente *Virtual Machine Monitor* (VMM, il controllore della macchina virtuale) avviato su hardware reale era in grado di eseguire numerose macchine virtuali indipendenti su copie virtuali di hardware. Ogni macchina virtuale del VM/370 era in grado di eseguire un'unica installazione del sistema operativo IBM in modo stabile, indipendente e con elevate prestazioni.

Un aspetto interessante del progetto fu lo sviluppo di un sistema operativo monoutente che procedette in parallelo con lo sviluppo dell'ambiente CP. Questo sistema operativo, chiamato Conversational Monitor System (CMS), eseguiva tutte le applicazioni in uno "pseudo-supervisore" che diede la possibilità agli utenti di eseguire qualsiasi tipo di istruzione, sia le istruzioni utente (con

indirizzamento virtuale) che le istruzioni "privilegiate" (con indirizzamento fisico). A quest'ultimo gruppo fanno parte tutte le istruzioni che vengono eseguite solo dal sistema operativo poiché, se lanciate su un sistema stand-alone nativo o a macchine virtuali standard in modo errato, sono in grado di intaccare (e spesso danneggiare) le risorse hardware della macchina; per questo motivo, se non espressamente autorizzate, vengono segnalate e bloccate dal sistema operativo stesso mediante un "trap". Questo fatto aumentò molto la sicurezza poiché sia il CMS che le applicazioni utente venivano eseguiti su una macchina virtuale che veniva protetta, attraverso il CP, da qualunque fattore che poteva colpire direttamente il funzionamento dell'hardware reale. La possibilità di utilizzare le istruzioni privilegiate nel loro codice, consentì agli utenti di sperimentare molti aspetti delle risorse delle loro macchine, che prima d'ora non era permesso loro di fare.

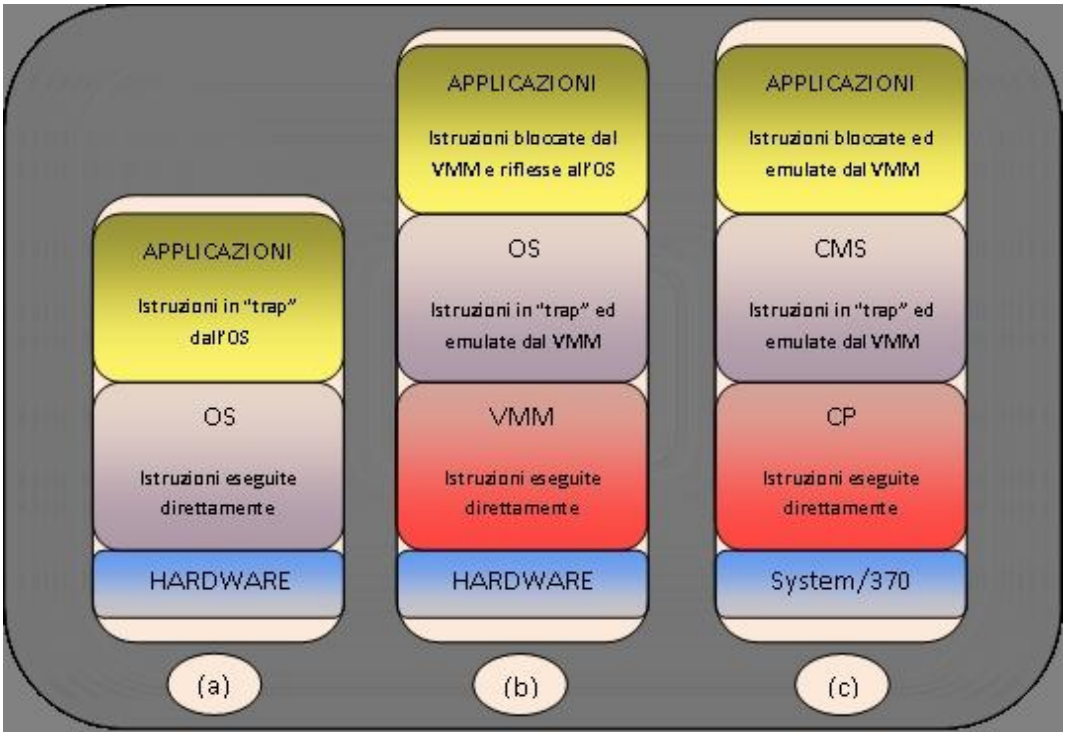


Fig. 1.1 – Esecuzione delle istruzioni "privilegiate" nei diversi tipi di struttura: (a) Macchina Stand-Alone standard; (b) Macchina Virtuale; (c) Sistema CP/CMS.

1.1.2 – L'ESPLOSIONE DELLA VIRTUALIZZAZIONE

Fino ad ora si è potuto vedere come, nell'arco degli anni, ci furono numerosi tentativi di virtualizzare e semplificare i datacenter non solo attraverso la vera virtualizzazione, definita dalla precedente serie di obiettivi, ma anche per mezzo della condivisione e il consolidamento dell'infrastruttura, che ha portato, a parità di carico di lavoro:

- Una diminuzione dei server da gestire;
- Una diminuzione dello spazio necessario ad ospitarli;
- Un cablaggio minore, per cui gestibile con più facilità;
- Un maggiore risparmio energetico.

Fino alla fine degli anni '80 del secolo scorso l'uso delle tecniche di virtualizzazione era ancora limitato e veniva impiegato a livello locale per suddividere tra più utenti le risorse di un mainframe centrale grazie alle varie tecniche di multiprogrammazione, e nell'ambito delle telecomunicazioni era affermato un paradigma centralizzato: internet con il passare degli anni si stava diffondendo, ma solo le grandi organizzazioni erano in grado di sostenere gli alti costi necessari alla messa in opera di tecnologie virtualizzate.

Dalla metà degli anni '90 e fino ai giorni nostri, un insieme di fattori hanno cambiato il modo di concepire i datacenter, cambiando radicalmente i paradigmi organizzativi e architetturali degli stessi.

Si è passati così da un modello centralizzato fruibile soltanto ad un ristretto numero di grandi organizzazioni ad uno decentralizzato accessibile anche alla piccola e media impresa, ridimensionando le infrastrutture in una struttura orizzontale.

Il motivo scatenante che ha portato a questa mutazione del settore ITC si può identificare con la crescita su scala mondiale di internet, grazie ad un miglioramento delle tecnologie disponibili e ad un abbassamento dei loro costi. L'uso di questo nuovo mezzo di informazione si è ingrandito a tal punto da dar vita ad una reazione a catena; da una parte le aziende e i privati, fino ad allora

estranei al mondo telematico, hanno iniziato a trovare in internet un nuovo spazio dove poter offrire i propri servizi o dove pubblicizzarsi. Dall'altro la nascita di un grandissimo numero di piccole e medie imprese, in grado di rispondere a questo nuovo tipo di clientela. È la nascita della New Economy, che ha avuto grande successo in poco tempo perché in grado di offrire la possibilità di operare in un mercato globale abbattendo i costi di gestione e di non vincolare la propria clientela ad uno spazio definito quale può essere la sede fisica di una società o di un esercizio commerciale.

Ed i pilastri della New Economy furono le imprese in grado di offrire la creazione e la gestione di siti internet ai prezzi più accessibili.

La richiesta era giunta ad un punto tale che uno sviluppo orizzontale era diventato inevitabile. Ci si trovò di fronte ad una "la proliferazione dei server"; invece dell'acquisto e del mantenimento di un unico host fisico e delle relative periferiche necessarie per ogni applicazione, si giunse alla conclusione che ad ognuna di tali applicazioni era possibile dare il suo ambiente operativo, completo di I/O, potenza di elaborazione e memoria, avendo in condivisione l'hardware rimanente. Così, mentre la quantità di applicazioni e ambienti applicativi dispiegati aumentava, anche il numero di server implementati all'interno dei datacenter cresceva a ritmi esponenziali. I server centralizzati erano visti come troppo costosi da acquistare e mantenere per le molte aziende non ancora fondate su tali piattaforme informatiche e, mentre i big frame, server dagli enormi carichi di lavoro delle grandi imprese continuavano a sopravvivere, il mercato dei server di fascia media e bassa era alla ricerca di nuova linfa e opportunità.

A colpo d'occhio la decentralizzazione offre due benefici principali: la sicurezza e la stabilità, due fattori interconnessi allo stesso meccanismo di fondo. La stabilità proviene dal mantenimento globale dell'infrastruttura in modo semplice ed efficace, grazie alle patch e agli aggiornamenti periodici facilmente applicabili senza interferire con gli altri sistemi in esecuzione. Per lo stesso motivo, la decentralizzazione ha favorito la sicurezza poiché un sistema compromesso è fisicamente isolato dagli altri sistemi della rete.

In aggiunta, con la popolarità di Windows e delle piattaforme distribuite con i più leggeri sistemi *open source*, la promessa che molti speravano di ottenere

includeva un migliore ritorno di investimento sulle attività e un minore costo totale di proprietà (TCO). La mercificazione delle piattaforme hardware e software a buon mercato ha aggiunto ulteriore carburante allo sviluppo di quel paradigma.

1.1.3 – IL RITORNO ALLA CENTRALIZZAZIONE

Tuttavia le imprese si resero presto conto che il ridimensionamento orizzontale necessario per nutrire le nuove istanze del server, e la conseguente necessità di spazio e componenti per supportare tale espansione contrastava con gli ideali da cui erano partiti. La proliferazione dei server si è intensificata principalmente con la domanda di spazio web, ma non solo; si è manifestata la necessità di effettuare iterazioni multiple della stessa applicazione per sostenere lo sviluppo dell'SDLC (*software development life cycle*). Il costo globale di sviluppo e produzione delle applicazioni è giunto ad un maggior consumo di potenza, un minor spazio fisico disponibile e un maggiore *effort* di gestione il quale, unito agli altri fattori, conta decine (se non centinaia) di migliaia di dollari in costi di manutenzione annuali per macchina (fonte: VIrtualization with Xen). In aggiunta a questa gestione e manutenzione globale, la decentralizzazione ha diminuito l'efficienza delle macchine, lasciando la media di inattività del server ad una percentuale dell' 85-90 per cento. Queste inefficienze hanno eroso ulteriormente qualunque potenziale risparmio sul costo del lavoro promessi dalla decentralizzazione.

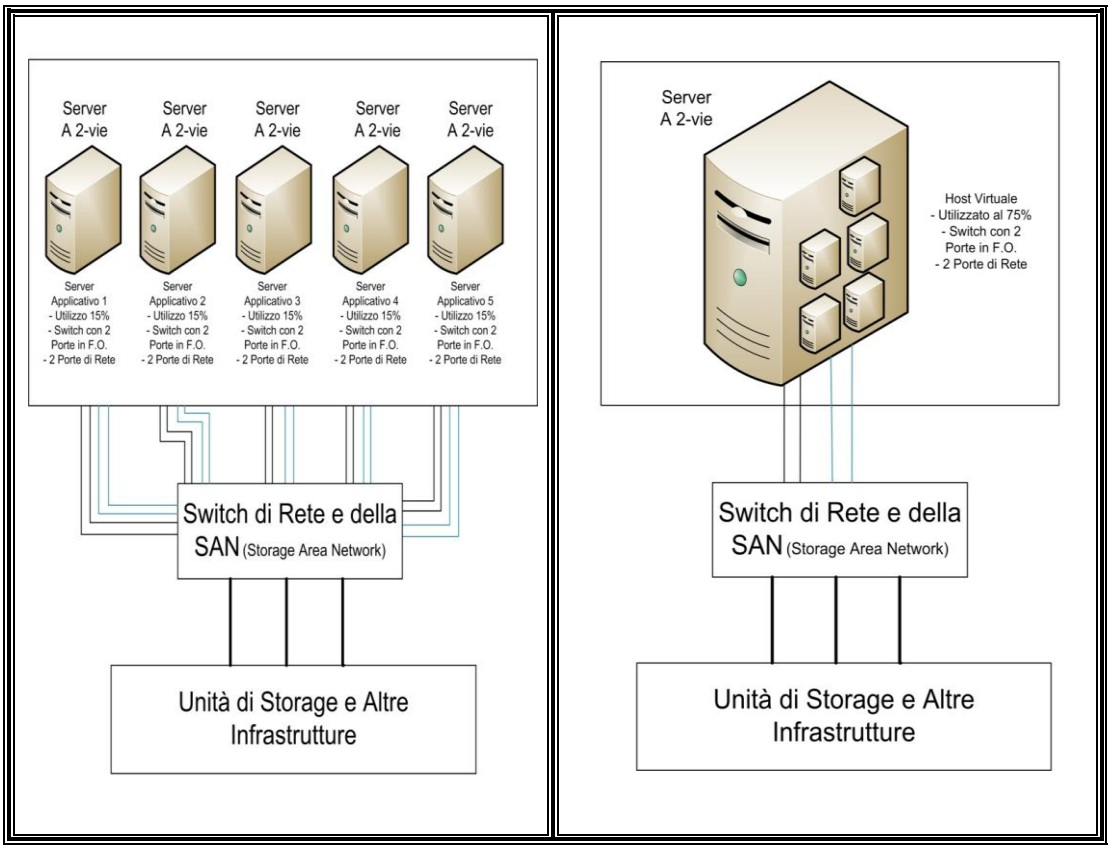


Fig. 1.2 – Paradigma di Decentralizzazione e Centralizzazione a confronto.

Così, contro ogni previsione, negli ultimi anni si sta ritornando al paradigma centralizzato. La situazione è molto diversa rispetto a trent'anni fa, sono diversi gli ambiti di utilizzo, i costi, le tecnologie ed anche i motivi che hanno spinto le aziende a questo ritorno; ma guardando la situazione dall'esterno ci si rende conto che la decentralizzazione è un capitolo che a poco a poco si sta chiudendo.

1.2 – CONSIDERAZIONI FINALI

In conclusione la virtualizzazione può essere definita come una struttura informatica in cui si pratica un metodo di divisione delle risorse hardware in ambienti multipli di esecuzione, applicando uno o più concetti o tecnologie quali il partizionamento hardware e software, il time sharing, la simulazione parziale o totale della macchina, l'emulazione, la qualità del servizio, e molti altri.

Proprio come si fece durante la fine degli anni ‘60 e gli inizi degli anni ‘70 con i primi VM/370 di IBM, la moderna virtualizzazione consente a più istanze del sistema operativo di essere eseguite simultaneamente su un singolo computer, anche se in modo molto più economico rispetto a quel periodo. Ognuna di tali istanze condivide le risorse disponibili sull’ hardware fisico comune, come illustrato in Fig 1.3. Un programma, denominato Virtual Machine Monitor (VMM), controlla l'uso e l'accesso alla CPU, alla memoria, alle unità di storage e alle risorse di rete presenti al livello sottostante.

Fig. 1.3 – Le macchine virtuali girano al di sopra dell’hardware fisico

CAPITOLO 2

I MECCANISMI DELLA VIRTUALIZZAZIONE

2.1 – BENEFICI PRINCIPALI

La virtualizzazione ricopre un ruolo fondamentale nell'ottimizzazione dei sistemi. Sebbene a prima vista potrebbe semplicemente apparire come un modo per ridurre e semplificare l'infrastruttura dei server, può invece essere uno strumento per trasformare il modo di concepire il datacenter nel suo complesso. In Fig. 2.1 si illustra il modello di ottimizzazione dei sistemi. Il consolidamento fisico è la base necessaria per poter affrontare gli step successivi, che sono il consolidamento logico e la razionalizzazione complessiva dei sistemi e delle applicazioni attraverso l'identificazione di applicazioni che sono inutili o ridondanti e possono quindi essere eliminate.

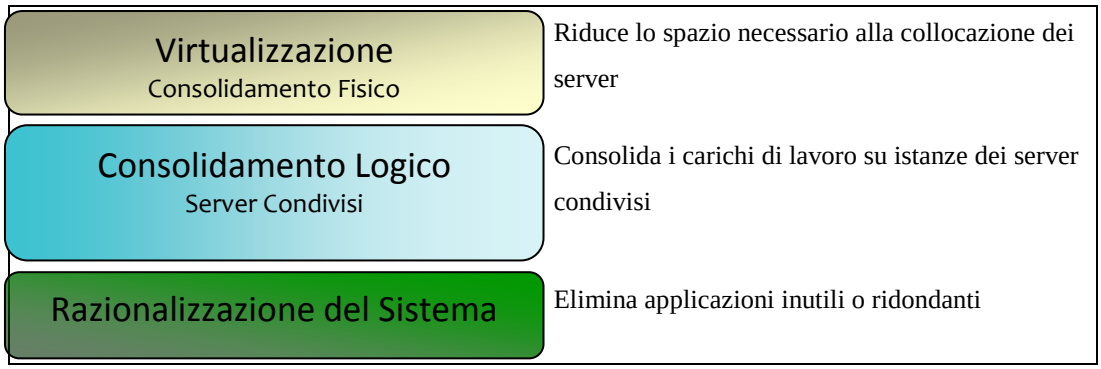


Fig. 2.1 – Ruoli della virtualizzazione nell'ottimizzazione dei server

I benefici che giustificano lo spostamento di un organizzazione IT verso un'infrastruttura virtuale si possono semplificare in tre concetti principali:

- **Consolidamento.** Aumenta lo sfruttamento dei server, semplifica la migrazione dei software *legacy* (programmi sviluppati anni addietro con le tecnologie disponibili all'epoca che, sebbene abbiano subìto interventi di

manutenzione, non sono mai stati rimpiazzati), ospita sistemi operativi misti per piattaforme fisiche e semplifica gli ambienti di sviluppo e di test.

- **Affidabilità.** Isola i difetti e i guasti causati dal software, rialloca le partizioni esistenti e crea partizioni di *failover*, all'occorrenza o dedicate (sistema di salvataggio nel quale le funzioni di un componente di sistema, in questo caso la partizione, vengono inviate ad un secondo componente quando si verifica un problema).
- **Sicurezza.** Contiene gli attacchi digitali attraverso l'isolamento dei guasti e dei *bugs* e applica differenti impostazioni di sicurezza ad ogni partizione.

2.1.1 – Consolidamento

L'obiettivo alle spalle della tecnologia di consolidamento si può riassumere con i termini *aggregare e unificare*. Nel caso della virtualizzazione, i carichi di lavoro vengono concentrati in un minor numero di piattaforme fisiche in grado di sostenere la loro richiesta di risorse informatiche quali CPU, memoria e I / O. Nei moderni datacenter molti carichi di lavoro sono ben lontani dallo sfruttare appieno l'hardware su cui girano, con conseguente spreco delle infrastrutture e abbassamento dei profitti. Attraverso il consolidamento, la virtualizzazione permette di concentrare in modo strategico le istanze del server o dei sistemi operativi e dei rispettivi carichi di lavoro e li pone su hardware condiviso con una disponibilità di risorse sufficienti a soddisfarne la domanda. In passato si credeva che i server non dovessero essere costretti a lavorare presso il loro limite di capacità; è invece vero il contrario. Al fine di massimizzare l'investimento, i server devono lavorare il più possibile vicino al loro carico massimo, senza però compromettere le prestazioni dei carichi di lavoro o dei processi aziendali in esecuzione. Seguendo una corretta pianificazione e comprendendo quanto incidono tali processi, la virtualizzazione contribuisce all'incremento dell'utilizzo dei server riducendo nel contempo il numero di piattaforme fisiche necessarie.

Un altro vantaggio del consolidamento si concentra sulle migrazioni dei sistemi *legacy*. L'hardware dei server si è sviluppato a livelli tali che spesso è diventato incompatibile con i sistemi operativi e le applicazioni *legacy*. La ricerca di

tecnologie di processori più recenti, dei relativi chipset supportati e dei bus ad alta velocità può, il più delle volte, paralizzare i sistemi *legacy*, se non addirittura renderli inutilizzabili.

La virtualizzazione consente la migrazione di questi sistemi in modo semplice, fornendo una piattaforma comune e largamente compatibile su cui possono essere eseguite le loro istanze.

In passato i sistemi operativi erano legati ad una specifica piattaforma hardware. Questo fatto legava le mani a molte organizzazioni, costringendole a fare grandi investimenti in hardware al fine di mantenere le loro applicazioni di business critiche. Tuttavia, grazie alla mercificazione dell'hardware, molti dei più comuni sistemi operativi attualmente disponibili possono essere eseguiti su una vasta gamma di architetture server, il più popolare dei quali è l'architettura x86. In essa è possibile eseguire Windows, UNIX o una delle distribuzioni Linux a propria scelta. Le tecnologie di virtualizzazione costruite sopra all'architettura x86 possono ospitare anche ambienti eterogenei. Sistemi operativi multipli, compresi quelli menzionati precedentemente, possono essere consolidati sullo stesso hardware fisico, riducendo ulteriormente i costi di acquisizione e di manutenzione.

Infine, gli effort di consolidamento snelliscono gli ambienti di sviluppo e di test.

Invece di avere una proliferazione incontrollata dell'infrastruttura, dovuta alla nascita di nuovi progetti o nuovi rilasci o al mantenimento di applicazioni esistenti, la virtualizzazione consente di consolidare molti di quei carichi di lavoro su un minor numero di server fisici.

2.1.2 – Affidabilità

Ora più che mai, l'affidabilità è diventata un punto cardine per le organizzazioni IT. Essa ha una relazione diretta con la disponibilità del sistema, i tempi di uptime delle applicazioni e, di conseguenza, con i redditi generati. Le società sono disposte a investire pesantemente nelle proprie infrastrutture server per assicurare che le loro applicazioni *line-of-business* critiche restino on-line e che, quindi, il loro funzionamento non sia mai interrotto. Investendo in hardware e software

aggiuntivi per monitorare gli errori e i guasti del software, le infrastrutture sono atte a gestire guasti e tempi di inattività non programmati senza interruzione.
Così facendo, però, i costi aumentano a dismisura.
Le tecnologie di virtualizzazione sono predisposte ad affrontare questa situazione, fornendo un forte isolamento tra le macchine virtuali in esecuzione. Un errore di sistema in una macchina virtuale o in una partizione non pregiudicherà l'esecuzione delle altre partizioni in esecuzione sulla stessa piattaforma hardware. Questo isolamento logico protegge le macchine virtuali a livello più basso rendendole inconsapevoli, e quindi non influenzandole, da errori o guasti al di fuori delle loro assegnazioni. Questo strato di astrazione, componente chiave della virtualizzazione, rende ogni partizione come se fosse in esecuzione su hardware dedicato. Tale isolamento però non impedisce la flessibilità che avrebbe un'architettura puramente fisica. Le partizioni possono all'occorrenza essere riassegnate per servire altre funzioni. Si immagini un server che ospiti un'applicazione client / server utilizzata solo durante l'orario che va dalle 8:00 alle 17:00 dal lunedì al venerdì, un altro server che abbia in esecuzione processi batch atti a chiudere le operazioni di business ogni notte, e un altro ancora responsabile dei lavori di manutenzione sui dati nel week-end. In un mondo puramente fisico, essi esisterebbero come tre server dedicati, molto utilizzati durante le ore di rispettivo esercizio, ma inattivi quando non svolgono la loro attività. Questo provocherebbe un grande sottoutilizzo e di conseguenza renderebbe più caro l'investimento. La virtualizzazione risolve il problema consentendo ad una singola partizione logica o fisica di essere riassegnata all'occorrenza a ciascuna funzione. Nei giorni feriali ospiterebbe l'applicazione client / server di giorno e eseguirebbe i processi batch durante la notte. Durante i fine settimana, sarebbe riassegnata alle attività di manutenzione dei dati, per ritornare poi ad ospitare l'applicazione client / server la mattina del lunedì. Questa flessibilità consente alle organizzazioni IT di utilizzare le partizioni "part-time", eseguendo i processi business principali nello stesso modo in cui lo farebbero i server fisici, ma realizzando anche una riduzione dei costi pur mantenendo elevati livelli di affidabilità.
Un altro settore che fa lievitare i costi è lo sviluppo di server di *standby* o di *failover* (malfunzionamento) necessari per mantenere il sistema disponibile

durante i periodi di interruzioni pianificate e non. Benché siano in grado di ospitare carichi di lavoro mirati, i server rimangono inattivi durante queste interruzioni e, in alcuni casi, non possono essere usati del tutto. Si riducono così ad essere sotto-utilizzati, fornendo poco valore all'azienda sebbene il loro costo sia molto elevato.

La virtualizzazione aiuta a risolvere questo problema consentendo all'occorrenza la dotazione di partizioni aggiuntive *just-in-time* (JIT, sul momento, assegnate in modo dinamico) o *on-demand* (su richiesta). Ad esempio, una partizione che è stata costruita con sistema operativo e applicazioni e configurata può essere messa in uno stato di inattività (spegnendola o mettendola in sospensione), pronta per essere attivata quando si verifica un errore. Quando necessario, la partizione si attiva senza alcun problema di fornitura, installazione o configurazione di hardware.

2.1.3 – Sicurezza

La stessa tecnologia che fornisce isolamento agli errori delle applicazioni è anche in grado di fornire isolamento alle falle di sicurezza. Se una particolare partizione venisse compromessa, essendo isolata dalle altre partizioni, circoscriverebbe il problema, evitando di compromettere anche le altre.

Soluzioni a questo eventuale problema possono essere attuate anche isolando ulteriormente le partizioni compromesse e le istanze del sistema operativo negando loro le risorse necessarie affinché esse esistano. I cicli della CPU possono essere ridotti, l'accesso alla rete e ai dischi di I / O possono essere interrotti, fino a raggiungere, nei casi più estremi, il blocco totale del sistema.

Sarebbe difficile, se non impossibile, eseguire questi compiti se l'istanza compromessa fosse in esecuzione direttamente su un host fisico.

Quando si stanno consolidando i carichi di lavoro attraverso la virtualizzazione, le configurazioni di sicurezza possono essere specificate per ogni singola partizione e non per tutto il server nel suo insieme. Un esempio di ciò potrebbe essere l'account *super-user* (comando *su* da console su sistema operativo Unix, permette l'accesso completo al computer). Le applicazioni, consolidate in un unico sistema

operativo, se eseguite direttamente su di un server fisico condividerebbero varie impostazioni di sicurezza; in particolare, l'accesso *root* sarebbe lo stesso per ciascuna.
Tuttavia, quando gli stessi carichi di lavoro vengono consolidati su partizioni virtuali, ciascuna partizione può essere configurata con credenziali diverse, mantenendo così l'isolamento dell'accesso al sistema con privilegi amministrativi.

Concludendo, la virtualizzazione è la scelta migliore in quasi tutte le società, piccole o grandi che siano. Essa risponde a pieni voti alle esigenze di contenimento dei costi e alla diminuzione di spazio all'interno dei datacenter, attuando i vari mandati aziendali in tempi più stretti e con ottimi risultati. Questa tecnologia è il fiore all'occhiello in grado di soddisfare le esigenze del business fornendo nel contempo un valore aggiunto alle operazioni IT e alla gestione e distribuzione delle infrastrutture.

2.2 – VIRTUALIZZAZIONE BASATA SUL SOFTWARE

Benché ci siano vari modi per virtualizzare le risorse informatiche utilizzando un vero VMM, ognuno di essi punta allo stesso obiettivo: consentire ai sistemi operativi di essere eseguiti in modo indipendente e isolato, come se venissero avviati direttamente sulla piattaforma hardware.
Sebbene le tecnologie di virtualizzazione *hardware-based*, che virtualizzano e astraggono l'hardware completamente esistano ancora, tendono però ad essere costose e poco flessibili.
Di conseguenza sono sorti moltissimi software hypervisor e VMM per eseguire la virtualizzazione attraverso meccanismi *software-based*. Essi garantiscono un livello di isolamento in cui il basso livello, nucleo centrale dell'architettura della CPU, è portato vicino ai livelli software dell'architettura per consentire ad ogni macchina virtuale di avere il proprio ambiente dedicato. Lo stretto rapporto tra l'architettura della CPU e i sistemi operativi virtualizzati è la chiave del suo successo.

2.3 – RELAZIONE TRA CPU E SISTEMA OPERATIVO

Le architetture hardware ideali sono quelle in cui il sistema operativo e la CPU sono progettati e costruiti l'uno per l'altra, e sono quindi associati strettamente. Un corretto uso delle chiamate di sistema richiede un attento coordinamento tra il sistema operativo e la CPU.

Questo rapporto simbiotico offre svariati vantaggi in materia di sicurezza e stabilità. Un esempio è il MULTICS Time-Sharing System, progettato per una particolare architettura della CPU, che a sua volta è stata progettata per esso.

Ciò che a suo tempo rese il MULTICS speciale è stato il suo approccio molto stretto e chiuso sulle operazioni software per eliminare il rischio o anche solo la possibilità che un componente difettoso non compromettesse o destabilizzasse gli altri componenti. Collocò meccanismi formali, chiamati anelli di protezione (protection rings), per separare il sistema operativo "trusted" dai programmi utente "untrusted".

2.3.1 – La Famiglia x86

L'architettura CPU più comune utilizzata nei moderni computer è l'IA-32, o x86-compatibile. Cominciando con il chipset 80286, la famiglia x86 fornì due metodi principali di indirizzamento della memoria: in modalità reale (attraverso le istruzione privilegiate) e in modalità protetta (con le istruzioni utente). Nel chipset 80386 e successivi venne introdotta una terza modalità chiamata modalità virtuale 8086, o VM86, che permetteva l'esecuzione di programmi scritti per la modalità reale, ma eludendo le regole della modalità reale senza dover innalzarli alla protetta.

La modalità reale, limitata ad un singolo megabyte di memoria, diventò rapidamente obsoleta; e la modalità virtuale venne bloccata in un'operazione a 16-bit, diventando anch'essa obsoleta quando i sistemi operativi a 32 bit divennero

ampiamente disponibili per l'architettura x86. La modalità protetta, componente fondamentale dell'x86, venne dotata di numerose nuove funzionalità per supportare il multitasking. Esse comprendevano ad esempio la segmentazione dei processi e il supporto hardware per la memoria virtuale e la commutazione dei processi.

Nella famiglia x86 la modalità protetta utilizza quattro livelli privilegiati, o Ring, numerati da 0 a 3. La memoria di sistema è divisa in segmenti e ogni segmento viene assegnato e dedicato ad un particolare Ring. Il processore utilizza il livello privilegiato per determinare cosa può e non può essere fatto con il codice o i dati contenuti nel segmento. Il termine "Ring" deriva dal sistema MULTICS, dove i livelli privilegiati erano visualizzati come una serie di anelli concentrici. Il Ring-0 è considerato l'anello più interno, il kernel al quale è assegnato il controllo totale del processore. Il Ring-1 e -2 attuano l'esecuzione dei servizi del sistema operativo e dei driver delle periferiche. Infine il Ring-3, l'anello più esterno, legato all'esecuzione delle applicazioni, fornisce solo accesso limitato, come illustrato nella Fig 2.2.

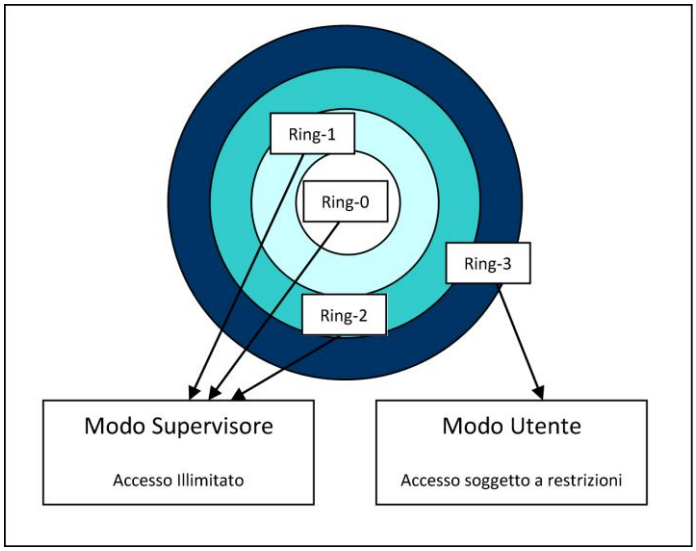


Fig. 2.2 – Anelli privilegiati nell'Architettura x86

Nella moderna architettura dei sistemi operativi Windows, Linux e la maggior parte dei sistemi UNIX viene ancora utilizzata la struttura ad anelli, anche se è stata ridotta ad un approccio a doppio strato che utilizza solo gli anelli 0 e 3. Al

Ring-0, comunemente chiamato Supervisor Mode (Modalità Supervisore), è stato demandato il compito assegnato in precedenza ai tre anelli interni: Ring-0 -1 -2, mentre il Ring-3, noto come User Mode (Modalità Utente), ha mantenuto le stesse funzioni. I meccanismi di sicurezza nell'hardware impongono restrizioni al Ring-3 limitando l'accesso di codice ai segmenti, al paging, e alle operazioni di input / output. Se un programma utente che viene eseguito sul Ring-3 cerca di indirizzare della memoria al di fuori dei suoi segmenti, un hardware interrupt (interruzione di hardware) interrompe l'esecuzione di codice.

2.4 – I TIPI DI VMM

Si è visto come il "Supervisor Mode" sia la modalità di esecuzione, su un processore x86, che consente l'esecuzione di tutte le istruzioni, comprese le istruzioni privilegiate come le operazioni di I / O e di gestione della memoria. E' in questa modalità (Ring-0) che il sistema operativo funziona normalmente. Siccome il Ring-3 è basato sul Ring-0, qualsiasi compromissione o instabilità del sistema collide direttamente con l'esecuzione dello User Mode nel Ring-3. Per isolare il Ring-0 per ogni guest virtualizzato, diviene allora necessario spostarlo vicino ad essi. In questo modo, un errore o guasto del Ring-0 per un determinato guest virtualizzato non incide sul Ring-0, e conseguentemente sul Ring-3, di qualsiasi altro guest. I Ring-0 percepiti dai guest possono risiedere in uno dei Ring-1, -2, -3 delle architetture x86. Naturalmente, quanto più la percezione del Ring-0 è lontana dal vero Ring-0, tanto più sarà difficile eseguire le operazioni hardware direttamente, con conseguente riduzione delle prestazioni.

La virtualizzazione muove il Ring-0 sopra il modello ad anelli privilegiati mettendo il Virtual Machine Monitor, o VMM, in uno degli anelli, che a sua volta rappresenta l'implementazione del Ring-0 sulle macchine virtuali guest. E' su questo particolare Ring-0 che i sistemi operativi guest vengono eseguiti, mentre la VMM gestisce l'interazione reale con la sottostante piattaforma hardware per l'accesso alla CPU, alla memoria e alle risorse di I / O. Esistono due tipi di VMM che indirizzano la rappresentazione del Ring-0 :

- **VMM Tipo 1**. software che viene eseguito direttamente su di una piattaforma hardware sul vero Ring-0. I sistemi operativi guest vengono eseguiti ad un livello al di sopra dell'hardware, consentendo un vero isolamento di ciascuna macchina virtuale.

- **VMM Tipo 2.** software che viene eseguito all'interno di un sistema operativo, di solito sul Ring-3. Poiché non ci sono anelli supplementari rispetto al Ring-3 nell'architettura x86, Il Ring-0 che le macchine virtuali eseguono su di esse è molto più lontano dall'effettiva piattaforma hardware. Anche se offre alcuni vantaggi, di solito questa VMM è aggravata dalle performance; ad esempio le chiamate verso l'hardware devono attraversare molti strati diversi prima di arrivare a destinazione.

2.5 – RUOLO E PROGETTAZIONE DELLA VMM

Per creare le partizioni virtuali in un server, viene eseguito un sottile strato di software chiamato Virtual Machine Monitor (VMM) direttamente sulla piattaforma hardware fisica. Uno o più sistemi operativi guest e un insieme di applicazioni possono quindi essere eseguiti al di sopra della VMM.

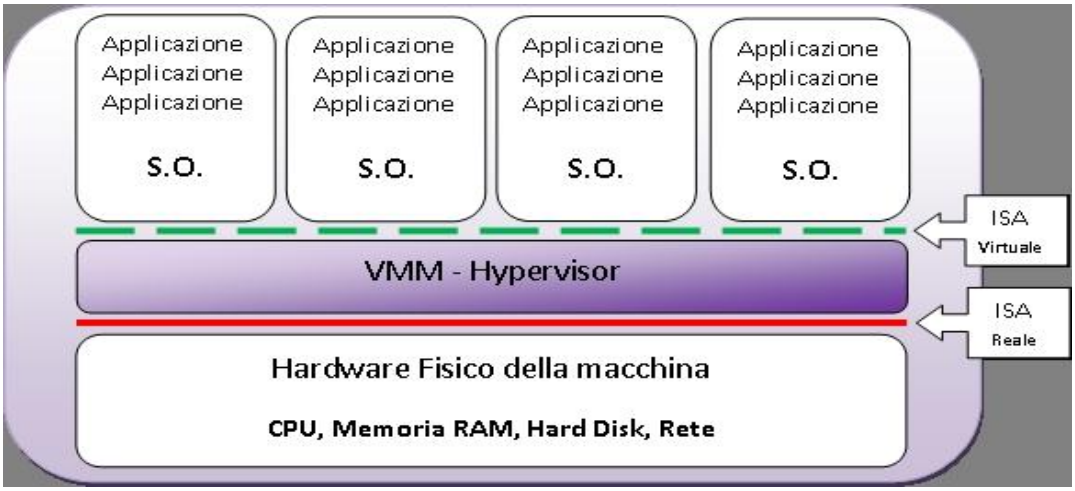


Fig. 2.3 – SO e pile di applicazioni gestite dallo strato software della VMM

Il VMM è il centro della virtualizzazione dei server. Gestisce da una parte le risorse hardware e dall'altra le richieste dei sistemi operativi guest e delle loro applicazioni. Assegna ad ogni guest un insieme virtuale di CPU, memoria, I / O, e risorse su disco basato sull'hardware fisico disponibile o sulla base di una selezione personalizzata dell'hardware sottostante.
Per quanto riguarda i criteri di progettazione della VMM, i requisiti Popek e Goldberg sono spesso indicati quale fonte di riferimento iniziale; essi definiscono le condizioni delle architetture dei computer atte a sostenere la virtualizzazione. Scritti nel 1974 per i computer di terza generazione di quel periodo, hanno generalizzato le condizioni che il software che fornisce l'astrazione di una macchina virtuale o VMM, deve soddisfare:

- **Equivalenza.** Un programma in esecuzione attraverso la VMM deve dimostrare un comportamento prevedibile che è sostanzialmente identico a quello mostrato durante la sua esecuzione direttamente sulla piattaforma hardware sottostante. Questa condizione viene chiamata anche "*Fidelity*".

- **Controllo delle risorse.** La VMM deve avere il controllo completo in ogni momento delle effettive risorse hardware virtualizzate per i sistemi operativi guest. Questa condizione viene chiamata anche "*Safety*".

- **Efficienza.** Un grande numero di istruzioni macchina devono essere eseguite senza l'intervento della VMM, ma dall'hardware stesso. Questa condizione viene chiamata anche "*Performance*".

Secondo Popek e Goldberg, il problema che gli sviluppatori della VMM devono affrontare è la creazione di una stessa che soddisfi le condizioni precedenti quando si opera all'interno delle caratteristiche del Istruction Set Architecture (ISA) della specifica piattaforma hardware. Il set ISA può essere classificato in tre gruppi di istruzioni: privilegiato, controllo sensibile, e funzionamento. Le istruzioni privilegiate vengono bloccate se il processore è in User Mode mentre vengono eseguite se è in Supervisor Mode. Le istruzioni di controllo sensibile tentano di modificare la configurazione delle reali risorse della piattaforma

hardware. Infine le istruzioni di funzionamento sono quelle il cui comportamento o il cui risultato dipende dalla configurazione delle risorse.
Le VMM devono lavorare con ogni gruppo di istruzioni, affinché si mantengano le condizioni di equivalenza, il controllo delle risorse e l'efficienza. Tutte le VMM di oggi soddisfano le prime due condizioni: equivalenza e controllo delle risorse. Gestiscono anche in modo efficace il sistema operativo guest e la piattaforma hardware sottostante tramite l'emulazione, l'isolamento, l'allocazione e l'incapsulamento:

- **Emulazione.** L'emulazione è importante per tutti i sistemi operativi guest. La VMM deve rappresentare un ambiente hardware completo, o macchina virtuale, sia per i sistemi operativi che per le applicazioni. Il sistema operativo e le applicazioni sono all'oscuro del fatto che condividono le risorse hardware con altre applicazioni. L'emulazione è la chiave per soddisfare la proprietà di equivalenza.

- **Isolamento.** L'isolamento, anche se non richiesto, è importante per un sicuro e affidabile ambiente. Attraverso l'astrazione hardware, ogni macchina virtuale deve essere sufficientemente separata e indipendente dalle operazioni e dalle attività delle altre macchine virtuali. I guasti che si verificano in una singola macchina virtuale non devono coinvolgere le altre; ciò fornisce sicurezza e disponibilità ad alti livelli.

- **Allocazione.** La VMM deve assegnare con metodo le risorse della piattaforma alle macchine virtuali che gestisce. Risorse di elaborazione, memoria, rete e unità di storage devono essere bilanciati per ottimizzare le prestazioni e allineare i livelli di servizio con i requisiti di business. Attraverso l'allocazione la VMM soddisfa la proprietà del controllo delle risorse e, in parte, anche la proprietà di efficienza.

- **Incapsulamento.** L'incapsulamento, anche se non è specificato nei requisiti Popek e Goldberg, permette ad ogni pila di software di essere altamente portabile, in grado di essere copiata o spostata da una piattaforma che esegue la VMM ad un'altra. In alcuni casi è possibile

“migrare a caldo” le macchine virtuali, cioè spostarle da un supporto ad un altro senza doverle spegnere, cioè mantenendole in esecuzione. L'incapsulamento deve contenere le informazioni di stato per mantenere l'integrità della macchina virtuale trasferita.

Per quanto riguarda l’architettura più diffusa, la IA-32 (x86), tutto il software viene eseguito solamente in uno dei quattro Ring privilegiati. Il sistema operativo di norma viene eseguito nel Ring-0, che offre un accesso privilegiato alla più ampia gamma di risorse, del processore e della piattaforma. Le singole applicazioni solitamente vengono eseguite nel Ring-3, che limita alcune funzioni (come la mappatura della memoria) che potrebbero disturbare altre applicazioni. In questo modo, il sistema operativo mantiene il controllo per garantire un funzionamento ottimale.
Poiché la VMM deve avere il controllo privilegiato delle risorse della piattaforma, la soluzione consueta è eseguire la VMM nel Ring-0 e i sistemi operativi guest nel Ring-1 o nel Ring-3. Tuttavia, i sistemi operativi moderni sono stati specificamente progettati per l'esecuzione sul Ring-0, creando una sorta di problema. In particolare, esistono 17 istruzioni privilegiate in grado di controllare le risorse critiche della piattaforma. Tali istruzioni vengono utilizzate in modo occasionale, ad esempio quando un sistema operativo non viene eseguito nel Ring-0: in questo caso una di queste istruzioni può creare un conflitto, provocando un errore di sistema o una risposta non corretta. La sfida incontrata dalla VMM dell'architettura IA-32 (x86) è il mantenimento dei requisiti Popek e Goldberg mentre si lavora con l’ISA IA-32.

2.6 – TIPI DI VIRTUALIZZAZIONE

Si è già parlato del fatto che esistono molte forme di virtualizzazione nella moderna tecnologia dell’informazione, anche se la più comune e diffusa è la virtualizzazione dei server, tipico caso associato a questo termine.

Le piattaforme e architetture CPU più diffuse su cui viene implementata la virtualizzazione dei server sono le IA-32 o x86. Le sfide poste dall'architettura ISA x86 e dai requisiti Popek e Goldberg hanno portato a diversi approcci per lo sviluppo della VMM. Benché ci siano diverse implementazioni della VMM per x86, si possono riassumere in quattro distinte categorie:

2.6.1 – FULL VIRTUALIZATION

È una tecnica di virtualizzazione che fornisce un ambiente in grado di simulare completamente l'hardware sottostante. Il risultato è un sistema in cui tutto il software in grado di essere eseguito su hardware "reale" può essere eseguito anche su macchina virtuale. La virtualizzazione "full" è il tipo più supportato dai sistemi operativi guest.

Questo approccio è nato dall'impossibilità di virtualizzare alcune istruzioni a causa dei limiti imposti dall'architettura hardware. Per questo motivo solo le istruzioni virtualizzabili sono eseguite senza manipolazione da parte del programma di gestione delle macchine virtuali, mentre le altre devono essere gestite in modo diverso.

L'espressione *full virtualization* sembra indicare una virtualizzazione completa per tutti i componenti dell'architettura, ma in realtà non è del tutto vero. O per lo meno non lo è nelle macchine x86 poiché alcune delle chiamate privilegiate non possono essere intercettate; si è riusciti a risolvere il problema solo dagli anni 2005/06, attraverso le tecniche di *trap-and-emulate* ("intercetta ed emula") delle istruzioni x86 privilegiate.

I software di full virtualization operano dunque in due modi: alternando la tecnica di traduzione binaria all'esecuzione diretta.

La traduzione binaria (o riscrittura binaria) prevede che il flusso delle istruzioni sia analizzato dal programma di virtualizzazione alla ricerca di particolari istruzioni critiche (sensibili non privilegiate), ossia istruzioni che, come si è visto in precedenza, richiedono di essere eseguite con particolari privilegi (*kernel mode*) ma non generano, per limiti intrinseci del processore, un'eccezione (*trap*) se la loro esecuzione viene richiesta in un contesto non privilegiato.

Queste istruzioni critiche, vengono modificate (ad. es. con VMware ESX) in modo da poter essere eseguite dal VMM con i corretti privilegi: la macchina virtuale non può operare in kernel mode ma deve lasciare che il VMM agisca per essa.

Un programma di virtualizzazione che sfrutta la traduzione binaria funziona similmente ad un debugger: utilizza le stesse funzioni implementate per il controllo del codice sorgente. Inserisce dei breakpoint dove ci sono istruzioni critiche per la virtualizzazione e le emula.

Una macchina virtuale viene eseguita su un interprete invece che direttamente sulla CPU; in questo modo l'interprete può risolvere i problemi dovuti ad operazioni che ostacolano la virtualizzazione.

A differenza di altri approcci alla virtualizzazione (come la paravirtualizzazione, che vedremo in seguito) il codice di un'applicazione o del kernel guest non deve essere modificato per poter interagire correttamente con le risorse del sistema.

Sfortunatamente la traduzione binaria peggiora le performance soprattutto durante attività intensa di I/O: le prestazioni sono generalmente tra 80-97% di quelle del computer ospitante.

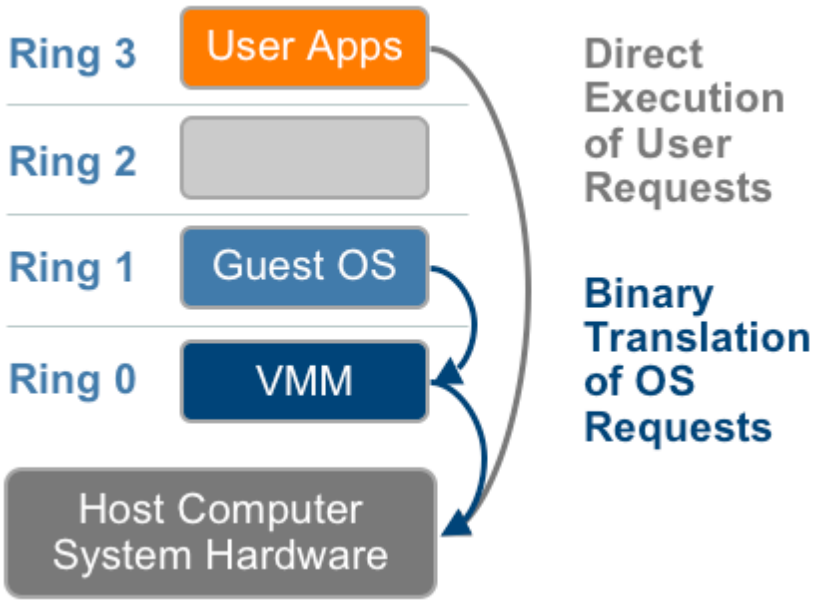


Fig. 2.4 – FULL VIRTUALIZATION

Esempi di prodotti che utilizzano questo tipo di virtualizzazione sono Adeos, Mac-on-Linux, Parallels Desktop per Mac, Parallels Workstation, VMware Workstation, VMware Server, VirtualBox, Win4BSD e Win4Lin Pro.

2.6.2 – PARAVIRTUALIZZAZIONE

"Para-" è un prefisso di origine greca che vuol dire "accanto", "con" o "a fianco." La paravirtualizzazione si riferisce infatti alla comunicazione tra il sistema operativo guest e l'hypervisor per migliorare le prestazioni e l'efficienza del sistema.

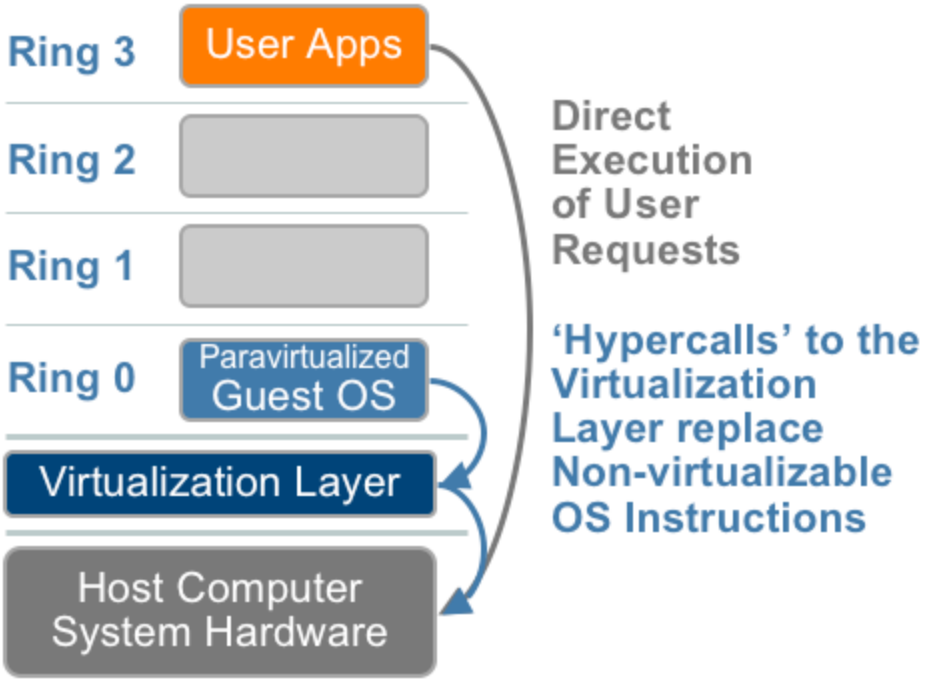


Fig. 2.5 – PARAVIRTUALIZZAZIONE

È una tecnica di virtualizzazione che prevede la simulazione parziale dell'hardware sottostante. Come mostrato in Fig. 2.5, comporta la modifica del kernel del sistema operativo per sostituire le istruzioni non virtualizzabili con delle hypercalls che comunicano direttamente con il livello di virtualizzazione hypervisor. L'hypervisor fornisce anche interfacce hypercall per altre operazioni critiche del kernel quali la gestione della memoria e degli interrupt.

La paravirtualizzazione è diversa dalla full virtualization, in cui il sistema operativo non modificato non sa di essere virtualizzato e le chiamate del OS sensibili sono intrappolate ("trapped") con la traduzione binaria. Il valore della paravirtualizzazione sta nel basso overhead di virtualizzazione, ma il vantaggio nelle prestazioni della paravirtualizzazione rispetto alla virtualizzazione full possono variare notevolmente a seconda del carico di lavoro. Come essa non può sostenere sistemi operativi non modificati (ad esempio Windows 2000/XP), la sua compatibilità e portabilità è molto limitata.

La paravirtualizzazione può anche introdurre importanti problemi di supporto e di manutenibilità negli ambienti di produzione in quanto richiede profonde modifiche al kernel del sistema operativo. Il progetto open source Xen è un esempio di paravirtualizzazione che virtualizza il processore e la memoria utilizzando una versione modificata del kernel Linux e virtualizza l'I/O usando driver di periferica del sistema operativo guest personalizzati.

Mentre costruire il sofisticato supporto della traduzione binaria necessario per la virtualizzazione completa è molto difficile, modificare il sistema operativo guest per consentire la paravirtualizzazione è relativamente facile.

2.6.3 – VIRTUALIZZAZIONE DEL SISTEMA OPERATIVO

Questo tipo di virtualizzazione si basa su una singola istanza di un sistema operativo. Essa tende ad essere leggera ed efficace, dovendo effettuare un'unica installazione del sistema operativo per la gestione e gli aggiornamenti. In aggiunta viene eseguita a velocità normale (senza rallentamenti) e supporta tutto l'hardware nativo e le caratteristiche del sistema operativo per cui la macchina host è configurata.

Di contro non supporta l'hosting di sistemi operativi misti, come Windows e Linux contemporaneamente e le macchine virtuali non sono così isolate e sicure come lo sono negli altri tipi di virtualizzazione. il Ring-0 poi è un sistema operativo completo, per cui aggiunge *overhead* e complessità rispetto allo spoglio microkernel della VMM. Infine è difficile identificare la sorgente da cui

provengono grandi richieste di risorse, per cui è molto complicato limitarne l'uso da parte dei guest.

2.6.4 – VIRTUALIZZAZIONE NATIVA

Detta anche virtualizzazione hardware assistita o ibrida, è la più recente tecnologia di virtualizzazione del gruppo x86.

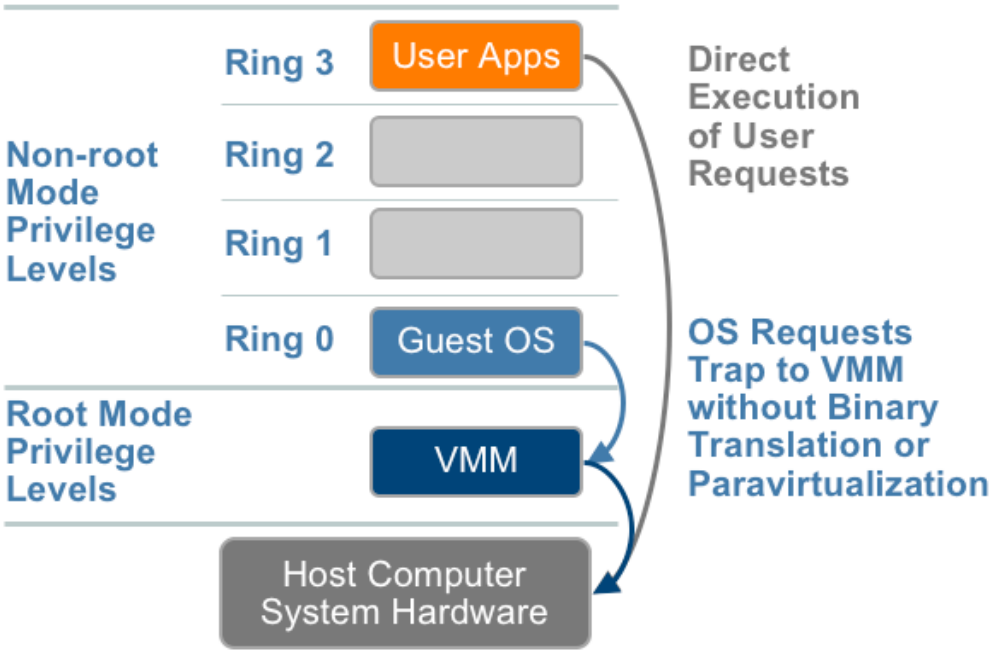


Fig. 2.6 – HARDWARE ASSISTED VIRTUALIZATION

I miglioramenti di prima generazione includono la Intel Virtualization Technology (VT-x) e la AMD-V di AMD le quali segnano le istruzioni privilegiate con una nuova modalità di esecuzione della CPU che consente di eseguire la VMM in una nuova modalità root sotto il ring 0. Come illustrato in Fig. 2.6, le chiamate privilegiate e sensibili sono impostate in modo che vengano automaticamente intrappolate dall'hypervisor, eliminando la necessità della traduzione binaria o della paravirtualizzazione. Lo stato guest viene memorizzato nelle Virtual Machine Control Structures (VT-x) o nelle Virtual Machine Control Blocks (AMD-V).

I processori con Intel VT e AMD-V sono disponibili dal 2006, per cui solo i sistemi più recenti contengono queste caratteristiche hardware assistite.
Come aspetti positivi impiega in modo selettivo le tecniche di accelerazione per la memoria e le operazioni di I/O. supporta i sistemi operativi x64 ed ha le più alte prestazioni di processore, memoria e I/O fra tutti i tipi di macchine virtuali per architetture x86.
Di contro richiede un processore che abbia l'accelerazione hardware assistita, tecnica che consente al sistema operativo ospite di avere accesso diretto alle risorse della piattaforma senza dover condividere il controllo dell'hardware o senza che venga emulato come faceva la VMM in precedenza e, per i guest paravirtualizzati, richiede alcune modifiche del sistema operativo (ma in misura minore rispetto alla paravirtualizzazione "pura").

CAPITOLO 4
CONCLUSIONI

In questo documento si è potuto vedere fino a che punto la virtualizzazione si sia sviluppata in questi ultimi anni, e che strada preferenziale ha intrapreso.
Partendo come tecnologia per sfruttare al meglio le risorse dei grandi e costosi mainframe degli anni '70, è poi mutata in una fase di decentralizzazione mirata alla clusterizzazione delle risorse hardware mediante il collegamento in rete di un certo numero di computer collegati tra loro con lo scopo di formare un'unica risorsa computazionale estremamente potente, affidabile e veloce ad un buon rapporto prezzo-prestazioni. Ciò è stato possibile grazie all'enorme boom economico delle macchine e dei sistemi operativi stand-alone durante l'ultimo decennio del secolo scorso. Successivamente però, all'aumentare della dimensione dei datacenter in infrastrutture orizzontali, al continuo aumento della potenza degli apparati di rete in rapporto al prezzo e soprattutto alla creazione di

software di virtualizzazione in grado di gestire tali macchine in modo sicuro ed efficace, negli ultimi anni si è tornati ad un paradigma di centralizzazione, preferendo consolidare i numerosi server stand-alone in apparati “blade” in grado di ospitare centinaia di server virtualizzati in un unico rack.

Questa situazione è la strada che si è deciso di intraprendere all’interno dell’azienda in cui ho svolto il tirocinio, la DBA LAB S.p.A.
Dopo attente valutazioni, in relazione soprattutto al rapporto costi/benefici, si è deciso di montare sull’apparato blade IBM già presente nel datacenter aziendale (vedi Appendice A, Rack 6) un sistema VMware di tipo paravirtualizzato che ha, come caratteristiche principali, il VMware Tools e l’ottimizzazione del driver di periferica virtuale. Il servizio VMware Tools fornisce una backdoor per l'hypervisor VMM utilizzata per i servizi come la sincronizzazione di tempo e la registrazione e lo spegnimento dei guest. Vmxnet è un driver di periferica I/O paravirtualizzato che condivide le strutture dati con l'hypervisor. Può usufruire delle funzionalità del dispositivo host per offrire un migliore throughput e un ridotto utilizzo della CPU. Il servizio VMware Tools e il driver di periferica vmxnet non sono però soluzioni di paravirtualizzazione della CPU; sono minimi cambiamenti non-intrusivi installati nel sistema operativo guest che non richiedono la modifica del kernel del sistema operativo stesso.

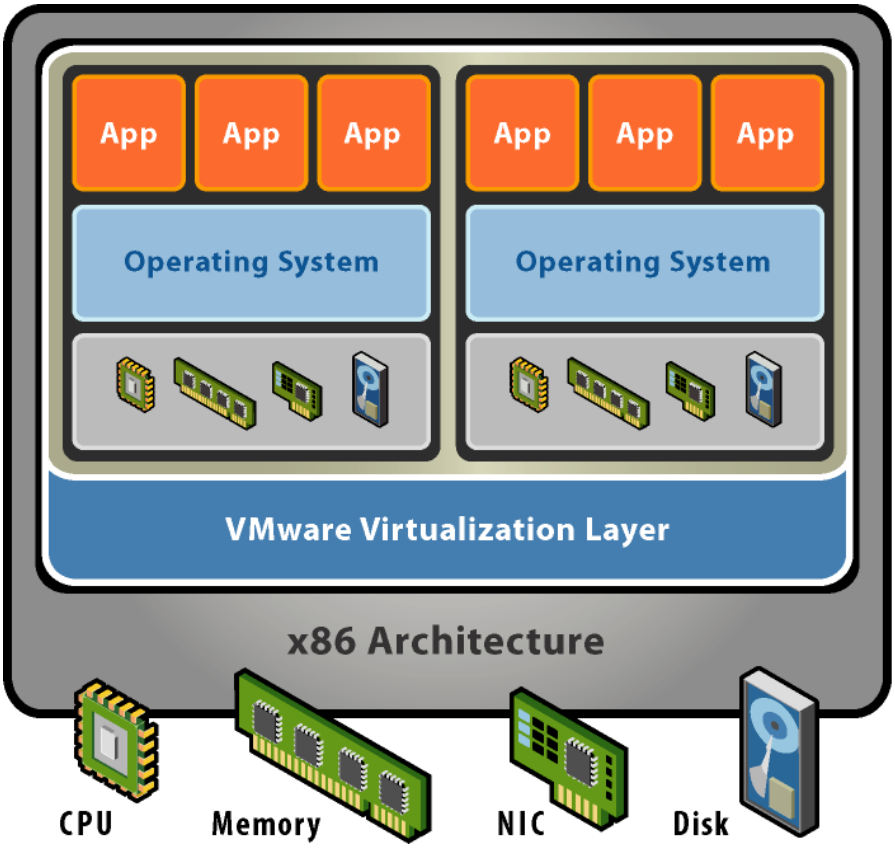


Fig. 4.1 – Virtualizzazione attraverso VMware

Grazie a questa infrastruttura, che sarà fisicamente implementata nel datacenter nei prossimi mesi, sarà possibile:

- **Consolidare i server e ottimizzare le infrastrutture**, aumentando il livello di utilizzo degli stessi e facendo così diminuire i costi energetici e di raffreddamento, nonché gestire e automatizzare i processi IT in modo da ottenere il massimo livello di disponibilità, prestazioni e scalabilità.
- **Applicare il Business Continuity**, una speciale tecnologia in grado di bilanciare automaticamente il carico di lavoro e migrare in tempo reale le macchine virtuali da una lama all'altra in caso di guasti dell'hardware sottostante, consentendo all'azienda di aumentare il tempo di operatività.
- **Automatizzare il ciclo di vita del software**, in modo che si semplifichi il processo di sviluppo e di collaudo del software degli sviluppatori, riducendo il tempo di provisioning dei server e incrementando la qualità dei loro prodotti.

- **Gestire in modo più efficace i desktop aziendali**, grazie ad apposite caratteristiche che migliorano il controllo di tali sistemi mediante amministrazione centralizzata.

In conclusione la soluzione di virtualizzazione ha risposto perfettamente alle esigenze dell'azienda.

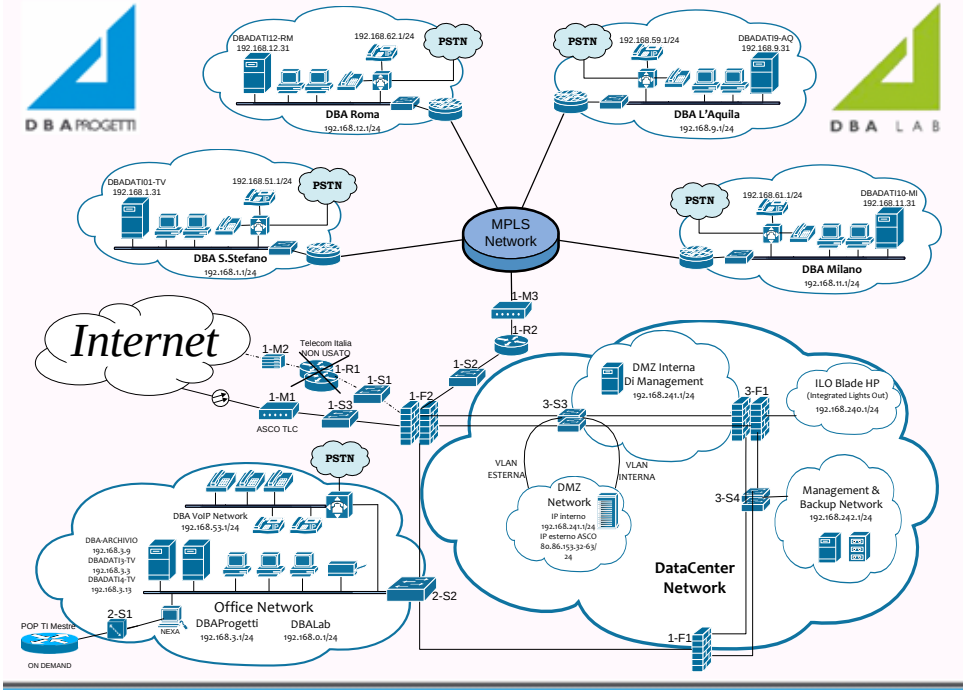
Appendice A

Struttura DataCenter

Diagramma di rete

Layout di sala

Composizione dei rack



Febbraio 2010

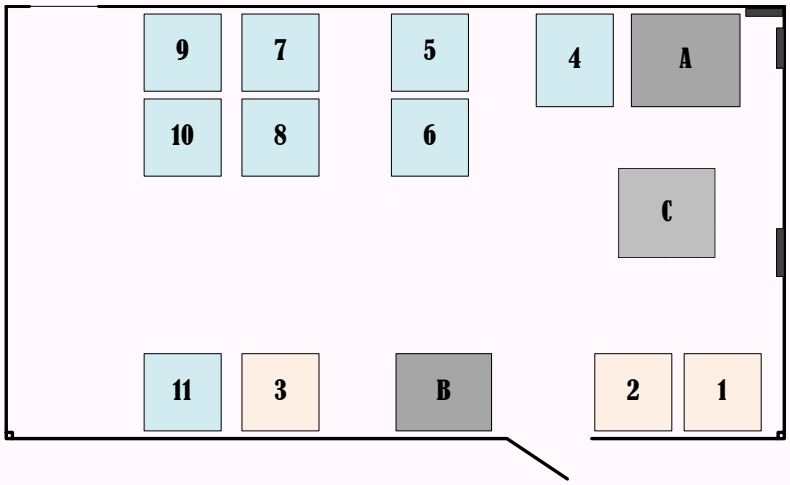
Legenda: Layout di Sala

- RACK
  - 1 : Collegamenti Geografici
  - 2, 3 : Apparati di rete "Office"
  - 4 : Server e apparati di Backup
  - 5, 6 : Server "blade" HP (5) e IBM (6)
  - 7 : Server Housing + DMZ
  - 8 : Server Veniceplaza
  - 9, 10 : Server DBA

- APPARATI SUPPLEMENTARI
  - A : Gruppo di continuità UPS GALAXY 5000 (40KW)
  - B, C : Unità Condizionatore HPM Digital D23UA (B)
    + Ventilatore a soffitto HPSE10 (C)

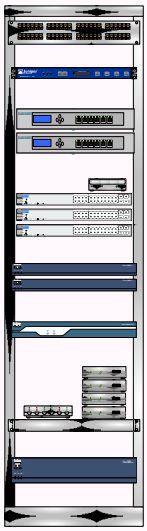
Febbraio 2010

Layout di sala



Febbraio 2010

Rack 1 – Collegamenti geografici



Patch Panel

F1 — 1 x Firewall – Juniper 25

F2 — 2 x Firewall – Clavister 3100

M1 — 1 x Media Converter ASCO TLC

S1, S2, S3 — 3 x Switch – HP Procurve 2626

R1 — 2 x Router – Cisco 2600

R2 — Router – Cisco 1840 MPLS

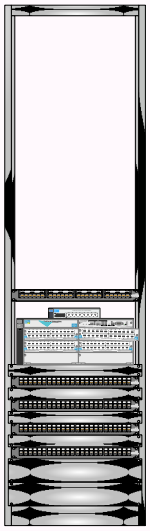
M2 — Borchie Telecom Italia

M3 — Modem MPLS

Cisco – RPS (Redundant Power Supply)

Febbraio 2010

Rack 2 – Apparati di rete "Office" (A)



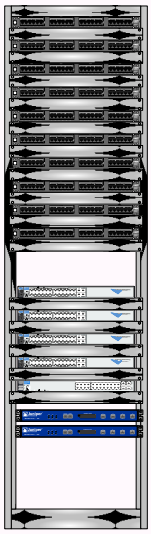
S1 — Patch Panel AMP NETCONNECT Cat.6 (24porte)
Switch – HP Procurve 408 (NeXa)

S2 — Switch – HP Procurve 5406 (Core Switch)

4x Patch Panel Panduit (24porte)

Febbraio 2010

Rack 3 – Apparati di rete "Office" (B)



10 x Patch Panel – AMP NETCONNECT Category 6 (A -- J)

S3 — 2 x Switch – HP Procurve 2510G-24 (192.168.241.x)

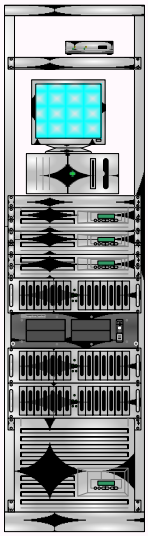
S4 — 2 x Switch – HP Procurve 2510G-24 (192.168.242.x)

S5 — 1 x Switch – HP Procurve 2626 (HOSTING2)

F1 — 2 x Firewall – Juniper 50

Febbraio 2010

Rack 4 – Apparati di backup



Dispositivo KVM – Master View Plus

Vecchio gestionale "MAGO" (HP BRIO)

DBAPALMINA: gestionale Palmina (HP-DL-320-G5)

BCKSRV - ex Backup-BACULA (HP-DL-320-G2)

DBAARCHIVIO: archivio DBAProgetti (HP 360 – G4P)

NAS (Buffalo Terastation Pro)

ULTRIUM TAPE 215/232

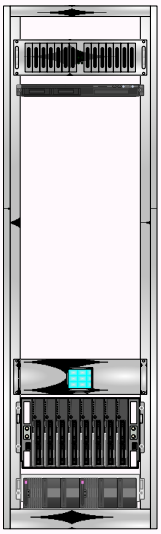
NAS (HP Storage work MSA-20)

NAS (Promise)

UPS (APC)

Febbraio 2010

Rack 5 – Apparati "blade" HP



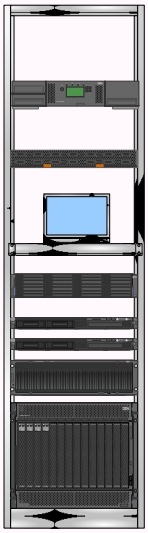
NAS (HP MSA 1000)

HP ProLiant DL320G5p

Server blade (HP Blade Server)

Febbraio 2010

Rack 6 – Apparati "blade" IBM



IBM TS3100 LTO-4 Tape Drive

MultiSwitch F.O. e Gb Ethernet

Console + monitor (IBM)

System Storage (SAN DS3400)

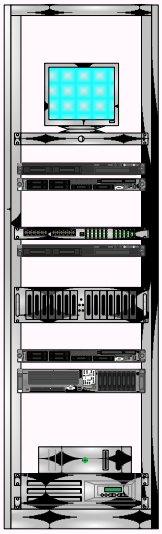
Startup servizi virtuali su Blade server (HP DL 360)
Startup servizi virtuali su Blade server (HP DL 320)

IBM System Storage (SAN DS4700)

Server blade (IBM Blade Server)

Febbraio 2010

Rack 7 – Server Housing + DMZ



Monitor per gestione locale

DBAMZ: Web Clienti DBALab+ Clienti Oil e TLC (HP-DL-320-G2)
DBASQL3: DB server gestionale (IBM SystemX 3650)

Dispositivo KVM – switch#3
NEROAV: hosting NeroAvorio SpA (HP-DL-320-G3)

NAS per backup datacenter (Buffalo TeraStation Pro)

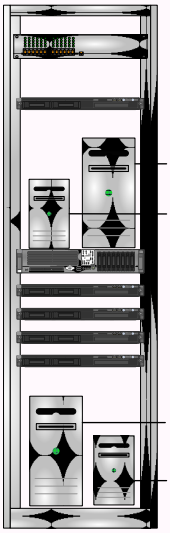
4M: Web Veniceplaza ASP (HP-DL-360-G3)
ERICKSON: Xen + 2 virt (Cluster + Blade) (HP-DL-380-G5)

DBASTAMPE3: Server stampanti (HP Brio)

SERVER01 IP 10.10.12.1 (Vecchio Gestionale)

Febbraio 2010

Rack 8 – Server VenicePlaza



Dispositivo KVM – switch#1

DBACONTROLLO: WebNeXa TI (Fujitsu Primergy L200)

Server controllo Clavister (HP WorkStation XW4000)

Server Xen Control (HP Compaq)

DBAVM: "Posic" (HP-DL-380-G5)

VPSRV02-TV: AD,DNS,Exchange DBALab (HP-DL-360-G4)
VPSRV03-TV: sviluppo PHP (HP-DL-360-G4P)
VPSRV04-TV: Database CRM e test (HP-DL-360-G4P)
VPSRV05-TV: sviluppo .NET (HP-DL-360-GL)

RES03: Hosting dedicato DAMA SpA (HP-ML-310)

Repository CentOS (HP-ML-110)

Febbraio 2010

Rack 9 – Server DBA(A)

Dispositivo KVM – switch#2

Monitor per gestione locale

DBAMAILBOX: Exchange DBAProgetti + AD (HP-DL-380-G5)

DBANET3: AD+DNS DBAProgetti (rack industriale)

DBADATI3: file server DBAProgetti (HP-DL-380-G4)

DBAGEST3: Citrix per collegamento a gestionale (HP-DL-320-G2)
DBAWEB3: intranet DBA Progetti (HP-DL-140)
Server spento (HP-DL-140-G3)
DBAWSUS: IP 192.168.3.60 (HP-DL-360-G4P)

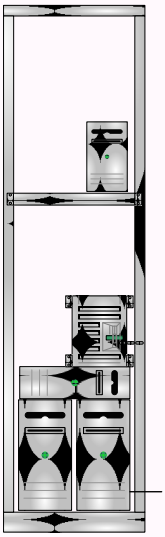
VPSRV06-TV: Exchange nuovo DBALAB (HP-DL-360-G4P)

BACULASRV: nuovo server di backup "Bacula" (HP-DL-380-G6)

ZABBIX (HP-DL-380-G5)
DBADOC3-TV (HP-DL-360-G4P)

Febbraio 2010

Rack 10 – Server DBA(B)



DBADATI4-TV 192.168.3.12 (HP-ML-110)

ARCHIVIOSYS 192.168.3.77 (Buffalo TeraStation Pro)

DBATECNOKEY-TV (HP-COMPAQ)

2 x WorkStations XW4000

SERVERCONCRETE-TV

Febbraio 2010

Bibliografia e link

Libri

Hoopes, J. et al. (2009), *Virtualization for Security*, Burlington, Syngress

Ruest, D. & Ruest, N. (2009), *Virtualization: A Beginner's Guide,* McGraw Hill

Smith, J. & Nair, R. (2005), *Virtual Machines: versatile platforms for systems and processes*, Morgan Kaufmann Publishers

Teti, A. (2009), *Il futuro dell'Information & Comunication Technology,* Springer Milan

Williams, D. & Garcia, J. (2007), *Virtualization with Xen*^{TM}, Burlington, Syngress

Articoli consultati

Popek, G. & Goldberg, R. (1974), *Formal Requirements for Virtualizable Third Generation Architectures*, Association for Computing Machinery, Inc.

Siti internet consultati

Morati, P. (2005), *La corsa alla virtualizzazione dei server*. Disponibile all'indirizzo: http://www.nwi.it

VMWare, (2007), *Understanding Full Virtualization, Paravirtualization, and Hardware Assist.* Disponibile all'indirizzo: http://www.vmware.com

Wikipedia, (2010). Disponibile all'indirizzo: http://it.wikipedia.org

Siti dell'azienda, (2010). Disponibili agli indirizzi: http://www.dbalab.it, http://www.dbagroup.it, http://www.dbaprogetti.it