

UNIVERSITÀ DEGLI STUDI DI UDINE

Facoltà di Scienze Matematiche, Fisiche e Naturali

Corso di Laurea Triennale in Informatica

Tesi di Laurea

**SISTEMI DI VIRTUALIZZAZIONE:
STATO DELL'ARTE,
VALUTAZIONI, PROSPETTIVE**

Relatore:
Prof. MARINO MICULAN

Laureando:
MARCO LONDERO

ANNO ACCADEMICO 2006-2007

A tutti coloro
i quali vogliono una dedica.

Introduzione

Per virtualizzazione si intende la creazione di una versione virtuale di una risorsa normalmente fornita fisicamente. Qualsiasi risorsa hardware o software può essere virtualizzata: sistemi operativi, server, memoria, spazio disco, applicativi, reti. Un tipico esempio di virtualizzazione è la divisione di un disco fisso in partizioni logiche.

I meccanismi più avanzati di virtualizzazione permettono la ridefinizione dinamica sia delle caratteristiche della risorsa virtuale, sia della sua mappatura su risorse reali. [1]

Ad oggi la virtualizzazione può essere effettuata a livello software ed a livello hardware. Tra gli svariati impieghi possibili, il più utilizzato è probabilmente la virtualizzazione di sistemi operativi. Via software è necessario un sistema operativo ospitante in esecuzione (*host*) che esegua un software di virtualizzazione che crei ad alto livello le varie “virtual machine” (*guest*), ognuna delle quali gira come se fosse un normale programma e che comunica con l’hardware solo indirettamente, tramite il software di controllo che agisce a basso livello. Via hardware, attualmente, Intel ed AMD hanno sviluppato indipendentemente le estensioni di virtualizzazione x86. Esse non sono tra loro completamente compatibili, ma grosso modo supportano le stesse istruzioni. Entrambe permettono ad una macchina virtuale di eseguire un sistema operativo ospite senza incorrere in grosse perdite prestazionali causate dall’emulazione via software. [2]

Questa tesi si occuperà di fornire una panoramica sullo stato dell’arte e sugli strumenti di virtualizzazione x86, legati in particolar modo alla virtualizzazione del sistema operativo.

Indice

Introduzione	v
Elenco delle figure	ix
1 Panoramica generale	1
1.1 Macchina virtuale	1
1.2 Hypervisor	2
1.3 Exokernel	3
1.4 Perché ricorrere alla virtualizzazione?	4
2 Macchine virtuali	7
2.1 VMware	8
2.1.1 VMware Player	9
2.1.2 VMware Server	10
2.1.3 VMware Workstation	13
2.1.4 VMware Infrastructure	15
2.2 Virtualizzazione nel mondo Microsoft	18
2.2.1 Microsoft Virtual Server	18
2.2.2 Microsoft Virtual PC	21
2.3 Ultimi arrivi	22
2.3.1 Oracle VM	23
2.3.2 Sun xVM	23
2.3.3 Microsoft Hyper-V	25
2.4 Tabelle riassuntive	26
3 Paravirtualizzazione	29
3.1 XEN	30
3.2 User-mode Linux	33
3.3 Tabelle riassuntive	36

4	Storage per macchine virtuali	39
4.1	Storage unico condiviso	39
4.1.1	Storage Area Network	40
4.1.2	Network Attached Storage	41
4.2	Storage distribuito	42
4.3	Alternative disponibili	43
5	VMware Vs. XEN: confronto sul campo	51
5.1	Installazione software virtualizzazione	52
5.2	Creazione VM/installazione guest OS	54
5.3	Costi e metodi di implementazione	56
5.4	Problemi riscontrati	58
5.4.1	Timing VM	58
5.4.2	Dischi virtuali	59
5.5	Conclusioni	60
6	Conclusioni e prospettive	61
	Bibliografia	63
	Ringraziamenti	69

Elenco delle figure

1.1	Hypervisor	2
1.2	Exokernel	3
2.1	Struttura di VMware	8
2.2	Struttura di VMware Server	11
2.3	Struttura di VMware Workstation	13
2.4	Struttura di VMware Infrastructure	15
2.5	Struttura di VMware VMFS	16
2.6	Struttura di VMware VMotion	17
2.7	Struttura di VMware Storage VMotion	17
2.8	Struttura di Microsoft Virtual Server	19
2.9	Struttura di Oracle VM	24
2.10	Struttura di Microsoft Hyper-V	26
3.1	Struttura di XEN	30
3.2	Struttura di User-mode Linux	34
3.3	Differenze strutturali tra UML e VMware	35
4.1	Struttura di una SAN	40
4.2	Struttura di sistema con NAS	42
5.1	Struttura exokernel	57

Capitolo 1

Panoramica generale

Prima di poter iniziare è fondamentale definire alcuni concetti che stanno alla base del mondo della virtualizzazione ed i motivi che spingono una realtà ad adottare una soluzione informatica virtualizzata.

In questo capitolo verranno quindi introdotti i seguenti termini: “*macchina virtuale*”, “*hypervisor*”, “*exokernel*”.

1.1 Macchina virtuale

Inizialmente, “*virtual machine*” veniva usato per indicare la creazione di una molteplicità di ambienti di esecuzione identici in un unico computer, ciascuno con il proprio sistema operativo, infatti questo genere di virtualizzazione è particolarmente utilizzata nel campo dei mainframe e dei supercomputer. Lo scopo di tale tecnica era quello di dividere tra più utenti l’uso di un singolo computer dando ad ognuno l’impressione di esserne l’unico utilizzatore.

Il significato più comune oggi è quello di un programma che emula un calcolatore (di solito un calcolatore astratto, al quale, cioè, non corrisponde un calcolatore reale). I programmi applicativi vengono scritti in un linguaggio che viene compilato per questo calcolatore immaginario (cioè tradotti nelle sue istruzioni native) e, una volta compilati, vengono eseguiti sulla macchina virtuale software. Dal momento che si possono scrivere diverse macchine virtuali per diverse piattaforme, il programma compilato può “girare” su qualsiasi piattaforma su cui “gira” la macchina virtuale. [3]

Un linguaggio moderno che fa uso della macchina virtuale è il *Java*: i programmi scritti in Java vengono infatti compilati nel linguaggio *bytecode*, che gira sulla “*Java Virtual Machine*”.

Progenitori delle macchine virtuali odierne si possono considerare sia la “*macchina p*”, cioè il calcolatore astratto per cui venivano (e vengono tuttora)

compilati i programmi in *Pascal* nelle prime fasi della compilazione (producendo il cosiddetto *p-code*), sia la “*macchina S*”, un altro calcolatore astratto (che però ebbe anche una realizzazione concreta), per cui venivano compilati i programmi in *Simula* nelle prime fasi della compilazione (producendo il cosiddetto *S-code*).

La virtualizzazione può essere vista in diversi modi:

- emulazione: la macchina virtuale simula completamente l’hardware, utilizzando un sistema operativo reale che poi gira sulla CPU virtuale; [4]
- paravirtualizzazione: la macchina virtuale non simula un hardware ma offre speciali API che richiedono modifiche nel sistema operativo;
- virtualizzazione nativa (o totale): la macchina virtuale simula solo parte dell’hardware e quindi il sistema operativo richiede modifiche ad hoc.

1.2 Hypervisor

L’*hypervisor*, detto anche “*virtual machine monitor*” (o *VMM*) è il componente chiave per un sistema basato sulla virtualizzazione e deve operare in maniera trasparente senza pesare con la propria attività sul funzionamento e sulle prestazioni dei sistemi operativi. Svolge attività di controllo al di sopra di ogni sistema, permettendone lo sfruttamento anche come monitor e debugger delle attività dei sistemi operativi e delle applicazioni in modo da scoprire eventuali malfunzionamenti e dare la possibilità di intervenire celermente. I requisiti richiesti a questo scopo sono quelli di compatibilità, performance e semplicità.

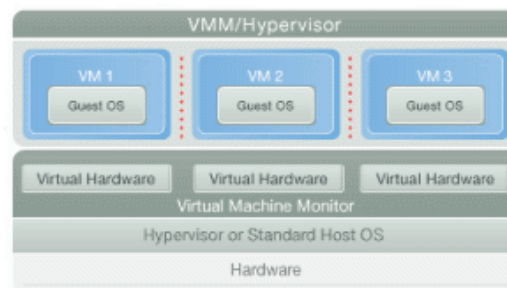


Figura 1.1: Collocazione dell’hypervisor

Gli ambiti di applicazione delle macchine virtuali sono molteplici ed eterogenei fra loro, poichè la virtualizzazione sta diventando sinonimo di sicurezza informatica ed affidabilità del sistema. L'*hypervisor* controlla ed interrompe eventuali attività pericolose e ciò fa sì che si usino macchine virtuali sempre più frequentemente in ambito di ricerca e collaudo di software.

L'*hypervisor* alloca le risorse dinamicamente quando e dove sono richieste, riduce in modo drastico il tempo necessario alla messa in opera di nuovi sistemi, isola l'architettura nel suo complesso da problemi a livello di sistema operativo ed applicativo, rende possibile una gestione più semplice di risorse eterogenee e facilita il collaudo ed il debugging in un ambiente controllato. [5]

1.3 Exokernel

Un *exokernel* è un tipo di *kernel* che non effettua l'astrazione dell'hardware in modelli teorici. È un approccio radicalmente differente alla progettazione dei sistemi operativi. L'idea centrale è "separare la protezione dalla gestione". [6]

L'idea che sta alla base è che nessuno sa come rendere efficiente l'uso dell'hardware disponibile meglio di uno sviluppatore, per cui l'obiettivo è dargli la possibilità di prendere le decisioni. Gli *exokernel* sono estremamente piccoli e compatti, in quanto le loro funzionalità sono arbitrariamente limitate alla protezione ed al multiplexing delle risorse.

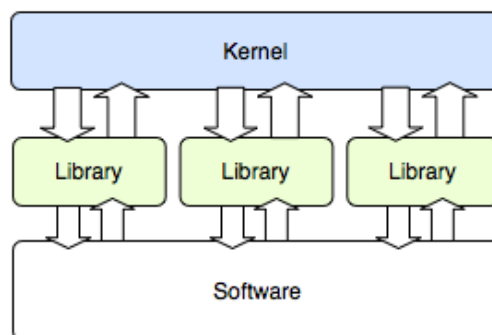


Figura 1.2: Struttura di un Exokernel

I *kernel* classici astraggono l'hardware, nascondendo le risorse dietro ad un livello di astrazione ("*hardware abstraction layer*" o *HAL*), o dietro server sicuri.

Ad esempio, in un sistema classico, se viene allocata della memoria il programma non può sapere in quale pagina fisica questa verrà riservata dal sistema

operativo e se viene scritto un file non c'è modo di sapere direttamente in quale settore del disco è stato allocato. È questo il livello di astrazione che un *exokernel* cerca di evitare. Esso permette ad un'applicazione di richiedere aree specifiche di memoria, settori specifici su disco e così via, assicurandosi solamente che le risorse richieste siano disponibili e che le applicazioni vi possano accedere. [7]

Dato che un *exokernel* fornisce un'interfaccia a basso livello dell'hardware, mancando di qualsiasi funzionalità di alto livello tipica degli altri sistemi operativi, esso è accompagnato da un sistema operativo-libreria, detto *libOS*, che si interfaccia con l'*exokernel* sottostante fornendo agli sviluppatori le funzionalità di un sistema operativo completo.

Tutto ciò ha un'importante implicazione: è possibile avere diversi *libOS* sul sistema. Se, per esempio, si installa un *libOS* che esporta un'API *Unix* ed uno che esporta un'API *Windows*, è possibile eseguire simultaneamente applicazioni compilate per *UNIX* e per *Windows*. Lo sviluppo dei *libOS* avviene a livello utente in piena protezione della memoria.

Attualmente, gli *exokernel* sono più che altro dei progetti di ricerca e non sono usati in sistemi operativi commerciali. [8]

1.4 Perché ricorrere alla virtualizzazione?

La tecnologia di virtualizzazione introduce un cambiamento radicale per quanto riguarda flessibilità delle operazioni IT, velocità di distribuzione, prestazioni delle applicazioni e disponibilità. La virtualizzazione permette di offrire un numero maggiore di funzionalità e, al tempo stesso, la possibilità di acquisire il controllo sui costi. Inoltre, permette di sviluppare ogni giorno nuove ed interessanti modalità di impiego per gli ambienti virtualizzati. Con la virtualizzazione, le risorse di dati e di elaborazione, server e storage, vengono raggruppate in maniera logica in un unico pool di risorse. [9]

I motivi che rendono la virtualizzazione un buon investimento sono numerosi, i principali sono descritti di seguito:

- **“consolidamento ed ottimizzazione”**: quando le tecnologie di virtualizzazione non erano ancora mature si è assistito al proliferare di sistemi nelle server farm, al fine di garantire adeguati livelli di servizio ed al contempo il corretto isolamento tra contesti differenti. Ad oggi è possibile consolidare i sistemi, mantenendo gli stessi livelli di sicurezza e performance.

È possibile consolidare numerose macchine virtuali su un numero limitato di sistemi e successivamente ripartizionare a piacimento le stesse in fun-

zione del carico (manualmente o in modo automatico) senza interrompere l'operatività delle singole VM. Strumenti specializzati di gestione possono accendere, spegnere e spostare le VM in base ad eventi configurabili. In tal modo è possibile ridurre la quantità dell'hardware ed ottenere un risparmio tangibile;

- **“disaccoppiamento tra hardware e servizi”**: ottenuto introducendo tra la vista logica e la vista fisica dell'hardware un livello di indirectione (*VMM*), la cui realizzazione dipende dal tipo di virtualizzazione che si intende adottare.

Grazie a questa proprietà è possibile sostituire l'hardware o migrare le applicazioni con grande facilità ed immediatezza;

- **“flessibilità”**: esiste la possibilità di convertire una macchina fisica in virtuale e viceversa, clonare macchine per differenti fini, spostare una macchina virtuale da un host ad un altro senza interromperne l'esecuzione, clusterizzare lo stesso strato di virtualizzazione per permettere la migrazione automatica delle macchine virtuali in caso di failure dell'hardware host;
- **“backup, disaster recovery, archiviazione”**: possibilità di eseguire tali azioni manualmente o in modalità schedulata senza interruzione del servizio. Qualsiasi sistema è tipicamente in continua evoluzione e spesso nasce l'esigenza di mantenere multiple versioni di tale sistema, tramite le quali rendere possibile un roll-back o eseguire uno o più fork per differenti finalità.

Il disaster recovery risulta notevolmente semplificato sfruttando la virtualizzazione: effettuare il restore di una VM è immediato e sicuro, rendendo possibile il ripristino dei servizi in un tempo molto limitato.

Anche le strategie di backup traggono dei benefici: effettuare il backup di una VM consiste nel copiare un numero estremamente limitato di file (il/i file che compone/compongono il disco virtuale ed i file di configurazione), il che permette un sistema di backup semplificato (parlando di backup della VM, non a livello di filesystem);

- **“isolamento”**: ogni VM può raggiungere alti livelli di isolamento, a seconda che si tratti di virtualizzazione secondo il paradigma delle macchine virtuali o paravirtualizzazione. Sfruttando le macchine virtuali, si ottiene un'isolamento completo e trasparente per i sistemi *guest*, come se ognuno di essi girasse su dell'hardware dedicato;

- **“supporto al networking”**: tramite i software di virtualizzazione disponibili attualmente è possibile emulare differenti segmenti di reti, switching e routing. Per tale motivo, anche complesse architetture composte da segmenti di frontend, backend, DMZ e management possono essere virtualizzate ed integrate nella VLAN reali dando potenzialmente la possibilità di virtualizzare interi datacenter;
- **“clustering”**: virtualizzando i server che faranno parte di un cluster, non solo sarà possibile effettuare un bilanciamento del carico a livello di VM, ma ci sarà la possibilità di bilanciare il carico dei processi che girano all’interno di ogni singola VM (dal momento che il cluster è costruito sulle VM e non sulla macchina *host*).
- **“sviluppo e testing”**: in alcuni contesti è comune la necessità di adoperare numerosi sistemi per brevi periodi di tempo, al fine di effettuare *porting* e test di compatibilità. La virtualizzazione, anche in questo contesto, è un’ottima soluzione sia tecnica che economica che permette di avere a disposizione ambienti di test operativi in poco tempo (ad esempio clonando una VM esistente) su cui effettuare tutte le prove necessarie in sicurezza;
- **“virtual appliance”**: un altro aspetto da non sottovalutare è il possibile mercato di VM preconfigurate (su media e/o web), vendute con soluzioni ad hoc o secondo modelli standard in pronta consegna. [10]

Dopo la breve carrellata dei vantaggi principali ed immediati derivanti dall’adozione della virtualizzazione, nei prossimi capitoli si parlerà di: *“virtualizzazione tramite macchine virtuali”*, *“paravirtualizzazione”*, *“storage e filesystem per macchine virtuali”* ed *“ultimi arrivi sul mercato”*.

Capitolo 2

Macchine virtuali

Le macchine virtuali emulano l'hardware reale e richiedono risorse fisiche dall'host su cui gira il software di virtualizzazione.

Un software d'emulazione è un programma che permette l'esecuzione di software scritto per un ambiente (hardware o software) diverso da quello sul quale l'emulatore viene eseguito. Tale approccio permette di eseguire un qualsiasi sistema operativo *guest*, poichè esso non è cosciente di girare su hardware virtuale. La complicazione, in tale caso, è dovuta al fatto che alcune istruzioni di CPU richiedono privilegi aggiuntivi e non possono essere eseguite in user space, perciò si utilizza, come già visto in precedenza, l'*hypervisor* (detto anche "*virtual machine monitor*" o "*VMM*") per analizzare e consentire in tempo reale tali istruzioni.

Esiste una linea di pensiero che tende a distinguere gli emulatori di un sistema dai simulatori di tale sistema, ed entrambi dalle macchine virtuali. Sebbene il risultato finale possa considerarsi, dagli utenti finali, lo stesso, la differenza tra i tre è fondamentale.

Un emulatore necessita sempre del software di quel sistema limitandosi quindi ad emulare l'hardware, in modo da poter interpretare correttamente i file contenenti i dump del software (che rimane scritto in un linguaggio comprensibile soltanto al sistema emulato e non al sistema emulante).

Un simulatore, invece, si prefigge di arrivare allo stesso risultato, riscrivendo però (in tutto o in parte) le routine del programma da simulare, in modo da renderlo comprensibile alla macchina su cui gira.

Non essendoci alcuna emulazione dell'hardware, cosa che necessita di diversi cicli-macchina, un simulatore è per forza di cose più veloce, tuttavia è spesso poco preciso nel riproporre fedelmente il software simulato. Per tali motivi, se i simulatori hanno dalla loro una notevole velocità di esecuzione, gli emulatori hanno l'accuratezza (che comprende, ovviamente, anche eventuali bugs del software e/o della macchina originale).

Le *virtual machine*, infine, si possono considerare un PC nel PC. Ossia, mediante una *virtual machine* è possibile installare un secondo sistema operativo in una macchina virtuale e farci girare software in un ambiente considerato più “protetto” che non la macchina *host* vera e propria. Tali sistemi a volte emulano anche parti hardware, altre volte si limitano a replicare l’hardware della macchina *host*. Tuttavia, non emulano mai macchine con architettura completamente diversa (a partire dalla CPU).

2.1 VMware

VMware Inc. è una tra le maggiori società che distribuisce software di virtualizzazione per sistemi x86 sia a 32 che a 64 bit. I prodotti disponibili sono: *VMware Workstation*, *VMware Infrastructure*, *VMware Server* e *VMware Player*. Gli ultimi due sono freeware, scaricabili ed utilizzabili gratuitamente, mentre i primi due sono a pagamento, dedicati alla classe enterprise.

L’istanza del computer e del sistema operativo che esegue *VMware* è detta “*host*”, mentre le istanze dei sistemi che girano dentro le *macchine virtuali* sono dette “*guest*”. Come un emulatore, ogni software della famiglia *VMware*, rende disponibile un completo set di hardware virtualizzato al sistema *guest*.

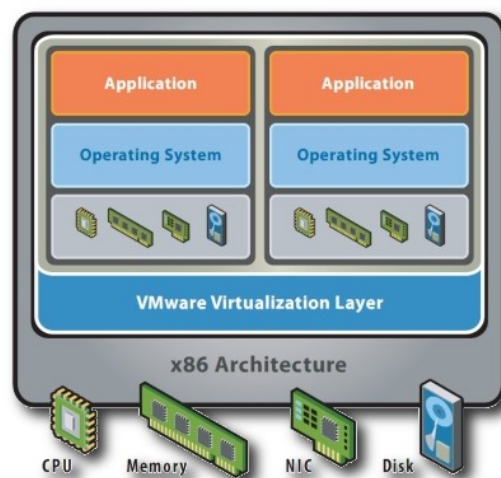


Figura 2.1: Struttura di VMware

Grazie al fatto che ogni sistema *guest* usa gli stessi driver indipendentemente dall’hardware su cui gira l’*host*, le VM sono estremamente portabili da un sistema ad un altro. Il primo vantaggio che si può intuire immediatamente è

la possibilità di poter implementare su una piattaforma con sistema operativo *Microsoft Windows* o *GNU/Linux* (con relativo hardware) un numero N di VM con sistema operativo diverso fornendo una completa emulazione dell'hardware senza intaccare le risorse della macchina *host*.

È plausibile pensare che le prestazioni del sistema operativo emulato possano essere inferiori rispetto a quelle che ha il sistema operativo eseguito realmente, ma in realtà le prestazioni sono comparabili perché il sistema è pensato per essere eseguito su macchine multiprocessore: ogni VM lavora con uno o più core, o li gestisce in real-time in base a criteri di condivisione offrendo quindi prestazioni elevate.

A seconda della versione utilizzata, *VMware* è in grado di emulare architetture x86-32, x86-64 e processori multi-core.

VMware distribuisce due prodotti per datacenter: *VMware ESX* e *GSX* (ora chiamato *VMware server* e distribuito gratuitamente). La differenza sostanziale tra l'ESX e la versione "server" è che quest'ultima necessita di un sistema operativo *host* ospitante, mentre la versione ESX è un sistema operativo a se stante ed è necessaria l'installazione su quello che *VMware* definisce "*the bare metal*". Ciò comporta un minor overhead di elaborazione, dato dal fatto che è il software di virtualizzazione stesso a gestire l'hardware direttamente. Dall'altro lato, la versione server (*GSX*) ha un più vasto supporto a livello di periferiche PCI ed USB.

Nelle prossime pagine verranno illustrate le singole soluzioni di virtualizzazione offerte da *VMware*.

2.1.1 VMware Player

È il software entry-level della famiglia *VMware*, ottenibile direttamente online ed utilizzabile gratuitamente. È disponibile sia per *Microsoft Windows* sia per *GNU/Linux* ed attualmente è alla versione 2.0.

Nonostante sia il software di più basso livello, *VMware Player* rende semplice l'utilizzo sia di VM create da *VMware Workstation*, *VMware Server* e *VMware ESX Server* sia di VM *Microsoft Virtual PC* e sia di *Symantec LiveState Recovery disks*, estendendo la compatibilità a prodotti di terze parti.

VMware Player è una classica applicazione standalone, la quale una volta installata fa girare la VM in una finestra separata e con cui è possibile configurare i permessi di accesso ai dispositivi fisici del sistema. Non è possibile però creare nuove VM, ma solamente utilizzare quelle già esistenti.

Le funzionalità principali di questo software sono:

- possibilità di far girare VM con sistemi operativi a 32 e 64 bit, tra cui

Microsoft Windows, Red Hat Linux, SUSE Linux, Ubuntu GNU/Linux, Debian GNU/Linux, FreeBSD e Solaris Sun;

- gli utenti possono far girare VM con CPU multiple, testando le potenzialità e le performance del lavoro con sistemi multi-threaded. È inoltre possibile emulare un sistema multiprocessore per le VM anche avendo a disposizione un solo processore a singolo core;
- apertura di immagini di VM *Microsoft virtual machine* e *Symantec Backup Exec* (formalmente chiamate *Live State Recovery*). Durante tale processo, l'immagine della VM iniziale non viene modificata mantenendo il suo formato nativo: ogni modifica viene salvata in un file a parte di *VMware*, il quale alla fine viene linkato all'immagine iniziale garantendo una piena compatibilità;
- condivisione di dati tra il sistema *host* ed il sistema *guest* tramite le pratiche e comuni funzioni “drag and drop”;
- ampio supporto sia per sistemi *host* sia per sistemi *guest* a 32 e 64 bit (attualmente oltre 60 ufficialmente supportati);
- supporto per dispositivi USB 2.0 ad alta velocità;
- accesso alle VM tramite un'intuitiva interfaccia con la possibilità di aprire VM esistenti o ricercarne di nuove nei cataloghi *VMware Virtual Appliance Marketplace* esistenti e messi a disposizione online.

I requisiti minimi consigliati da *VMware* (esclusa la CPU, la memoria e lo spazio su disco utilizzati dalle singole VM) per far girare correttamente *VMware Player* sono:

- CPU a 400MHz (consigliata 500MHz);
- 512Mb di memoria RAM (consigliata 1Gb);
- 150Mb di spazio su disco per l'installazione del software.

2.1.2 VMware Server

VMware Server è l'applicativo gratuito con supporto di classe enterprise dedicato alla virtualizzazione ed al consolidamento dei server. È installato e gira come una normale applicazione, sia su sistemi *Microsoft Windows* sia su sistemi *linux-based*. Lo strato di virtualizzazione partiziona il server fisico in modo che molte VM possano girare simultaneamente su un singolo server.

Le risorse elaborative del server fisico sono viste come un unico pool di risorse allocabili in maniera sicura alle VM. *VMware Server* isola ogni singola VM dal sistema ospitante e dalle altre macchine virtuali presenti, proteggendola da eventuali crash delle altre VM. I dati non transitano tra le VM e le applicazioni possono comunicare solo tramite connessioni di rete configurate e funzionanti.

VMware Server incapsula l'ambiente di ogni VM (dati e configurazioni) come un set di file, estremamente pratici da spostare, copiare ed effettuare backup.

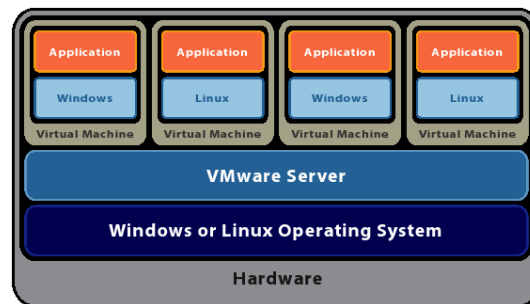


Figura 2.2: Struttura di VMware Server

Le principali funzionalità rese disponibili da *VMware Server* sono:

- gira su qualunque hardware x86-compatibile;
- supporto per sistemi *guest* a 64 bit, inclusi *Microsoft Windows*, *GNU/Linux* e *Sun Solaris*;
- supporto “SMP virtuale biprocessore” che offre la possibilità ad una singola VM di sfruttare più processori fisici;
- “snapshot” dell'intero stato di una VM ed eventuale “rollback” in qualsiasi momento;
- wizard per la creazione di nuove VM;
- apertura di immagini di VM *Microsoft virtual machine* e *Symantec Live State Recovery* tramite il *VM Importer* (software di conversione di VM di terze parti in formato *VMware*);
- supporto per “*IVT*” (*Intel Virtualization Technology*), l'estensione di virtualizzazione *Intel* direttamente inclusa nel processore;

- supporto per hard-disk e lettori CD/DVD sia IDE che SCSI;
- sistemi operativi *guest* possono risiedere su file di dischi virtuali o su partizioni fisiche (*raw*) del disco;

A differenza di *VMware Player*, nel caso di *VMware Server*, vengono definiti anche le risorse massime gestibili dall'ambiente:

- supporto di server multiprocessore fino a 16 vie con processori a 32 o 64 bit;
- 64 VM concorrenti per singolo host (consigliate non più di 4 per ogni processore);
- fino ad 8Gb di memoria RAM per singola VM;
- CPU x86 a 733MHz che supporta l'instruction set del Pentium. I processori ufficialmente supportati sono: Pentium II, Pentium III, Pentium 4, Pentium M Xeon, EM64T, Athlon, Athlon MP, Athlon XP, AMD Opteron, AMD Athlon 64, Turion 64 (supporto sperimentale per AMD Sempron);
- i processori dual-core sono supportati e vengono contati come un solo processore a livello di licenza d'uso;
- memoria RAM: minima consigliata 512Mb (massima gestita 64Gb);
- spazio su disco: installazione su *Microsoft Windows* 250Mb, installazione su *GNU/Linux* 200Mb;

È importante infine accennare a *VMware Virtual Center*, l'interfaccia centralizzata di gestione di *VMware Server*. A differenza della versione server, non è gratuita, ma il prezzo è stato pensato per piccole e medie aziende che non si affidano all'infrastruttura completa, quindi risulta molto più accessibile. Questa interfaccia gestisce contemporaneamente più installazioni di *VMware Server*, agendo da "server di controllo" per tutto il sistema.

Il "Virtual Center" necessita però di alcuni componenti software aggiuntivi:

- **"VirtualCenter Management Server"**: software che gira come servizio di *Microsoft Windows 2000*, *Microsoft Windows XP Professional* o *Microsoft Windows Server 2003* interagendo con l'infrastruttura hardware e software già presente;
- **"VirtualCenter Database"**: costituito dal database di gestione di VirtualCenter, che gira su *Oracle* o su *Microsoft SQL Server*;

- “**Virtual Infrastructure Client**”: client di gestione dell’intera infrastruttura che gira su qualsiasi versione sufficientemente recente di Windows e che si collega al “VirtualCenter Management Server”;
- “**VirtualCenter Agent**”: connette ogni singola installazione di *VMware Server* con il relativo “VirtualCenter Management Server” al fine di renderne possibile la gestione centralizzata.

2.1.3 VMware Workstation

VMware Workstation, attualmente alla versione 6.0, è il software di punta per quanto riguarda la virtualizzazione desktop. È in grado di creare e far girare multiple VM contemporaneamente, convertire macchine fisiche in macchine virtuali, passare da un sistema operativo ad un altro istantaneamente, condividere file tra il sistema *host* ed il sistema *guest* tramite le funzionalità “drag-and-drop” ed accedere a tutte le periferiche a disposizione sulla macchina ospitante.



Figura 2.3: Struttura di VMware Workstation

VMware Workstation include tutte le funzionalità di *VMware Player*, estendendole. Il motivo per cui non è gratuito è da ricercarsi tra le svariate funzionalità avanzate che comprende:

- “**ampio supporto di sistemi *host* e *guest***”: gira sia su *Microsoft Windows* sia su *GNU/Linux* e supporta come sistemi *guest* la maggior parte delle versioni desktop e server di *Microsoft Windows*, *GNU/Linux*, *Sun Solaris x86*, *Novell Netware* e *FreeBSD*. Sia il sistema ospitante che le VM possono utilizzare sistemi operativi sia a 32 che a 64 bit. Supporta “two-way Virtual SMP”, che consiste nell’assegnamento di uno o due

processori ad una singola VM; inoltre è stato introdotto un supporto sperimentale per kernel *GNU/Linux* paravirtualizzati;

- **“avanzata architettura per le VM”**: gestione fino ad 8Gb di memoria RAM per singola VM, driver audio a 64 bit, possibilità di creare cloni di VM esistenti, supporto sperimentale per “registrare” e “ripetere” le azioni effettuate su un sistema *guest*;
- **“opzioni avanzate di networking”**: impostazioni di networking predefinite che comprendono NAT, bridge, ed host-only o personalizzabili dall’utente. Server DHCP integrato, fino a 10 switch virtuali per connettere VM con VM, VM con *host* e VM con reti esistenti. Possibilità di specificare banda dedicata alle VM o la percentuale di perdita di pacchetti accettabile, generando quindi una specie di traffic shaper virtuale;
- **“operatività a livello desktop”**: è possibile configurare una VM affinché utilizzi più monitor o fare in modo che multiple VM utilizzino monitor separati; inoltre la risoluzione dello schermo delle VM varia automaticamente in base alla dimensione della finestra in cui sono visualizzate oppure c’è la possibilità di farle girare in background senza sfruttare l’interfaccia di *VMware Workstation*.

È possibile creare directory condivise e funzionalità “drag-and-drop” tra i vari sistemi attivi. Inoltre, possono essere utilizzati dispositivi USB 2.0 ad alta velocità e le unità Windows mappate tramite lettere possono essere associate a dischi virtuali.

Infine è possibile catturare tutta l’attività sullo schermo della VM come una registrazione video ed eseguirne il playback.

- **“utilità di sviluppo e testing”**: debug applicazioni nel sistema *guest*, accedendo ad un processo in esecuzione per poi fare un “rollback” dopo le operazioni di debugging (integrazione con *Microsoft Visual Studio* ed *Eclipse* integrata).

Cattura e gestione di copie “point-in-time” di VM in esecuzione e “rollback” dei cambiamenti grazie alla funzionalità degli “snapshot” multipli.

Supporto per *VNC* built-in che dà la possibilità a client remoti di connettersi alle VM in esecuzione.

Disponibili “*API VIX Automation*” per eseguire qualsiasi comando sulla VM tramite script.

2.1.4 VMware Infrastructure

VMware Infrastructure è il prodotto di classe enterprise più completo sviluppato da *VMware*. Permette di virtualizzare server e storage e gestire l'intero sistema come un pool di risorse condivise al fine di sviluppare proprietà di alta affidabilità, mobilità e bilanciamento del carico.

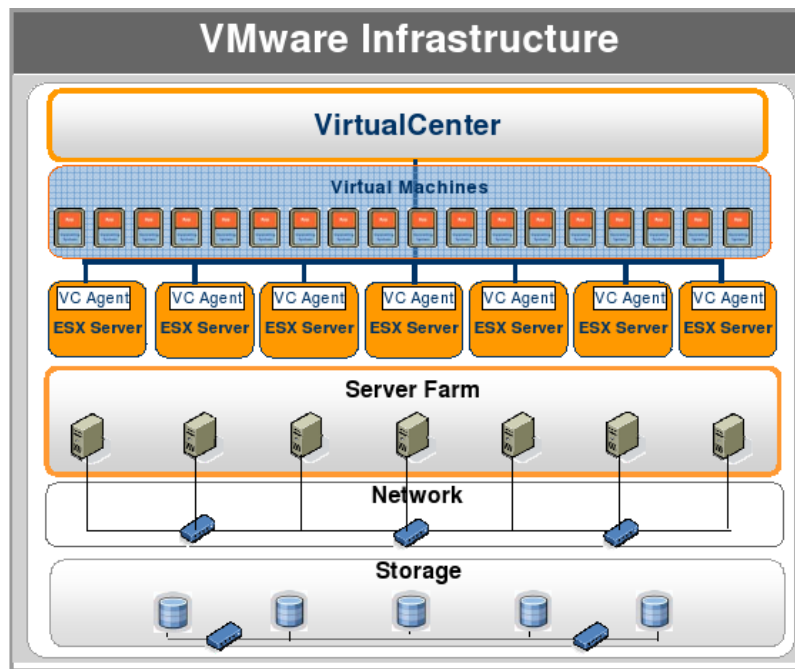


Figura 2.4: Struttura di VMware Infrastructure

Attualmente alla versione 3, è composto da una serie di pacchetti atti a soddisfare richieste diverse ma complementari:

- “*VMware ESX Server*”: è un *hypervisor* a livello di macchina fisica, installato cioè come sistema operativo standalone su un server e non più come un software che necessita di un ambiente già operativo. Ciò comporta un notevole aumento nelle prestazioni, rendendole molto vicine a quelle dell'equivalente server fisico.

Ogni singola installazione di “*ESX Server*” viene poi interfacciata con il software di gestione dell'intera infrastruttura, rendendo possibile un controllo centralizzato di tutti i server e di tutte le risorse disponibili;

- “*VMware Virtual SMP*”: dà la possibilità ad ogni singola VM di usare fi-

no a quattro processori fisici contemporaneamente. Questa caratteristica di “symmetric multiprocessing” è fornita unicamente da *VMware*;

- “*VMware VMFS*”: filesystem ad alte prestazioni orientato al clustering ed ottimizzato per macchine virtuali. È il filesystem di default per lo storage delle VM su dischi e partizioni SCSI, gestito in maniera centralizzata che assicura controllo, flessibilità e performance; dà la possibilità ad ogni server ESX di attingere allo storage condiviso in maniera concorrente;

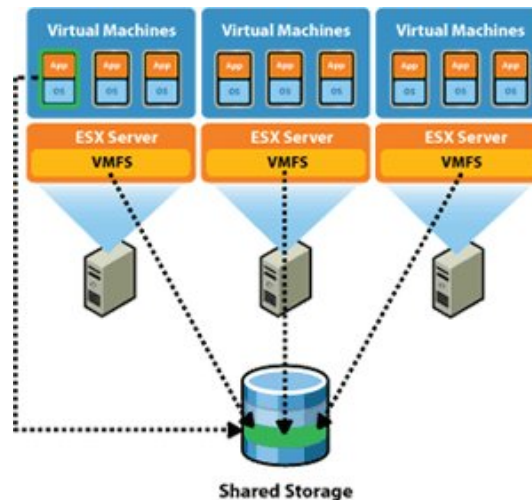


Figura 2.5: Struttura di VMware VMFS

- “*VMware DRS*”: acronimo di “*Distributed Resource Scheduler*” che si occupa di allocare e bilanciare dinamicamente le risorse a disposizione. Monitorizza costantemente l’utilizzo del pool di risorse allocando intelligentemente quelle disponibili tra le varie VM basandosi su regole predefinite. Quando le risorse necessarie al corretto funzionamento di una o più VM sono troppo scarse, avviene una migrazione “live” verso uno dei server fisici più scarichi (grazie alla tecnologia *VMware VMotion*);
- “*VMware High Availability*”: si occupa di monitorare e gestire i problemi hardware dei server fisici che compongono l’infrastruttura; quando viene rilevato il down di uno dei nodi, tutte le VM vengono migrate verso gli altri server disponibili, minimizzando il downtime e contemporaneamente ottimizzando il carico per ogni singolo nodo. Nell’ultima versione disponibile, rilasciata nel novembre 2007, è stato introdotto un componente sperimentale che non solo monitorizza lo stato dei server fisici, ma tenta di capire se vi sono problemi all’interno delle singole macchine virtuali

(es esempio crash di sistema, risorse insufficienti...) e, se rilevati, decide di riavviare la VM;

- “*VMware VMotion*”: grazie al *VMware VMFS cluster filesystem*, gli stati di tutte le VM sono incapsulati in un set di file presenti nello storage condiviso, dando quindi la possibilità ai vari server ESX di accedervi concorrentemente. Inoltre, lo stato della memoria e di esecuzione può essere trasmesso rapidamente su un network sufficientemente veloce.

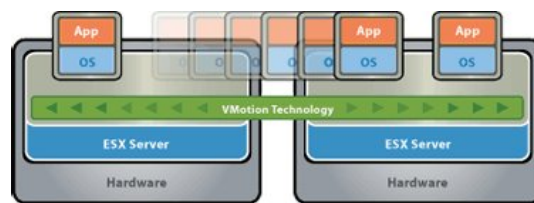


Figura 2.6: Struttura di VMware VMotion

Tutto ciò permette lo spostamento di una VM in esecuzione da un server all'altro istantaneamente e senza la necessità di riavviarla, caratteristiche sfruttate da *VMware High Availability* e *VMware DRS*;

- “*VMware Storage VMotion*”: è l'evoluzione di *VMware VMotion* applicata allo storage condiviso.

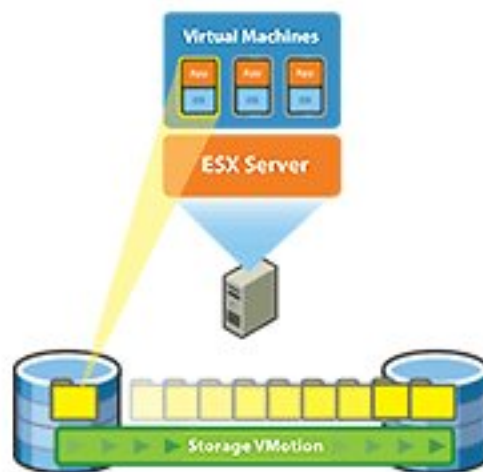


Figura 2.7: Struttura di VMware Storage VMotion

Offre la possibilità di eseguire una migrazione in runtime dei dischi delle macchine virtuali attraverso array di storage eterogenei, assicurando integrità dei dati e nessuna interruzione di servizio.

Oltre a quelli descritti precedentemente, esistono altri componenti più comuni presenti in *VMware Infrastructure*, in particolare *VMware Consolidated Backup* e *VMware Update Manager*.

Questi prodotti, come intuibile dai relativi nomi, aiutano l'amministratore del sistema a mantenere aggiornata l'intera infrastruttura semplificandone la gestione.

“Update Manager” integra la procedura ottimale di aggiornamento per ogni singolo server: migrazione delle VM presenti nel nodo da aggiornare verso gli altri server del sistema, installazione degli aggiornamenti disponibili e ripristino della situazione iniziale riportando sul nodo aggiornato le VM spostate in precedenza.

“Consolidated Backup” invece semplifica notevolmente la gestione ed il ripristino dei backup delle VM attraverso un server ed un componente software dedicati a tale scopo.

2.2 Virtualizzazione nel mondo Microsoft

Il settore virtualizzazione di *Microsoft* si suddivide in due specifici prodotti: *Microsoft Virtual Server* e *Microsoft Virtual PC*.

Il primo è dedicato alla classe d'utenza enterprise, con soluzioni per aziende di medio-grandi dimensioni che necessitano di un'infrastruttura virtuale paragonabile a quella messa a disposizione da *VMware*. Il pregio dei prodotti di virtualizzazione disponibili direttamente da *Microsoft* è la grande integrazione col resto dei suoi prodotti.

Il secondo è orientato all'utenza desktop e si colloca tra *VMware Player* e *VMware Workstation*. Le funzionalità di cui dispone sono decisamente superiori al semplice “player” fornito da *VMware*, ma non raggiungono quelle fornite dalla versione “workstation”. Questo software è particolarmente indicato per chi ha la necessità di far girare applicazioni legacy che col passare del tempo non hanno visto un porting verso i sistemi operativi di ultima generazione.

Di seguito verranno illustrati entrambi i prodotti, analizzandone la struttura e le caratteristiche principali.

2.2.1 Microsoft Virtual Server

Microsoft Virtual Server, attualmente alla versione “2005 R2”, è uno strumento che consente il consolidamento di più ambienti di lavoro in un server fisico,

per assicurare un utilizzo più efficiente delle risorse hardware. Permette di aumentare la produttività e distribuire rapidamente nuovi server in base alle esigenze, implementando e configurando automaticamente macchine virtuali amministrabili tramite strumenti standard di gestione server. *Virtual Server* è una soluzione di virtualizzazione testata e supportata, sia sui sistemi operativi e sulle applicazioni server *Microsoft*, sia sui prodotti di terze parti.

Consente di creare macchine virtuali portabili, connesse e completamente controllabili tramite script, al fine di automatizzare la configurazione e il deployment. Inoltre incapsula le macchine virtuali in dischi rigidi virtuali detti *VHD* (acronimo di *Virtual Hard Disk*), mentre le funzionalità per le reti virtuali consentono di creare reti flessibili per la connettività tra guest e guest, tra guest ed host e tra guest e rete esistente.

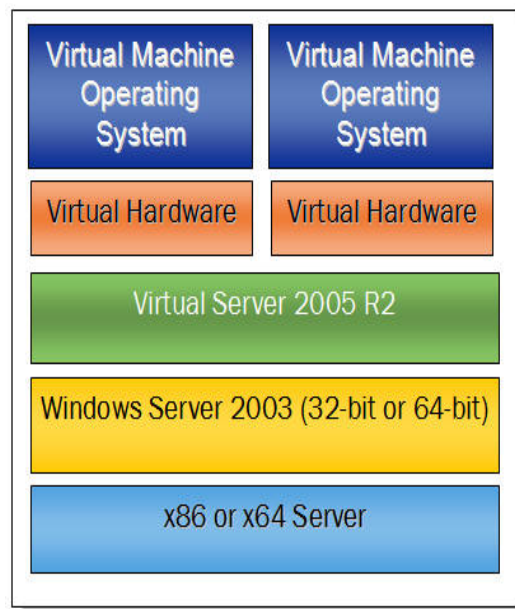


Figura 2.8: Struttura di Microsoft Virtual Server

Il sistema operativo host *Windows Server 2003* sfrutta le caratteristiche di amministrazione dei sistemi *Windows* per la gestione degli ambienti “Virtual Server”: l’integrazione con *Active Directory* permette la delega dell’amministrazione e l’autenticazione dell’accesso guest, la console web consente l’amministrazione autenticata e l’accesso client remoto e *Microsoft Systems Management Server* permette il rilevamento delle relazioni tra macchine virtuali e fisiche a scopo di inventario.

Virtual Server è ampiamente supportato da tutti i sistemi operativi *Win-*

dows Server e le applicazioni server Microsoft. Le macchine virtuali utilizzano i driver di periferica del sistema host, garantendo un supporto stabile per i dispositivi ed un'ampia compatibilità. In linea con i criteri *Common Engineering* per *Windows Server System*, *Microsoft Virtual Server* è supportato da tutti i prodotti server, ognuno dei quali deve poter essere eseguibile in un'istanza virtuale; dispone inoltre di un ampio ecosistema di supporto: ISV ("Independent Software Vendor") e clienti possono sfruttarne le caratteristiche di integrazione per ottenere un alto livello di funzionalità e gestibilità. I produttori di strumenti di gestione pubblicano e utilizzano complete interfacce di gestione COM, e per assicurare agli ISV un'interoperabilità nativa con "Virtual Server", *Microsoft* fornisce il formato di file *VHD* tramite una licenza senza royalty.

Supporta soluzioni di virtualizzazione ad alte prestazioni in un ambiente Microsoft. La vasta compatibilità con i sistemi operativi guest x86 permette di installare nelle VM la maggior parte di tali sistemi. Eseguito in modo nativo nei sistemi host Windows a 64 bit, *Virtual Server* supporta l'allocazione delle risorse CPU in base a pesi predefiniti assegnati alle VM o tramite l'impostazione di limiti ed il ridimensionamento della memoria in fase di avvio della macchina virtuale.

La funzionalità *ADS* (*Automated Deployment Services*) ed il *Virtual Server Migration Toolkit* offrono strumenti eseguibili da riga di comando per la conversione da macchine fisiche a virtuali o da macchine virtuali a virtuali, semplificandone la migrazione. L'*API COM* completa consente il controllo delle macchine virtuali tramite script e la tecnologia *PXE* rende il provisioning delle macchine virtuali identico a quello dei server fisici.

Microsoft dichiara quelli elencati di seguito come requisiti minimi per far girare *Virtual Server*:

- processori supportati: Celeron, Pentium III, Pentium 4, Xeon, Opteron, Athlon, Athlon 64, Athlon X2, Sempron o Duron;
- velocità minima CPU: 550MHz (consigliata 1GHz)
- numero di processori: da 1 a 4 per la "Standard Edition" e fino a 32 per la "Enterprise Edition";
- memoria RAM minima: 256Mb;
- spazio su disco: 2Gb;
- monitor: Super VGA (800x600) o superiore;

- sistemi operativi host supportati: *Microsoft Windows Server 2003* Standard Edition, Standard Edition SP1, Standard x64 Edition, Enterprise Edition, Enterprise Edition SP1, Enterprise x64 Edition, Datacenter Edition, Datacenter Edition SP1, *Microsoft Windows XP* Professional SP2, Professional x64 Edition, *Microsoft Windows Small Business Server 2003* Standard Edition, Premium Edition, Standard Edition SP1, Premium Edition SP1.

2.2.2 Microsoft Virtual PC

Microsoft Virtual PC, attualmente alla versione “2007”, è una soluzione a basso costo la creazione di macchine virtuali per *Windows* rilasciata da *Microsoft Corporation*. Basato su una tecnologia sviluppata da *Connectix Corp.*, “Virtual PC” è l’unica soluzione per la creazione di macchine virtuali basata su client completamente supportata da *Microsoft*. Nel mese di agosto 2003 *Microsoft* ha inoltre rilasciato la versione di “Virtual PC” per *Macintosh*.

Con “Virtual PC” è possibile eseguire due o più sistemi operativi contemporaneamente sullo stesso computer. Dopo l’installazione, l’utente può creare una o più macchine virtuali, in ognuna delle quali è possibile installare e configurare un sistema operativo *guest* e le relative applicazioni. Il numero delle macchine virtuali che è possibile creare ed eseguire simultaneamente nello stesso computer fisico dipende dalla quantità di memoria disponibile.

Ogni macchina virtuale emula, in un ambiente isolato ed autocontenuto, un computer standalone dotato di un processore specifico e di una propria combinazione di schede audio, video e di rete. Le modifiche apportate ad una macchina virtuale non hanno alcun effetto sul computer fisico in cui viene eseguita. “Virtual PC” offre una struttura basata su file per l’archiviazione della configurazione di ogni macchina virtuale ed utilizza dischi virtuali per eseguire i sistemi operativi e fornire lo spazio di archiviazione per le VM. Poiché grazie a questa struttura non è necessario suddividere il disco in partizioni per isolare i sistemi operativi, è possibile eseguire più sistemi operativi concorrentemente e passare dall’uno all’altro senza riavviare il computer.

L’utilizzo più classico per cui è stato pensato *Microsoft Virtual PC* è il supporto per applicazioni legacy che, in caso di migrazione, consente agli utenti di continuare temporaneamente a utilizzare applicazioni meno recenti mentre è in corso la migrazione ad un nuovo sistema operativo senza che sia necessario sostituirle, aggiornarle o acquistare dispositivi aggiuntivi al solo scopo di supportarle. Oltre che in caso di migrazione, l’utilizzo di più sistemi operativi si rende necessario anche in altri scenari, come ad esempio nell’ambito dei servizi di supporto tecnico e controllo qualità. Nel caso del supporto tecnico, “Virtual PC” aiuta il personale a fornire risposte più tempestive ed a ridurre la

durata delle chiamate, permettendo ai tecnici di avviare quasi istantaneamente una macchina virtuale che corrisponde alla configurazione dell'utente, anzichè gestire più computer o riavviare uno stesso computer ad ogni richiesta di supporto. "Virtual PC" consente inoltre ai responsabili del testing di verificare il funzionamento delle applicazioni molto più rapidamente e con una più ampia gamma di configurazioni.

I requisiti minimi dichiarati da *Microsoft* per l'utilizzo di "Virtual PC" sono i seguenti:

- processori supportati: Athlon, Duron, Celeron, Pentium II, Pentium III, Pentium 4, Core Duo, Core2 Duo ("Virtual PC" può girare su un computer multiprocessore, ma ne utilizzerà solo uno);
- velocità minima CPU: 400MHz (consigliata 1GHz)
- memoria RAM minima: 256Mb;
- spazio su disco: 2Gb;
- monitor: Super VGA (800x600) o superiore;
- sistemi operativi host supportati: *Microsoft Windows Vista* Business, Enterprise, Ultimate, *Microsoft Windows Server 2003* Standard Edition, Standard x64 Edition, *Microsoft Windows XP* Professional, XP Tablet PC Edition;
- sistemi operativi guest ufficialmente supportati: *Microsoft Windows* 98, 98SE, Millennium Edition, 2000 Professional, XP Home Edition, XP Professional, Vista Enterprise, Vista Business, Vista Ultimate, *IBM OS/2 Warp* Version 4 Fix Pack 15, OS/2 Warp Convenience Pack 1, OS/2 Warp Convenience Pack 2.

2.3 Ultimi arrivi

Nel campo della virtualizzazione, così come nell'informatica in generale, gli aggiornamenti e le novità si susseguono in continuazione ed illustrare lo stato dell'arte spesso è più complesso del previsto. Le novità e le scoperte compaiono di giorno in giorno e spesso rendono quelle precedenti obsolete nel giro di pochissimo tempo. È per questo motivo l'introduzione del capitolo "ultimi arrivi", in cui sono inserite le novità più scottanti riguardando l'infuocato mercato della virtualizzazione.

Con nomi come Oracle e SUN che scendono in campo, non è di certo un mercato piatto, anzi, è continuamente ricco di sorprese, condito da acquisizioni, nuovi prodotti e soluzioni che compaiono con cadenza quasi mensile.

2.3.1 Oracle VM

A metà novembre 2007 anche *Oracle* si è gettata nel mercato della virtualizzazione con una soluzione standalone chiamata “*Oracle VM*”. È disponibile come download gratuito e può essere utilizzata per virtualizzare sia i software della stessa Oracle, sia le applicazioni di terze parti che girano su GNU/Linux o Microsoft Windows. L’azienda non manca di sottolineare come la sua sia l’unica soluzione di virtualizzazione oggi sul mercato a supportare i software da lei prodotti.

OVM si basa sul codice dell’hypervisor gratuito *XEN* e per funzionare non necessita di alcun sistema operativo host (come si può vedere in figura 2.9): grazie ad un kernel GNU/Linux integrato, il software è in grado di **girare direttamente sull’hardware** di server x86 a **32 o 64 bit** e di eseguire, all’interno di una macchina virtuale, varie versioni di GNU/Linux (“Oracle Linux” in primis) e di Windows: per far girare quest’ultimo, il software necessita di un processore che supporti la virtualizzazione a livello hardware. Oltre al server vero e proprio, Oracle VM include un’**interfaccia di gestione web-based** da cui è possibile creare, clonare, condividere, configurare, avviare e migrare le macchine virtuali.

Oracle ha rilasciato il proprio software di virtualizzazione sotto forma di **immagine ISO masterizzabile su di un singolo CD**. L’utilizzo è **completamente gratuito**, anche in ambiti commerciali: l’azienda intende guadagnare attraverso la vendita di contratti di supporto. Da sottolineare come la licenza di Oracle VM garantisca agli utenti un **indennizzo per eventuali cause legali relative alla violazione di copyright**.

Sebbene oggi il mercato della virtualizzazione sia dominato da *VMware*, il primo rivale di Oracle rimane anche in questo caso Microsoft: quest’ultima è infatti una delle poche aziende che, come Oracle, è in grado di fornire ai propri clienti una singola fonte di supporto per ambiente di virtualizzazione, sistema operativo, database, middleware ed application server. [36]

2.3.2 Sun xVM

Verso la metà di dicembre 2007, *SUN* ha ufficialmente presentato *xVM Ops Center*, il primo prodotto commerciale della neonata famiglia di soluzioni *xVM* per la virtualizzazione dei datacenter. Il software è stato lanciato sul mercato all’inizio di gennaio 2008, e fa leva sulla stretta integrazione con il sistema

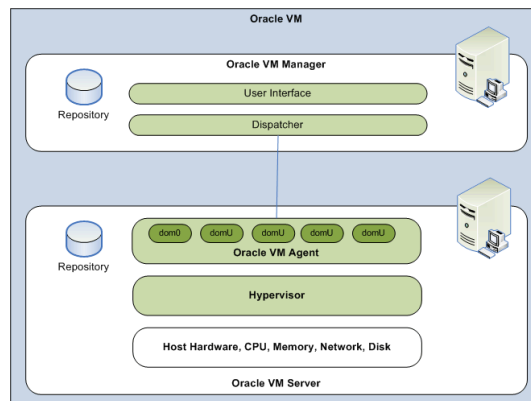


Figura 2.9: Struttura di Oracle VM

operativo di casa SUN, Solaris. SUN ha annunciato per la prima volta xVM a settembre 2007 includendo la prima versione beta nell'*OpenSolaris Developer Preview* di novembre 2007, un'anteprima della successiva release del sistema operativo opensource di SUN.

Componente chiave di questa piattaforma è **xVM Server**, un server gratuito di virtualizzazione progettato per girare su Solaris. xVM Server poggia sul codice di XEN, già alla base delle soluzioni per la virtualizzazione di *Oracle* e *Virtual Iron*. SUN ha spiegato che il nuovo "xVM Ops Center" consente di semplificare aspetti quali rilevamento degli asset, monitoraggio, provisioning del sistema operativo, aggiornamenti software e firmware e gestione delle patch. Il software è in grado di supportare **ambienti multipiattaforma x86 e SPARC basati su GNU/Linux e Solaris**.

Una distribuzione di "xVM Ops Center" è stata rilasciata come download gratuito all'inizio di quest'anno. I programmi di supporto sul software, che includono assistenza e servizi erogati via rete, hanno prezzi che partono da 100 dollari per server gestiti per arrivare a 350 dollari in base al set di funzionalità scelte dal cliente.

SUN inoltre ha creato una comunità, OpenxVM.org, dove pubblicherà progressivamente il codice di alcune delle tecnologie alla base di xVM: al momento l'azienda ha aperto il codice di **Common Agent Container (CAC)**, e prossimamente renderà disponibile quello di Ops Center. Da notare come tutto il codice scritto da SUN (dunque ad esclusione di quello preso a prestito da XEN, che è GPL2) sarà rilasciato sotto GPL3: questa è la prima volta che la mamma di Solaris si avvale della più recente licenza GNU. [37]

2.3.3 Microsoft Hyper-V

Hyper-V, nome in codice “Viridian”, formalmente conosciuto come “**Windows Server Virtualization**” è un sistema di virtualizzazione basato su hypervisor per sistemi **Windows Server 2008 a 64-bit**, che sarà reso disponibile 180 giorni dopo il rilascio del sistema operativo utilizzabile in produzione. “Hyper-V” sarà disponibile anche come offerta standalone, privo delle funzionalità “Windows Server”, sotto il nome di “**Microsoft Hyper-V Server**”. Da sottolineare il fatto che le RC di Windows Server 2008 includono un’anteprima di “Hyper-V”.

“Hyper-V” supporta l’**isolamento a livello di partizione**. Una partizione è un’unità logica di isolamento, supportata dall’hypervisor, entro la quale viene eseguito un sistema operativo. Un’istanza dell’hypervisor necessita di almeno una partizione principale (chiamata *root partition*) in cui gira Windows Server 2008. Lo stack di virtualizzazione gira in tale partizione principale ed ha accesso diretto all’hardware del sistema. Successivamente la “root partition” crea le partizioni figlie che conterranno i sistemi operativi *guest*. Inoltre, una partizione figlia può generare a sua volta altre partizioni figlie sfruttando le *API hypercall* che si interfacciano direttamente con “Hyper-V”.

Una partizione virtualizzata non ha accesso diretto al processore fisico e di conseguenza non gestisce direttamente i suoi interrupt. L’hypervisor gestisce gli interrupt del processore reindirizzandoli alla partizione corretta tramite il “*logical Synthetic Interrupt Controller*” (o *SynIC*).

Come già detto in precedenza, tutte le partizioni figlie non hanno accesso diretto alle risorse hardware, bensì si appoggiano ad una vista virtuale di tali risorse, i *device virtuali*. Ogni richiesta al device virtuale è redirezionata tramite il *VMBus* ai dispositivi della partizione genitore, che andrà a gestire la richiesta. Se il dispositivo richiesto nella partizione genitore è anch’esso un dispositivo virtuale, il ciclo si ripete fino a raggiungere la “root partition”, dove si avrà effettivamente accesso alla risorsa necessaria. Le partizioni genitore mantengono in esecuzione il “*VSP*” (acronimo di “*Virtualization Service Provider*”) che si connette al “*VMBus*” e gestisce le richieste di accesso ai dispositivi provenienti dalle partizioni figlie. Ogni dispositivo virtuale di queste ultime esegue internamente il “*VSC*” (acronimo di “*Virtualization Service Consumer*”) che redireziona le richieste ai relativi “*VSP*” delle partizioni genitore tramite il “*VMBus*”. Ovviamente tutto questo processo è trasparente al sistema operativo *guest*. [38]

Un’ultima funzionalità degna di nota è il cosiddetto “*Enlightened I/O*” per storage, networking e sottosistema grafico in particolare. È un’implementazione di alcuni protocolli di alto livello sviluppati con la consapevolezza di utilizzare un sistema virtualizzato. Il vantaggio sta nel fatto che sfruttano il

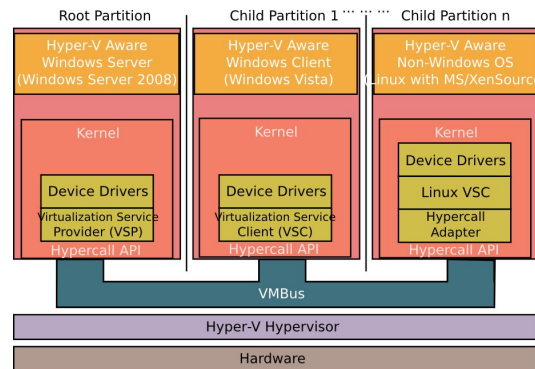


Figura 2.10: Struttura di Microsoft Hyper-V

“VMBus” direttamente, bypassando i livelli di emulazione dei dispositivi in oggetto. Ciò rende la comunicazione più efficiente, ma ha il difetto di rendere necessario il supporto per “Enlightened I/O” sul sistema operativo *guest*.

2.4 Tabelle riassuntive

Informazioni generali

Nome	Produttore	Host CPU	Guest CPU
VMware Player 2.0	VMware	Intel_x86, AMD64	Intel_x86, AMD64
VMware Server	VMware	Intel_x86, AMD64	Intel_x86, AMD64
VMware Workstation 6.0	VMware	Intel_x86, AMD64	Intel_x86, AMD64
VMware ESX Server 3	VMware	Intel_x86, AMD64	Intel_x86, AMD64
Virtual Server 2005 R2	Microsoft	Intel_x86, AMD64	Intel_x86
Virtual PC 2007	Microsoft	Intel_x86, x64	Intel_x86
Oracle VM	Oracle Corporation	Intel_x86, Intel_x86-64, Intel_VT-x	Intel_x86, Intel_x86-64, Intel_VT-x
Sun xVM	Sun Microsystems	x86-64, SPARC	stesso dell'host
Hyper-V	Microsoft	x86-64	?

Nome	Host OS	Guest OS	Licenza
VMware Player 2.0	Windows, GNU/Linux	DOS, Windows, GNU/Linux, FreeBSD, Netware, Solaris	Proprietaria (free)

Nome	Host OS	Guest OS	Licenza
VMware Server	Windows, GNU/Linux	DOS, Windows, GNU/Linux, FreeBSD, Netware, Solaris	Proprietaria (free)
VMware Workstation 6.0	Windows, GNU/Linux	DOS, Windows, GNU/Linux, FreeBSD, Netware, Solaris	Proprietaria
VMware ESX Server 3	nessuno (in- stallazione su "bare metal")	Windows, Red Hat, SuSE, Netware, Solaris	Proprietaria
Virtual Server 2005 R2	Win 2003, Win XP	Win NT, Win 2000, Win 2003, Red Hat, SUSE	Proprietaria (free)
Virtual PC 2007	Win Vista, Win XP Pro, Win XP Tablet PC	DOS, Windows, OS/2	Proprietaria (free)
Oracle VM	Oracle Enter- prise Linux 4/5, RHEL3, RHEL4, RHEL5, Win 2003 ¹ , Win 2003 server ¹ , Win XP ¹	Windows, Oracle Enter- prise Linux, Red Hat Enterprise Linux	Free, Proprie- taria
Sun xVM	Solaris	Win XP, Win 2003 server (solo x86-64), Linux, Solaris	GPLv3

Nome	Host OS	Guest OS	Licenza
Hyper-V	Windows Server 2008	Windows	Proprietaria

Caratteristiche tecniche

Nome	Guest SMP	Metodo adottato ²	Migrazione live
VMware Player 2.0	No	V	/
VMware Server	Si	V	/
VMware Workstation 6.0	Si	V e P	/
VMware ESX Server 3	Si (fino a 4 vie)	V	Si
Virtual Server 2005 R2	No	V	/
Virtual PC 2007	No	V	/
Oracle VM	Si	V e P	Si
Sun xVM	V e P	?	
Hyper-V	?	V	?

Nome	USB	GUI	Supporto commerciale
VMware Player 2.0	Si	Si	Si
VMware Server	Si	Si	Si
VMware Workstation 6.0	Si	Si	Si
VMware ESX Server 3	/	Si	Si
Virtual Server 2005 R2	Si	Si	?
Virtual PC 2007	Si	Si	?
Oracle VM	Si	Si	Si
Sun xVM	Si	Si	Si
Hyper-V	Si	Si	Si

¹Supportato solo su processori con estensioni hardware per la virtualizzazione

²V = Virtualizzazione, P = Paravirtualizzazione (vedere Capitolo 3 Paravirtualizzazione)

Capitolo 3

Paravirtualizzazione

La caratteristica principale di questo approccio è che la macchina virtuale non simula l'hardware, ma offre speciali *API* al sistema *guest* per l'interazione con l'*host* sottostante.

Anche questa tecnica richiede un *VMM*, che rende disponibile al sistema operativo ospitato un'interfaccia virtuale per le periferiche che i S.O. *guest* devono saper richiamare per aver accesso alle risorse:

- i kernel dei S.O. *guest* devono quindi essere modificati per aver accesso all'interfaccia del particolare *VMM*;
- la struttura del *VMM* è semplificata perchè non deve più preoccuparsi di individuare e catturare le operazioni privilegiate dei S.O. *guest*.

Le applicazioni dei sistemi virtualizzati rimangono invariate, nonostante sia richiesto un porting dei sistemi operativi esistenti.

Questo tipo di approccio consente di ottenere il minimo decadimento prestazionale rispetto all'esecuzione non virtualizzata, poichè le istruzioni provenienti dalle macchine virtuali vengono eseguite quasi tutte direttamente sul processore, senza l'intervento di un sistema operativo che si ponga tra la macchina virtuale e le risorse fisiche.

Nei processori di ultima generazione sono integrate alcune primitive hardware utili alla virtualizzazione. In particolare le due principali tecnologie sono *Intel Vanderpool* (o *Intel VT, IVT*) e *AMD Pacifica*. Il discorso dell'inserimento nelle CPU di tali primitive è innanzitutto una questione pratica. Un tempo la complessità di programmazione di una CPU era alla portata di una singola persona, ora però l'hardware è talmente complesso da richiedere interi team di programmatori per essere utilizzato al meglio. In quest'ottica è lecito pensare che le future generazioni di sistemi operativi virtualizzeranno le risorse su più livelli al fine di sfruttare l'hardware disponibile nel miglior modo possibile.

3.1 XEN

XEN è un “*virtual machine monitor*” per architetture *IA-32*, *x86-64*, *IA-64* e *PowerPC*. È scritto completamente come software libero e gira su sistemi operativi *GNU/Linux* o *NetBSD* opportunamente modificati a tale scopo. Una vasta gamma di sistemi unix-like modificati sono supportati come *guest*; inoltre, su alcuni tipi di hardware, a partire dalla versione 3.0 di *XEN*, possono essere virtualizzate anche versioni non modificate di *Microsoft Windows* ed altri sistemi operativi proprietari. [16]

XEN è nato come un progetto di ricerca all’Università di Cambridge istituito da “Ian Pratt”, diventato poi il fondatore di “XenSource Inc.”. Quest’azienda ora supporta lo sviluppo del progetto opensource e vende versioni enterprise del software di virtualizzazione. La prima release pubblica di *XEN* è stata distribuita nel 2003. Recentemente, nell’ottobre 2007, “XenSource Inc.” è stata acquisita da “Citrix Systems” e tutti i prodotti sono stati rinominati sotto il brand “Citrix”: “*XenExpress*” è diventato “*XenServer Express Edition*” e “*XenServer OEM Edition*” (la versione con l’*hypervisor* embedded), “*XenServer*” è diventato “*XenServer Standard Edition*” e “*XenEnterprise*” ha preso il nome “*XenServer Enterprise Edition*”. [17]

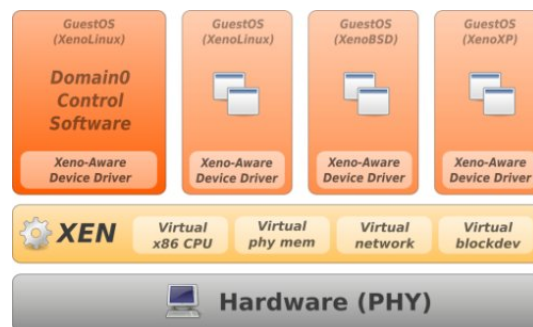


Figura 3.1: Struttura di XEN

Sulla maggior parte delle CPU, *XEN* utilizza il paradigma della paravirtualizzazione, il che significa che il sistema *guest* dev’essere modificato per usare un’interfaccia virtuale tramite chiamate ad opportune *API*. Grazie alla paravirtualizzazione, *XEN* può ottenere alte prestazioni, paragonabili a quelle native aventi la macchina non virtualizzata.

“Intel” ha contribuito a modificare *XEN* per supportare la propria estensione architetturale di virtualizzazione *VT-x* (chiamata *Vanderpool*). Similmente, anche *AMD* ha contribuito al supporto per le estensioni *AMD-V*. Entrambe le tecnologie, pur essendo sostanzialmente diverse nell’implementazione e

nell' "instruction set", sono gestite in *XEN* da un "abstraction layer" comune, offrendo la **possibilità di far girare sistemi operativi *guest* non modificati**, a partire dalla versione 3.0. Tutto ciò è uno sviluppo molto significativo, poichè è possibile virtualizzare sistemi operativi proprietari senza che il loro kernel richieda alcuna modifica se vengono fatti girare su hardware *VT-x* o *AMD-V*.

La virtualizzazione "hardware assisted" offre nuove istruzioni per supportare chiamate dirette da *guest* o driver paravirtualizzati nell'*hypervisor*, tipicamente utilizzate per I/O o altre chiamate di sistema. Offre inoltre modalità di esecuzione addizionali: "root mode" e "non-root mode". Entrambe le modalità sono composte dagli strati di sicurezza da 0 a 3; il sistema *host XEN* opera in "root mode" ed ha accesso all'hardware reale, mentre i *guest* non modificati operano negli strati da 0 a 3 della modalità "non-root" e tutti i loro accessi all'hardware avvengono sotto il completo controllo dell'*hypervisor*.

Attualmente, con la versione 3.0.2 di *XEN*, la lista dei sistemi *guest* non modificati supportati è limitata ad alcune versioni di *Microsoft Windows* e *GNU/Linux*.

Un'altra importante funzionalità offerta da *XEN* è la possibile **migrazione a caldo** delle VM tra host fisici attraverso una *LAN* con downtime impercettibile e senza la necessità di riavvio della macchina virtuale. Durante questa procedura, la memoria della macchina virtuale è copiata nella destinazione senza bloccarne l'esecuzione. È necessario un blocco di tutte le attività per circa 60–300ms (per avere un ordine di grandezza, 60ms è il tempo impiegato per eseguire un "ping ICMP type 8" tra due host con connettività ADSL, quindi al massimo viene perso un tempo pari a 5–6 ping, del tutto trascurabile anche in un ambiente di produzione comune, non "real-time") con lo scopo di effettuare la sincronizzazione finale, per poi riprendere la normale attività dall'host destinazione. [18]

Un sistema operativo che fa girare *XEN* come *host* è anche detto "*domain number 0*" (o *dom0*), mentre un sistema che gira come VM è conosciuto sotto il nome di "*unprivileged domain*" (o *domU*). *XEN* generalmente effettua il boot e carica una versione modificata di *GNU/Linux* o *NetBSD* nel *dom0*. Attualmente la maggior parte delle distribuzioni linux include package di *XEN* precompilati. *XEN* gira su architetture *x86* (con processori *Pentium II* o più recenti), *x86-64*, *IA-64* e *PowerPC*, supportando macchine multiprocessore fino a 64 vie. Inoltre è sotto sviluppo il porting per l'architettura *SPARC*. Le distribuzioni *GNU/Linux* e di altri sistemi operativi liberi che **distribuiscono package precompilati** sono:

- *Debian*;
- *Ubuntu*;

- *Gentoo*;
- *Red Hat*;
- *Fedora*;
- *SUSE*;
- *openSUSE*;
- *NetBSD*;
- *OpenBSD*;
- *ArchLinux*;
- *OpenSolaris* (seppur con qualche restrizione).

I sistemi operativi che possono **girare come *guest* per *XEN*** (a fronte di opportune modifiche) sono:

- *GNU/Linux* (paravirtualizzazione direttamente integrata nel kernel a partire dalla versione 2.6.23, patch altrimenti);
- *Minix*;
- *Bell Labs Plan 9*;
- *NetBSD* (la versione 2.0 ha supporto per *XEN* 1.2, la 3.0 ha supporto per *XEN* 2.0 e la 3.1 supporta *XEN* 3.0);
- *OpenBSD*;
- *FreeBSD* (supporto limitato);
- *OpenSolaris*;
- *Novell NetWare*;
- *GNU Hurd*;
- *OZONE* (supporto per *XEN* 1.2).

Attualmente è possibile **far girare *Microsoft Windows* come sistema *guest*** senza doverlo adattare, utilizzando le estensioni di virtualizzazione hardware fornite dalla tecnologia *Intel Vanderpool* (o *Intel VT, IVT*) o dalla *AMD Pacifica*. Questo supporto è stato introdotto da *XEN 3.0*. Durante lo sviluppo di *XEN 1.x*, il gruppo *Microsoft Research*, in collaborazione con l'*Operating System group* dell'Università di Cambridge, ha prodotto una versione alpha del port di *Windows XP* per *XEN*. Ciò è stato possibile grazie al "*Microsoft's Academic Licensing Program*", ma i termini della suddetta licenza non permettono la pubblicazione del port.

Sono stati sviluppati inoltre una serie di tool di terze parti (chiamati "*XEN Management Consoles*") per facilitare i compiti comuni di amministrazione di un host *XEN* come configurare, accendere/spegnere e monitorare lo stato delle varie VM. Alcuni tra i più comuni sono:

- "*Enomalism dashboard*", rilasciato con licenza LGPL e basato su Python;
- "*XEN Tools*";
- "*HyperVM*", web application;
- "*ConVirt*", applicazione standalone con GUI;
- "*virt-manager*", sviluppato da *Red Hat*.

3.2 User-mode Linux

User-mode Linux (o abbreviato, *UML*) è un progetto opensource sviluppato a partire dal febbraio 1999 da "Jeff Dike", unico mantainer, ed è in continuo sviluppo ed aggiornamento grazie all'attivo "UML Team". La roadmap dello sviluppo di *UML* è stata la seguente:

- fine 1998, inizi 1999: iniziato a pensare se *UML* fosse possibile;
- febbraio 1999: iniziato a sviluppare *UML*;
- 3 giugno 1999: annunciato *UML* sulla mailing list del kernel *GNU/Linux*;
- 12 settembre 2002: *UML* introdotto nativamente nel kernel 2.5.34.

Prima di poter parlare di *UML* è necessario introdurre la suddivisione dei metodi operativi utilizzata sui moderni sistemi, che prevede due distinte modalità di funzionamento:

- “*kernel mode*”: modalità all’interno della quale la macchina opera con risorse critiche come l’hardware, ad esempio effettua l’accesso diretto alla memoria, IRQ, DMA, ecc.;
- “*user mode*”: modalità al di sopra di “*kernel mode*” in cui gli utenti del sistema utilizzano le loro applicazioni senza preoccuparsi di possibili e repentini blocchi di sistema.

Tale suddivisione permette al “*kernel mode*”, che opera ad un livello più basso nei sistemi, di controllare le applicazioni che girano in “*user mode*” ed eventualmente bloccarle prima che possano arrecare danni o blocchi all’intero sistema.

UML è un kernel *GNU/Linux* (chiamato *guest kernel*) patchato per essere compilato ed eseguito completamente in “*user space*” sulla macchina *host* (su cui gira *GNU/Linux*), lavorando “sopra” il kernel residente (*host kernel*) e risultando quindi equivalente ad un normale processo utente a livello di privilegi ed accesso alle risorse. [23]

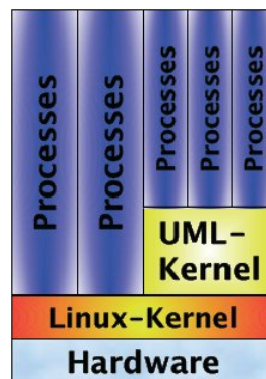


Figura 3.2: Struttura di User-mode Linux

In *UML*, a differenza di altri sistemi di virtualizzazione che emulano la macchina fisica, vengono intercettate le “*system-call*” per offrire la necessaria piattaforma hardware su cui girare (come mostrato in figura 3.3 a pagina 35).

Poichè *UML* rende possibile il funzionamento dello stesso binario del *kernel host* sulle macchine *guest* e tale binario esegue le “*system-call*” nello stesso modo dell’*host*, deve essere in grado di:

- intercettare le *system call*;
- prevenire la loro esecuzione nel *kernel host* ed eseguirle invece nel loro ambiente.

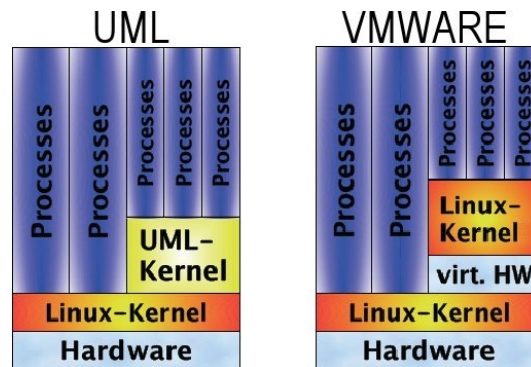


Figura 3.3: Differenze strutturali tra UML e VMware

Tutto ciò è realizzato tramite “*ptrace*”, meccanismo di tracing delle “system-call” che permette ad un processo di intercettare e modificare le chiamate di sistema di un altro. Il “tracing thread” annulla le chiamate di sistema reindirizzandole nel kernel *UML*, che è accessibile direttamente dal processo *UML*. Questo tipo di funzionamento, chiamato “modo TT” ha dei problemi di sicurezza e velocità. Il kernel *UML* è presente nello spazio di indirizzamento di ognuno dei suoi processi, ognuno dei quali, tecnicamente, ha permessi in scrittura (almeno nell’installazione standard). [19]

Esiste una modalità più sicura, detta *UML jail*, che rende i dati del kernel solo leggibili durante l’esecuzione di un processo, ma tale modalità è anche estremamente più lenta. Per ovviare a tale limitazione, l’applicazione della patch “SKAS” (acronimo di “Separate Kernel Address Space”) al kernel della macchina *host* fa sì che il kernel *UML* venga eseguito in uno spazio di indirizzamento completamente differente da quello dei suoi processi, che quindi non possono più accedervi. Lo spazio di indirizzamento dei processi diviene quindi identico a quello che avrebbero girando sulla macchina *host* apportando inoltre miglioramenti notevoli sul piano della velocità.

Le **caratteristiche principali di *UML*** sono:

- possibilità di far girare più macchine virtuali allo stesso tempo sullo stesso host;
- ognuna delle macchine virtuali può avere: una console (tipicamente “xterm” in pop-up), una memoria (un singolo file sull’*host*), un suo *filesystem* (un singolo file sull’*host*), una o più interfacce di rete;
- ogni interfaccia di rete virtuale può essere connessa ad un dominio di collisione (indifferentemente fisico o virtuale);

- ciascun dominio di collisione può essere connesso a più interfacce.
- i processi e gli utenti, attivi sotto *UML*, non hanno accesso alle risorse dell'*host* che non siano state esplicitamente fornite alla macchina virtuale;
- è possibile assegnare alla macchina virtuale solo l'accesso all'hardware che si vuole concedere, quindi scegliendo adeguatamente le risorse condivise, tutto ciò che avviene sulla VM non influenzerà in alcun modo l'integrità della macchina reale;
- possiede un filesystem virtuale, chiamato *hostfs* che fornisce accesso diretto al *filesystem* della macchina *host* permettendo alla macchina virtuale di accedere ai file contenuti sul sistema ospitante in modalità sia di sola lettura che in scrittura (questa opzione è disponibile semplicemente utilizzando il comando "*mount -t hostfs*").

Oltre agli utilizzi tipici dei sistemi virtualizzati, gli **impieghi maggiormente sfruttati dagli utilizzatori di *UML*** sono:

- *virtual hosting*: ogni cliente avrà la percezione di utilizzare un server dedicato accedendovi come "*root*";
- *educazione*: ogni studente o corsista potrà utilizzare la macchina virtuale con privilegi amministrativi, con pregi e difetti che ciò comporta (oltre all'emozione di essere *root* :-));
- *sviluppo, test e debugging*: utilizzo della macchina virtuale come box di testing del kernel o di particolari configurazioni dei servizi;
- *virtual networking*: studio dei protocolli esistenti, emulazione di network complessi, clustering. Inoltre è possibile emulare nodi *TCP/IP* standard, switch e hub, bridge, router, firewall. [22]

3.3 Tabelle riassuntive

Informazioni generali

Nome	Produttore	Host CPU	Guest CPU
XEN	Università di Cambridge, Intel, AMD	Intel x86, AMD64 (WiP ¹ su ports IA-64 e PowerPC)	stessa dell'host
User-mode Linux	Jeff Dike, UML Team	Intel x86, PowerPC	stessa dell'host

Nome	Host OS	Guest OS	Licenza
XEN	NetBSD, GNU/Linux, Solaris	GNU/Linux, NetBSD, FreeBSD, OpenBSD, Solaris, Plan 9, Win XP ² , Win 2003 ²	GPL
User-mode Linux	GNU/Linux	GNU/Linux	GPL (v2)

Caratteristiche tecniche

Nome	Guest SMP	Metodo adottato ³	Migrazione live
XEN	Si	P	Si
User-mode Linux	No	P	No

Nome	USB	GUI	Supporto commerciale
XEN	Si	Si	Si
User-mode Linux	Si	No	No

Come è facile notare dalle tabelle proposte, *XEN* risulta essere un prodotto molto più maturo rispetto a *User-mode Linux*, nonostante la sua più recente nascita.

¹WiP = Work in Progress

²Necessità di XEN 3.0 e CPU con estensioni "*Intel Vanderpool*" o "*AMD Pacifica*"

³P = Paravirtualizzazione

Capitolo 4

Storage per macchine virtuali

In un sistema virtualizzato non solo a livello “workstation”, ma a livello server (come ad esempio *VMware Infrastructure* o *Xen*, descritti nei capitoli precedenti), la componente fondamentale dopo i server fisici che permettono alle VM di girare è lo *storage*.

Con il termine *storage* si indicano i dispositivi hardware per l’immagazzinamento in modo non volatile di dati elettronici.

In particolare, nell’ambito della virtualizzazione e delle macchine virtuali, si parla sempre più spesso di *network storage*, ovvero qualunque tipo di *storage* che coinvolge l’accesso alle informazioni tramite una rete di computer. Ciò dà la possibilità a multipli nodi di condividere lo storage nello stesso tempo. Condividendo lo *storage*, la gestione delle informazioni è centralizzata, riducendone la duplicazione ed evitando quindi ridondanze inutili e spesso dannose.

4.1 Storage unico condiviso

È un modello di *network storage* dove tutti i dati sono mantenuti in uno stesso posto. Tale nodo del sistema deve quindi essere affidabile, scalabile, veloce e sicuro. Sarà quindi composto da array *RAID*, dispositivi di backup ed interfacce veloci e ridondanti.

I due sottomodelli più utilizzati in sistemi a storage unico condividono sono sicuramente:

- *SAN (Storage Area Network)*: sistema estremamente veloce, ma che sfrutta hardware e tecnologie costose;
- *NAS (Network Attached Storage)*: sistema più lento, ma più economico e semplice da mettere in opera, gestire ed integrare; è basato su standard universalmente riconosciuti e tecnologie/hardware non costosi.

4.1.1 Storage Area Network

Una *Storage Area Network* (*SAN*) è una rete ad alta velocità composta da dispositivi di memorizzazione di massa condivisi. I protocolli attualmente più diffusi per l'utilizzo degli storage via rete sono *FC* (Fibre Channel) ed *iSCSI* (Internet SCSI).

Esiste una definizione formale di rete *SAN* sul dizionario tecnico pubblicato dalla “*SNIA*” (Storage Networking Industry Association) che la definisce nei seguenti termini:

“Una rete il cui scopo principale è il trasferimento di dati tra sistemi di computer ed elementi di storage e tra elementi di storage. Una rete SAN consiste in un’infrastruttura di comunicazione, che fornisce connessioni fisiche e in un livello di gestione che organizza connessioni, elementi di storage e sistemi di computer in modo da garantire un trasferimento di dati sicuro e robusto.”

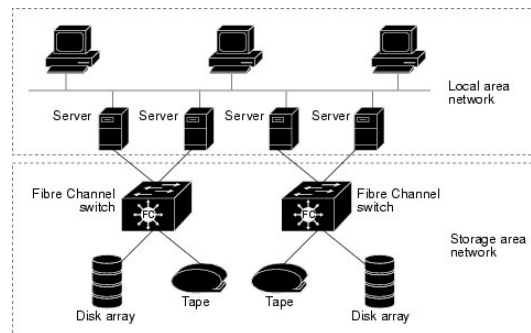


Figura 4.1: Struttura di una SAN

Un’architettura SAN lavora in modo che tutti i dispositivi di memorizzazione siano disponibili a qualsiasi server della rete LAN o MAN di cui la SAN fa parte; una SAN può essere anche condivisa fra più reti interconnesse, anche di natura diversa: in tal caso uno dei server locali agisce come “bridge” fra i dati memorizzati e gli utenti finali. Il vantaggio di un architettura di questo tipo è che tutta la potenza di calcolo dei server è utilizzata per le applicazioni, in quanto i dati non risiedono direttamente su alcuno di tali server.

Vantaggi

Le aziende devono poter accedere ai dati in modo rapido e sicuro e quindi la filosofia dell’architettura SAN è quella di poter integrare tutte le caratteristiche dei tradizionali sistemi di memorizzazione:

- alte prestazioni
- alta disponibilità
- scalabilità
- facilità di gestione

Tutto ciò integrato con le caratteristiche di connettività ed accesso distribuito del network computing, attraverso un'architettura di rete dedicata alla gestione ed archiviazione dei dati, in grado di non sovraccaricare i server nelle operazioni di scrittura e lettura degli stessi, da e verso lo storage. Le reti SAN forniscono una serie di indubbi vantaggi rispetto ai dispositivi di storage connessi direttamente ai server (detti "*Direct Attached Storage*"). Offrono una connettività any-to-any tra server e dispositivi di storage, aprendo in tal modo la strada al trasferimento diretto di dati tra periferiche di memorizzazione, con conseguenti miglioramenti dell'efficienza nello spostamento di dati e processi, quali il backup o la replica dei dati.

Applicazioni

L'impiego delle attuali tecnologie di networking proposte per le reti SAN consente di:

- raggiungere distanza di connettività superiori e prestazioni migliori rispetto a quanto non sia possibile con l'attuale tecnologia *SCSI*;
- facilitare il compito di centralizzare la gestione dello storage;
- consolidare lo storage ed il clustering dei sistemi presenti;
- condividere i dati tra piattaforme eterogenee.

Garantendo alte prestazioni ed un accesso diretto ai dischi, le SAN faciliteranno lo spiegamento di database centralizzati e di applicazioni enterprise, facilitandone il backup ed eventuali disaster recovery. [24]

4.1.2 Network Attached Storage

Un *Network Attached Storage (NAS)* è un dispositivo collegato ad una rete di computer la cui funzione è quella di condividere tra gli utilizzatori della rete un'area di storage. I *NAS*, generalmente, sono dei semplici computer con a bordo il minimo necessario per poter utilizzare le funzionalità di networking.

Attualmente i più diffusi sono macchine con sistema operativo *GNU/Linux* (trasparente all'utente finale) contenenti numerosi hard disk per l'immagazzinamento dati. Tale architettura ha il vantaggio di rendere disponibili i file contemporaneamente su piattaforme eterogenee, in quanto il sistema operativo implementa i server di rete per tutti i protocolli più diffusi (*FTP*, *NFS*, *SMB*, ecc ecc.).

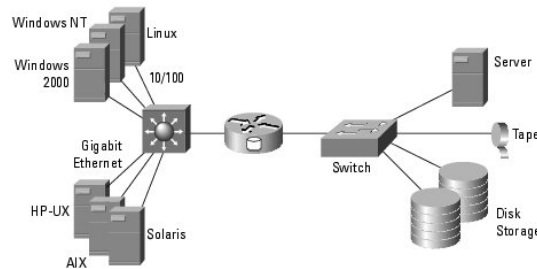


Figura 4.2: Struttura di sistema con NAS

Questi apparati non vanno confusi con le *SAN*, soluzioni di storage ben differenti: come visto nella sezione precedente, tali sistemi comprendono una rete e sono assoggettati a protocolli spesso proprietari e tecnologie molto più costose. Piuttosto, un sistema *NAS* può essere utilizzato come nodo di una *SAN*, data la scalabilità di tale architettura.

Vantaggi e svantaggi

Il vantaggio dei *NAS*, oltre a centralizzare l'immagazzinamento dei dati in un solo posto invece che spargerli su diversi sistemi di una rete, è che sono unità altamente specializzate dal punto di vista prestazionale e della sicurezza dei dati: gran parte di tali sistemi può implementare array RAID al fine di garantire la ridondanza dei dati e permettendo inoltre l'aggiunta e la rimozione di dischi rigidi in runtime senza bloccare il funzionamento dell'intero sistema ("hot swap").

Nell'ambito di questa architettura, lo svantaggio maggiore è costituito principalmente dall'enorme quantità di dati che transita sulla rete e dai limiti prestazionali e di stabilità dei *filesystem di rete* disponibili sul mercato. [25]

4.2 Storage distribuito

Un *filesystem distribuito* (o più semplicemente *DFS*) è una tipologia di "network filesystem" che permette la memorizzazione di file e risorse in dispositivi

di archiviazione distribuiti in una rete informatica. A differenza di un comune file system locale, i dati non vengono letti o archiviati su di un dispositivo locale, ma, attraverso un meccanismo client-server, su dispositivi remoti, collegati in maniera trasparente alla propria gerarchia di file. Ogni nodo del sistema ha accesso diretto solo ad una parte dell'intero filesystem, a differenza delle soluzioni "storage unico condiviso" in cui tutti i nodi hanno accesso diretto all'intero sistema di storage. [26]

Generalmente i *DFS* includono strutture per garantire la replicazione trasparente dei dati e meccanismi di tolleranza ai guasti. Il che significa che se un numero limitato di nodi va offline, il sistema continua a lavorare senza perdita di dati.

Un "*distributed data store*" è una rete in cui l'utente archivia i propri dati su un insieme di nodi della rete. Normalmente l'utente stesso rende disponibile il suo computer per archiviare i dati di altri utenti agendo da nodo attivo a sua volta.

Il tipico esempio di una rete di questo tipo è la rete "BitTorrent", in cui il nodo origine può andare offline senza interrompere la distribuzione dei dati condivisi. Questo però è il caso semplice in cui i file sono un numero limitato. In una rete come "Freenet" tutti i nodi rendono disponibili tutti i file. [27]

4.3 Alternative disponibili

Fibre Channel

È una tecnologia per reti dati, usata principalmente per implementazioni in *Storage Area Network*. *Fibre Channel* è standardizzata nel T11 del "Comitato Internazionale per gli Standard dell'Information Technology" (o "INCITS"), un comitato degli standard accreditato dall'*American National Standards Institute* (*ANSI*).

Si sviluppò per un utilizzo principalmente nel campo dei supercomputer, ma è diventato il tipo di connessione standard per le *Storage Area Network* nell'enterprise storage. Nonostante la connotazione comune del suo nome, il segnale *Fibre Channel* può andare sia su cavi di rame *UTP* che su cavi a *fibra ottica*. Il *Fibre Channel Protocol* (o *FCP*) è il protocollo di interfaccia dello *SCSI* sul *FC*.

Fibre Channel è nato nel 1988, approvato come standard dall'*ANSI* nel 1994, per semplificare il sistema *HIPPI* allora in uso per ruoli simili. *HIPPI* usava delle coppie da 50 cavi e connettori di enormi dimensioni, avendo inoltre una limitata lunghezza di cablaggio.

FC primariamente è stato concepito per semplificare il tipo di connessione ed incrementare la distanza di cablaggio, invece che per la pura velocità di

trasmissione. Successivamente è stato focalizzato sull'indirizzamento dei dischi *SCSI*, provvedendo anche all'alta velocità e ad un crescente numero di dispositivi da poter connettere. È stato inoltre aggiunto il supporto ad un alto numero di protocolli di alto livello, inclusi SCSI, ATM ed IP. [28]

iSCSI

La versione *iSCSI* (che sta per *internet SCSI*) è un'evoluzione dello *SCSI-3* che mantiene praticamente invariata l'impostazione dello SCSI di base, specialmente per quanto riguarda il set di comandi. Essa si basa sull'idea di incorporare nello standard SCSI il protocollo TCP/IP, realizzando un protocollo di storage che possa viaggiare su Ethernet. I sostenitori dello standard iSCSI ritengono che questa tecnologia possa rimpiazzare, nel lungo termine, le tecnologie concorrenti, compresa la tecnologia *Fibre Channel*, poichè le velocità di trasmissione dati ottenibili con reti Ethernet stanno attualmente aumentando più rapidamente di quelle ottenibili con le altre tecnologie. In prospettiva l'iSCSI ha dunque i requisiti necessari per affermarsi sia nel mercato di fascia bassa che di fascia alta, utilizzando la medesima tecnologia. [29]

iSCSI semplicemente dà la possibilità a due host di negoziare ed infine scambiare comandi SCSI utilizzando reti IP. Tale protocollo riutilizza il popolare bus ad alte performance per storage locale rendendolo disponibile su una rete TCP/IP e realizzando una *SAN*. A differenza di altri protocolli per *SAN*, iSCSI non necessita di un cablaggio dedicato, può essere utilizzato sull'infrastruttura presente. "iSCSI" diventa quindi un'alternativa low-cost al "Fibre Channel", che richiede infrastrutture dedicate. [30]

ATA over Ethernet

AoE (acronimo di *ATA over Ethernet*) è un protocollo di rete sviluppato dalla "Brantley Coile Company", progettato per accedere a dispositivi di storage con interfaccia *ATA* tramite reti *Ethernet*. Dà la possibilità di costruire *SAN* con tecnologie standard low-cost. *AoE* non sfrutta i protocolli di rete al di sopra di Ethernet (come ad esempio TCP, IP, UDP, ecc.). Ciò significa che *AoE* non sottostà alle classiche regole di routing presenti sulla rete.

I sistemi operativi che forniscono supporto nativo per *AoE* sono: *GNU/Linux* dal kernel 2.6.11, *SUN Solaris* dalla versione 1.4, *Bell Labs Plan 9* e *OpenBSD* dalla versione 4.3. Il protocollo *AoE* semplicemente inserisce comandi *ATA* in frame Ethernet, quindi tale rete rimpiazza virtualmente il cavo che arriva al disco fisico.

Sfruttando il protocollo *AoE* non è consigliabile utilizzare un filesystem tradizionale, poichè si potrebbe incappare in corruzioni del filesystem stesso o in kernel panic del sistema operativo. Questo perchè i filesystem tradizionali

sono stati progettati seguendo l'idea che il FS sia l'unico utente del disco rigido del sistema su cui gira. I "cluster filesystem negano l'assunzione che solo un singolo computer ha accesso allo storage. Sono progettati per rendere disponibili le risorse ad utenti multipli di un cluster, allocandole in maniera sicura e coordinandone le azioni. [31]

Network File System

Il termine network file system viene spesso erroneamente utilizzato come termine generico per indicare un filesystem in grado di gestire dispositivi di memorizzazione remoti.

Network File System, o più comunemente *NFS*, è un protocollo sviluppato inizialmente da "Sun Microsystems" nel 1984 e definito dagli *RFC* 1094, 1813, (3010) e 3530. L'*NFS* è un filesystem che consente ad un sistema di utilizzare la rete per accedere a dischi remoti come fossero dischi locali. La versione 2 del protocollo originale utilizzava unicamente UDP e prevedeva che i server non dovessero conservare memoria degli accessi degli utenti. In passato quindi eventuali meccanismi di blocco delle risorse (lock) dovevano essere implementati esternamente al protocollo. La versione 3 introdusse il supporto per il TCP per il trasporto delle informazioni. L'utilizzo del TCP consente di utilizzare l'*NFS* attraverso una rete WAN, sebbene non sia nè molto semplice nè efficiente. La versione 4 venne influenzata dall'*AFS* ed includeva miglioramenti nelle prestazioni, aggiungeva un supporto migliorato alla sicurezza ed introduceva un protocollo che teneva conto dello stato dei client.

NFS è spesso associato a sistemi *Unix*, sebbene spesso sia utilizzato anche da macchine utilizzanti i sistemi operativi *Macintosh* e *Microsoft Windows*. Il "server message block" (o *SMB*) è un'implementazione simile di un filesystem di rete per computer *Microsoft Windows*.

Il *NFS* permette ad *N* calcolatori che compongono un sistema distribuito di condividere file, directory o un intero filesystem utilizzando il protocollo client-server. Un client deve richiedere esplicitamente ad un server del sistema distribuito di condividere una directory o un file, dichiarando un punto di montaggio. Una volta effettuato un montaggio, un utente sul client accede alla directory montata in modo assolutamente trasparente, ossia accede alla directory remota credendo di accedere ad una directory locale. [32]

CIFS/SMB

Server Message Block, o *SMB*, è un protocollo usato principalmente per condividere file, stampanti, porte seriali e comunicazioni di varia natura tra diversi nodi di una rete. Esso include anche un meccanismo di comunicazione tra processi autenticati. È utilizzato soprattutto dai sistemi *Microsoft Windows*.

SMB è stato inventato da Barry Feigenbaum presso la *IBM*, ma la versione più largamente usata è stata pesantemente modificata da *Microsoft*. Originariamente, il protocollo venne studiato per operare al di sopra di *NetBIOS*, sebbene possa agire anche direttamente su TCP/IP a partire dall'implementazione in *Windows 2000*.

Nel momento in cui Sun Microsystems annunciò il *WebNFS*, *Microsoft* lanciò l'iniziativa, nel 1998, di chiamare SMB “*Common Internet File System*” (in breve, *CIFS*) ed incluse diverse migliorie, compreso il supporto ai link simbolici, la gestione di files di grosse dimensioni e la possibilità di operare senza alcun supporto di *NetBIOS*. [33]

A causa dell'importanza del protocollo SMB, necessario all'interoperabilità con la piattaforma *Microsoft Windows*, nacque il progetto *Samba*, che rappresenta una implementazione free usata per garantire compatibilità SMB con sistemi operativi non *Microsoft*.

L'implementazione:

- approccio client/server: SMB agisce con un approccio client/server, per cui un client avanza le sue richieste ed il server risponde opportunamente. Una sezione del protocollo è dedicata specificatamente all'accesso al filesystem, in modo che i client possano fare richieste al file server, ma ci sono altre sezioni specializzate per le comunicazioni tra processi. SMB è stato ottimizzato per l'uso in rete locale, ma può essere usato anche attraverso Internet. I server SMB rendono il loro filesystem e le altre risorse disponibili ai client sulla rete. I client possono avere i loro propri dischi che non sono pubblicamente accessibili ed accedere comunque al filesystem condiviso ed alle stampanti del server. Quest'ultima è la modalità primaria di utilizzo del protocollo;
- prestazioni: spesso viene considerato SMB un protocollo che richiede un pesante utilizzo di banda in quanto ogni client notifica la sua presenza con un broadcast all'intera rete. In realtà, SMB non usa broadcast, esso viene usato da *NetBIOS* per localizzare i server disponibili ad intervalli periodici. Ciò è accettabile in reti con meno di 20 host, ma il traffico dovuto ai pacchetti broadcast può creare problemi con l'aumentare del numero di macchine connesse. Tale problema può essere mitigato sfruttando *WINS*, un protocollo di localizzazione dei servizi: usa un sistema di registrazione e centralizzazione delle richieste di servizi più avanzato, ma implica una crescita nella complessità di manutenzione della rete;
- modifiche di *Microsoft*: con l'obiettivo di semplificare l'interfaccia di utilizzo del protocollo, *Microsoft* ha aggiunto diverse caratteristiche alla

sua implementazione di SMB rispetto alla concezione originaria standard; ad esempio, la seconda versione di NTLM venne sviluppata perchè la prima (derivata dalla originaria specifica di SMB) utilizzava la cifratura “*Data Encryption Standard*” (DES) in maniera troppo complessa;

- punti di interesse: il meccanismo di comunicazione tra processi incluso in SMB merita una menzione speciale. Grazie ad esso sono fornite le “named pipes”, con cui è implementato il meccanismo di autenticazione e l’implementazione Microsoft del DCE/RPC (conosciuta come MSR-PC). SMB è anche usato come fondamento del *Microsoft Distributed File System*.

Andrew File System

Il “file system Andrew”, o *AFS* è un filesystem distribuito sviluppato dalla Carnegie Mellon University, all’interno del progetto Andrew. Il nome è stato dato in onore di Andrew Carnegie e Andrew Mellow. L’utilizzo principale di tale filesystem è nell’elaborazione distribuita.

AFS ha diversi vantaggi rispetto ai tradizionali filesystem distribuiti, in particolare riguardo alla sicurezza ed alla scalabilità. Non è raro in ambito commerciale che questo file system supporti oltre 50000 client contemporanei. AFS utilizza *Kerberos* per eseguire le autenticazioni ed implementa liste per il controllo degli accessi (ACL) alle singole directory per utenti e gruppi. Il caching a livello del client permette un miglioramento delle prestazioni e limita l’accesso al filesystem in caso di un crash del server o di un sovraccarico della rete. Nell’AFS i file vengono mantenuti nella cache su richiesta della singola workstation. Le operazioni di lettura e scrittura vengono eseguite direttamente sulla copia mantenuta nella cache locale. Quando un file che ha subito delle modifiche viene chiuso, la parte modificata viene copiata sul file presente sul server. La consistenza della cache viene mantenuta da un meccanismo denominato “callback”. Quando un file si trova nella cache di un client, il server ne prende nota, e in caso tale file venga modificato, si impegna a notificarlo ai client che ne hanno una copia. Tale meccanismo viene a cadere e quindi riavviato ogni qualvolta un server, un client o un errore nella rete generino un timeout. Il ripristino del callback consta nel controllo dello stato dei file nelle cache dei vari client e non richiede la ritrasmissione dei file.

Una caratteristica peculiare dell’AFS è il volume, un’organizzazione gerarchica ad albero dei file e delle directory di ordine inferiore. I volumi sono creati dall’amministratore e collegati a specifici nomi di percorso in una cella AFS. Una volta creato, gli utenti del filesystem possono creare directory e file, senza preoccuparsi della locazione fisica dei dati. Un volume può avere una quota assegnata, in modo da limitarne lo spazio a disposizione. Inoltre, i volumi AFS

possono essere duplicati in copie di sola lettura, con tutti gli evidenti vantaggi che ciò comporta. [34]

Il filesystem Andrew ha profondamente influenzato la versione 4 di NFS. Inoltre, una variante di AFS, il “Distributed File System” (DFS) è stato adottato dalla “*Open Software Foundation* nel 1989 come parte del loro ambiente di elaborazione distribuito.

Le tre maggiori implementazioni sono Transarc (IBM), OpenAFS ed Arla.

ZFS

ZFS è un filesystem opensource sviluppato dalla *Sun Microsystems* per il suo sistema operativo Solaris. È stato progettato da un team con a capo Jeff Bonwick. Il nome originario doveva essere “Zettabyte File System”, ma è diventato un acronimo. ZFS fu annunciato nel settembre del 2004. Il codice sorgente fu rilasciato assieme a quello di Solaris il 31 ottobre del 2005 e rilasciato nella build 27 di *OpenSolaris* il 16 novembre 2005. ZFS fu fornito assieme all’aggiornamento 6/06 di Solaris 10 nel giugno del 2006.

ZFS è noto per la sua alta capacità e per l’integrazione di diversi concetti presi da vari filesystem in un unico prodotto. È un filesystem a 128-bit, potendo fornire uno spazio di 16 miliardi di miliardi (16 quintilioni) di volte la capacità dei file system attuali a 64-bit. I limiti del ZFS sono stati progettati per essere così ampi da non essere mai raggiunti in una qualsiasi operazione pratica. Bonwick affermò che “per riempire un filesystem a 128 bit non sarebbero bastati tutti i dischi della terra”. Un utente che volesse creare mille file al secondo, impiegherebbe 9000 anni a raggiungere il limite.

La particolarità più interessante dopo l’enorme capacità è senza dubbio lo “storage pool”. ZFS è costituito su un insieme di pool di storage virtuali. I pool si basano su uno o più device virtuali (vdevs), ciascuno dei quali può fare riferimento ad un device fisico, ad un mirror (RAID1) di uno o più device, oppure su un gruppo di device RAID. Lo spazio di tutti i vdevs è reso disponibile a tutti i file systems nello zpool. Per limitare lo spazio, è stata istituita la gestione delle quote disco. [35]

Di ZFS è stato detto: “Anche se ci piacerebbe che la legge di Moore possa continuare per sempre, la meccanica quantistica impone alcuni limiti fondamentali sul calcolo computazionale e sulla capacità di memorizzazione di una qualsiasi unità fissa. In particolare è stato dimostrato che un chilo di materia confinata in un litro di spazio può effettuare al massimo 10^{51} operazioni al secondo su al massimo 10^{31} bit di informazioni (vedere Seth Lloyd, *Ultimate physical limits to computation*. Nature 406, 1047-1054 (2000)). Un pool di storage a 128-bit completamente riempito dovrebbe contenere 2^{128} blocchi (nibble) = 2^{137} bytes = 2^{140} bits; quindi lo spazio minimo richiesto dovrebbe

essere $(2^{140} \text{ bit}) / (10^{31} \text{ bits/kg}) = 136$ miliardi di kg. Con il limite dei 10^{31} bit/kg, l'intera massa di un computer dovrebbe essere sotto forma di energia pura. Secondo l'equazione $E = mc^2$, l'energia residua dei 136 miliardi di kg è di $1,2 \times 10^{28}$ J. La massa dell'oceano è circa $1,4 \times 10^{21}$ kg. Occorrerebbero 4000 J per aumentare la temperatura di 1 kg di acqua per 1 grado Celsius e circa 400000 J per bollire 1 kg di acqua ghiacciata. La fase di vaporizzazione richiede altri 2 milioni di J/kg. L'energia richiesta per bollire l'oceano è circa $2,4 \times 10^6$ J/kg * $1,4 \times 10^{21}$ kg = $3,4 \times 10^{27}$ J. Quindi, riempire uno storage a 128-bit dovrebbe richiedere più energia che bollire gli oceani.”

Capitolo 5

VMware Vs. XEN: confronto sul campo

Durante la mia carriera accademica ho svolto attività di tirocinio presso “Itaca Informatica e Sistemi srl” di Udine, una società che offre soluzioni ad ampio raggio nei più comuni settori informatici per PMI. Tra questi c’è l’amministrazione di sistemi server sia in ambiente *Microsoft Windows*, sia *GNU/Linux*. Ciò che riguarda maggiormente questa tesi di laurea è il fatto che gli obiettivi previsti nel progetto di tirocinio messi a disposizione sono stati: “Messa in opera di server installando il S.O. su macchina virtuale o virtualizzando un sistema già in produzione al fine di agevolarne la gestione e la politica di backup.”. Oltre a ciò ho avuto la possibilità di sperimentare l’installazione e l’utilizzo e la gestione dei due maggiori software di virtualizzazione oggi presenti sul mercato, *VMware* e *XEN*.

Al giorno d’oggi, questi nomi rappresentano i due più grossi rivali nel campo di virtualizzazione, in particolare lato server. Sono dell’idea che siano due prodotti leader nel proprio settore, ma che siano destinati a due mercati diversi. L’ambiente proposto da *VMware*, fin dal primo impatto, è molto “enterprise”, ma allo stesso tempo user-friendly rispetto al rivale open source, decisamente più ostico nella sua installazione base (di cui parlerò in seguito). Solo negli ultimi mesi *XEN* si è concentrata sul consolidamento in ambito aziendale con soluzioni che raggiungono l’avversario. Questo perché *VMware* è nato col preciso scopo commerciale e molto tempo prima, mentre *XEN* ha visto la luce come progetto di ricerca “in ritardo” rispetto al concorrente.

Quest’ultimo capitolo vuol fare una panoramica su pregi e difetti presenti in entrambe le soluzioni, scrutandole da una visuale intermedia, cioè da un utente che conosce gli argomenti trattati, ma che non possiede le competenze necessarie per definire una soluzione migliore dell’altra basandosi sulla sola implementazione dell’hypervisor a livello kernel.

I dati sono stati raccolti secondo due criteri: ove fosse possibile utilizzando *VMware Server* e *XEN 3.2*, mentre per quanto riguarda la stima dei costi sono state prese in esame le distribuzioni enterprise di entrambi i prodotti.

5.1 Installazione software virtualizzazione

Nei test che ho svolto, ho utilizzato come sistemi di virtualizzazione *VMware server* (la versione 1.0.2 distribuita direttamente sul sito ufficiale) e *XEN* (la versione 3.0.1 disponibile nella distribuzione *Debian GNU/Linux*, basata sul codice *Xensource* ed integrata nel *kernel 2.6*).

Prendendo in esame l'installazione dei due sistemi su un server host equipaggiato con *Debian GNU/Linux*, la differenza fondamentale sta nel fatto che *XEN* è disponibile già pacchettizzato sia per Debian, sia per ogni altra distribuzione GNU/Linux adottata, mentre *VMware* deve essere scaricato e registrato direttamente sul sito VMware stesso.

La scelta di *GNU/Linux* come sistema host è stata principalmente dettata dal fatto che *XEN* è disponibile solo per sistemi unix-like, escludendo quindi *Microsoft Windows*. Inoltre, al fine di fare un paragone sufficientemente valido, mantenere lo stesso sistema come host è abbastanza logico.

XEN

L'installazione di questo hypervisor è molto semplice se si utilizza, come nel mio caso, un gestore dei pacchetti che risolve automaticamente le dipendenze. Ciò significa che è sufficiente lanciare:

```
root@server:/root# apt-get install xen-linux-system-$(uname -r)-xen-686
```

per installare tutto il necessario per l'utilizzo di *XEN*, dal kernel col supporto per *Dom0* alle svariate utility per creare e gestire le macchine virtuali.

Dopo lo scaricamento e l'installazione dei pacchetti è sufficiente riavviare il sistema utilizzando il nuovo kernel per avere una macchina host su cui andranno a girare i nostri guest virtualizzati grazie alla tecnologia di *XEN*.

VMware

L'installazione di *VMware* è più impegnativa rispetto a quella del rivale open-source, ma non presenta comunque grosse difficoltà per un utente non completamente a digiuno di *GNU/Linux*. Il metodo più semplice per installare questo sistema è senza dubbio sfruttando il pacchetto disponibile in formato “.rpm” e convertendolo in “.deb”, per poi installarlo grazie al gestore di pacchetti della nostra distribuzione.

I passi per avere disponibile un pacchetto *Debian* di *VMware* sono i seguenti:

- download del pacchetto “.rpm” dal sito ufficiale di *VMware*;
- installazione del pacchetto “alien”, un software capace di convertire pacchetti delle più comuni distribuzioni *GNU/Linux* disponibili;
- conversione da “.rpm” a “.deb” grazie al software “alien”;

Prima di procedere con l’installazione del pacchetto appena generato, è necessario installare altri software tra cui sorgenti degli header del kernel in uso, il compilatore “GNU C Compiler” ed alcune librerie, in particolare:

- linux-headers-‘uname -r’-686
- gcc
- libX11-6
- libxtst6
- libxt6
- libxrender1

Nonostante l’utilizzo di un gestore dei pacchetti intelligente, è necessario installare “a mano” i software nella lista perchè il pacchetto generato grazie ad “alien” non contiene la lista delle dipendenze al suo interno. Il “GCC” è uno tra i pacchetti fondamentali, se non già installato, poichè senza di esso, *VMware* non compila il modulo ad-hoc per il kernel in uso.

Ora è possibile installare *VMware*:

```
root@server:/root# dpkg -i vmware-x.y.z.deb
```

Al termine dell’installazione verrà automaticamente lanciato il software di configurazione CLI che permette di impostare i più comuni attributi come ad esempio il/i tipo/i di networking (bridge, NAT...) e le directory di lavoro.

A questo punto il nuovo server host è pronto per l’utilizzo e se tutto è andato per il verso giusto sarà possibile creare/avviare/gestire sistemi guest.

Verdetto

Un punto a favore della tecnologia di *XEN* è il fatto che ad un possibile aggiornamento del kernel risponde in maniera più adatta. Se viene rilasciato un

update del kernel, dopo averlo installato e riavviato il sistema, tutto rimane come prima, senza dover riconfigurare alcunchè. In maniera inversa invece, *VMware* necessita della ricompilazione del modulo ad-hoc per il kernel se quest'ultimo dovesse subire qualche aggiornamento. Da notare comunque, è il fatto che un aggiornamento del kernel (nel 99% dei casi fornito dal team di sicurezza e non da quello degli sviluppatori) è un avvenimento abbastanza raro se si utilizza distribuzioni definite "stable" e quindi l'avvento di una nuova release diventa un evento decisamente importante che risolve, nella maggior parte dei casi, problemi di sicurezza.

5.2 Creazione VM/installazione guest OS

In questa sezione verrà considerata la semplicità con cui è possibile rendere operativo un sistema guest all'interno di una macchina virtuale. Sarà presa in considerazione l'installazione base dei due sistemi di virtualizzazione, con il relativo client integrato.

XEN

XEN presenta, di default, una CLI relativamente semplice, anche se di primo impatto risulta poco intuitiva. Le modalità per creare una nuova VM sono essenzialmente due: un file di configurazione che verrà letto dall'utility di creazione oppure sfruttando la stessa utility e passandole ogni attributo come opzione da riga di comando.

È un'interfaccia funzionale adatta particolarmente all'ambiente server in cui non è presente un ambiente grafico. Una volta presa familiarità col set di istruzioni, la creazione ed il deploy delle macchine virtuali diventa molto veloce e pratico. Anche l'avvio di una VM avviene tramite un client testuale che rende possibile ogni operazione, compreso un terminale sul sistema virtualizzato.

Ricordando che *XEN* necessita di una versione modificata di *Microsoft Windows* per virtualizzarlo, nei miei test ho creato macchine virtuali solo con sistema *GNU/Linux*. La via più semplice per installare una distribuzione è sfruttare, in ambiente Debian, il pacchetto "debootstrap", il quale dà la possibilità di creare una distribuzione "from scratch" senza utilizzare i comuni gestori di pacchetti (dpkg ed apt). Ad una mano poco esperta, tutto ciò risulta sicuramente macchinoso e non intuitivo ed è anche normale, si sta creando una specie di nuova distribuzione *GNU/Linux*. Una volta creato il nuovo sistema ed avviato, grazie ad un'utility in CLI sarà possibile accedervi.

Per chi volesse provare *XEN* senza però creare una VM da zero, esistono svariati siti internet in cui è possibile scaricare template e VM già pronte per l'utilizzo, solo da avviare.

VMware

VMware mette a disposizione un client grafico chiamato “VMware Console” che permette di eseguire tutte le operazioni disponibili su una VM: creazione, avvio, riavvio, reset, effettuare snapshot e collegare periferiche (eventualmente in runtime). Questa console crea una connessione remota verso il server sulla porta “902/tcp” e si autentica sfruttando “PAM”, di conseguenza tutti gli account di sistema, se non opportunamente bloccati, possono accedervi.

Anche *VMware* mette a disposizione una CLI per svolgere tutte le attività di creazione e gestione dei sistemi guest, ma viste le potenzialità del suo client grafico, il terminale è poco utilizzato. Un'altra funzionalità interessante è la completa gestione remota delle macchine virtuali. La possibilità di avere la console grafica in esecuzione su un client remoto, magari a migliaia di chilometri di distanza, è un fattore non indifferente nella scelta. Dalla mia esperienza, ho rilevato alcuni problemi di lentezza nella connessione accedendo tramite console ad un server con interfaccia grafica su un'adsl italiana. Il collo di bottiglia è sicuramente la banda disponibile in upload del suddetto server, ma a livello client VMware non è possibile impostare una risoluzione adeguata: quest'operazione va fatta sul server stesso, ma anche con 16bit di colori ho rilevato lo stesso comportamento. Tutti questi problemi spariscono se il server remoto ha un'interfaccia testuale, come ad esempio gran parte dei server *GNU/Linux*.

Per quanto riguarda l'installazione di un sistema guest, *VMware* si comporta esattamente come un sistema non virtualizzato e cioè permette l'avvio della VM da un lettore CD/DVD fisico oppure tramite un'immagine ISO, presente lato server, di un sistema operativo. Tutto ciò è configurabile direttamente dalla console. L'unica pecca rimane, dal mio punto di vista, lo scarso supporto USB, compreso il boot. Secondo informazioni ufficiali però, nelle prossime release di “VMware server” vi sarà un notevole aumento nel supporto di dispositivi USB.

Verdetto

Sul terreno di gioco “default client ed installazione guest” il mio punto va senza dubbio a *VMware* per la facilità ed il tempo necessario con cui è possibile creare ed avviare una macchina virtuale partendo da zero. Le svariate interfacce di amministrazione aggiuntive sono senza dubbio un valore aggiunto, ma è un campo troppo vasto per essere valutato oggettivamente, quindi mi sono attenuto, come detto in precedenza, alla condizione di valutazione delle sole interfacce ufficiali e disponibili di default.

5.3 Costi e metodi di implementazione

Un fattore più che rilevante nell'introduzione di una struttura server virtualizzata sono senza dubbio i costi di conversione ed implementazione del nuovo sistema. Nel caso specifico prenderemo in esame le versioni commerciali di *XEN* e *VMware*, visto che quelle base sono gratuite per entrambi.

Le soluzioni disponibili per quanto riguarda il settore enterprise si chiamano rispettivamente “*Citrix XenServer Enterprise Edition*” e “*VMware Infrastructure*”. Prestazionalmente ed a livello di feature sono due prodotti molto simili e scegliere uno o l'altro probabilmente è solo questione di simpatia. Un punto a favore di *VMware* è probabilmente l'anzianità nel settore e forse è proprio per questo che possiede maggiori quote di mercato rispetto al rivale. Dal mio punto di vista, essendo *XEN* un software opensource, ha maggiori probabilità di sviluppo in tempi rapidissimi, grazie all'immensa community che si è formata intorno a questo progetto.

In entrambi i casi, l'ordine di grandezza dei prezzi per mettere in piedi una soluzione di server virtualizzati è migliaia di euro. La differenza principale sta nel fatto che *VMware*, data la sua politica puramente commerciale, impone alcuni vincoli che fanno lievitare il costo. Un esempio pratico è l'obbligatorietà di avere un server equipaggiato con “Microsoft Windows 2003 server” su cui gira “Microsoft SQL Server 2005” per la gestione del cluster di server fisici ESX. Un altro vincolo è l'acquisto dell'assistenza, che come costo è lo stesso di quella di *XEN* che però è opzionale.

A conti fatti, la soluzione basata su *XEN* costa circa la metà rispetto a quella proposta da *VMware*. Nonostante condivida e sia completamente a favore della filosofia del software libero ed opensource, rimango comunque dell'idea che siano due prodotti ancora abbastanza diversi e ritengo il secondo più maturo ed adatto ad un'utenza enterprise: di recente è stato quotato in borsa ed è stato scelto da *SUN Microsystems* come partner nel mercato virtualizzazione.

Un ulteriore aspetto legato indirettamente ai costi di messa in opera è il seguente: supponendo di voler passare ad una soluzione virtualizzata dei server aziendali, come ci si comporta con tutti i server fisici esistenti? Considerando un ambiente eterogeneo in cui sono presenti sia server *Microsoft Windows* sia server *GNU/Linux*, il problema del porting non è da sottovalutare. Per quanto riguarda *GNU/Linux* è noto che il porting è molto meno problematico rispetto alla stessa operazione su sistemi *Microsoft*. Spesso è sufficiente copiare pari pari il sistema, rivedere i moduli caricati al boot ed installare nuovamente il bootloader impostando i nomi corretti per le nuove partizioni. Con sistemi “windows” invece, tutto ciò non è possibile ed altamente sconsigliato dalla

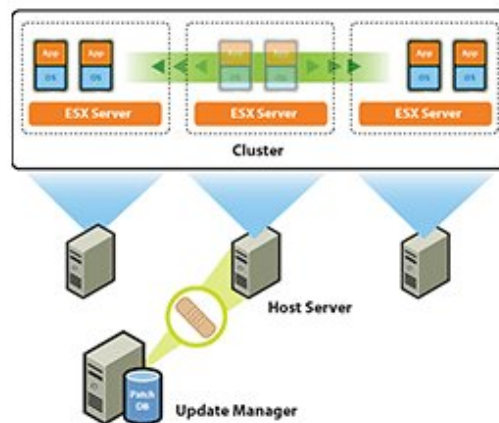


Figura 5.1: Funzionamento di VMware Update Manager

Microsoft stessa. Per ovviare a questi problemi, *VMware* ha rilasciato un software che permette di convertire un sistema nella relativa VM in runtime.

Se la conversione è un dettaglio da non sottovalutare, allora *VMware* guadagna un altro punto. In più, per far girare efficientemente sistemi *Microsoft* su *XEN*, è altamente consigliato avere un processore con estensioni di virtualizzazione implementate in hardware (come già visto nei capitoli precedenti “Intel IVT” o “AMD Pacifica”).

Per quanto riguarda i costi di manutenzione e le procedure di aggiornamento dei sistemi host, entrambi si comportano più o meno allo stesso modo. Il costo annuale del supporto tecnico è quantificato nell’ordine delle centinaia di euro per ambedue le soluzioni ed i costi di aggiornamento per eventuali nuove release sono compresi nella licenza rilasciata all’acquisto. Ciò non esclude comunque il fatto che nuove funzionalità non comprese nelle release precedenti possano essere integrabili pagando un supplemento (cosa fin’ora non riscontrata nè in *VMware* nè in *XEN*).

Prendendo sempre in esame le soluzioni enterprise proposte e considerando il fatto che, di solito, i sistemi che le sfruttano sono basati su cluster, le procedure di aggiornamento non introducono particolari problemi. Entrambe le soluzioni, nel momento in cui viene rilasciata una nuova release, rendono disponibili online le relative immagini. Una volta masterizzate, viene avviata la procedura di aggiornamento sui singoli nodi, migrando il carico di lavoro sugli altri e rendendo di fatto nullo il downtime complessivo.

Recentemente *VMware* ha reso disponibile nel suo pacchetto “Infrastructure” il *VMware Update Manager*, un’integrazione alla propria suite enterprise che permette di gestire in maniera centralizzata ed automatica gli aggiorna-

menti. La figura 5.1 esprime nel migliore dei modi tale funzionamento.

5.4 Problemi riscontrati

Durante la mia esperienza di tirocinio, avendo avuto la possibilità di sperimentare in maniera abbastanza approfondita l'utilizzo di "VMware server", mi sono imbattuto in alcuni problemi che poi si sono rivelati molto comuni nel campo della virtualizzazione. I due principali rompicapi sono descritti di seguito.

5.4.1 Timing VM

Problema della sincronizzazione dell'orologio tra sistema host e sistemi guest.

Forse su un server non si legge la data quotidianamente, ma appena si ha a che fare con file di log, allora balza agli occhi il problema più largamente diffuso nei sistemi virtualizzati: l'orologio del sistema guest che corre troppo o troppo poco (quest'ultima è la situazione più diffusa). Nei calcolatori attuali, il tempo viene calcolato in base ai cicli di clock della macchina, quindi è evidente che introducendo un layer tra il processore fisico e quello virtuale si genera un ritardo, che se non gestito correttamente provoca grossi errori di misura (nell'ordine di due volte).

Una soluzione empirica molto gettonata su numerosi forum di discussione è quella di disabilitare il supporto multiprocessore nelle macchine virtuali. Così facendo si emula un unico processore, che, a detta di molti (tra cui lo stesso team di sviluppo di *VMware* [39]), dovrebbe gestire in maniera più opportuna il clock di sistema.

La seconda soluzione, molto più logica dal mio punto di vista, è quella di sfruttare il "Real Time Clock" (il cui modulo nel kernel 2.6 è "rtc") di *GNU/Linux* tra il sistema host e quello guest. Caricando questo modulo sul guest, già attivo di default su molti sistemi, e rendendo disponibile il device "/dev/rtc" sull'host si ottiene una buona sincronizzazione del clock fisico con quello virtuale, andando di fatto a risolvere il problema.

Nella mia esperienza il problema del clock è stato riscontrato solo con l'utilizzo di *VMware*, mentre *XEN* non ne ha risentito. La curiosità di scoprire il motivo mi ha portato ad una soluzione più che logica: visto che la CPU da me utilizzata non aveva alcuna estensione hardware per la virtualizzazione, *XEN* operava con il paradigma della paravirtualizzazione, mettendo quindi a disposizione particolari API per far interagire il sistema guest direttamente con l'hardware, cosa che permise di non saltare cicli di clock e mantenere l'ora di sistema corretta.

Un'ulteriore nota per quanto riguarda *VMware* sono i cosiddetti “VMware tools” da installare sui sistemi guest. Sono una particolare famiglia di utility composta da framework di sviluppo per programmatori e vari software che ottimizzano il lavoro della macchina virtuale rendendola conscia di non essere un sistema fisico. Lo sviluppo di questi “tool” sta migliorando col tempo, ma la loro versione per *Microsoft Windows* è ancora nettamente superiore rispetto a quella disponibile per *GNU/Linux*. Tutto ciò si concretizza nel fatto che, anche in questo caso particolare del problema del “timekeeping”, i sistemi *Microsoft Windows* virtualizzati non presentano alcun difetto sull'orologio di sistema, andando a sincronizzarsi correttamente con la macchina host.

5.4.2 Dischi virtuali

Problema della dimensione effettiva dei dischi virtuali sullo storage fisico.

Ancora una nota pseudo-negativa legata al sistema *VMware* ed in particolare all'implementazione dei dischi virtuali “non-preallocati”. Un disco virtuale non-preallocato è un file che andrà a contenere i dati della VM. Al momento della sua creazione ha dimensioni estremamente ridotte che crescono man mano che si riempie il disco virtuale. Il motivo di dare all'utente questa possibilità è ovviamente quello di non dover sprecare preziosi gigabyte di spazio fisico quando il sistema virtuale non li usa effettivamente. Sono quindi file destinati a crescere fino a quando il disco virtuale non raggiunge la sua capacità massima.

Il problema nasce al verificarsi di una situazione decisamente semplice: supponiamo di scaricare un grosso file da internet all'interno della VM; il file contenente il disco virtuale aumenterà la sua dimensione in base al peso del file scaricato. Successivamente spostiamo quel file su un disco USB e lo eliminiamo dal nostro disco virtuale. Risultato: il file del disco non ha diminuito la sua dimensione rispetto al momento successivo al download. Tutto ciò si spiega andando a vedere come lavora il filesystem in uso sulla VM. Se prima di allocare nuovi blocchi (che in questo caso si traduce nell'aumento della dimensione del file) scrivesse sui blocchi appena cancellati non ci sarebbe questo problema e l'aumento della dimensione del file sarebbe proporzionale all'aumento dello spazio disco utilizzato sul sistema guest. L'*hypervisor* di *VMware* ovviamente non ha gli strumenti per gestire una situazione del genere in runtime, poichè entrerebbe in conflitto col filesystem della VM, generando probabilmente qualche catastrofe. Nonostante ciò, all'interno dei “VMware tools” è presente un'utility di “Shrink del disco”, qualcosa di simile a quello che gli utenti windows chiamano “deframmentazione” (che su un filesystem fatto bene è ridotta al minimo). Quest'utility è molto comoda e funziona egregiamente, l'unica grossa pecca è che per funzionare come ci si aspetta, necessita di

uno spazio disponibile sul sistema host paragonabile a quello utilizzato dal file incriminato che cresce troppo. Spesso (forse troppo spesso) ci si ritrova a dover effettuare lo shrink del disco virtuale proprio perchè si è in carenza di spazio fisico e quindi si cade nel medesimo problema, la cui soluzione più semplice è appoggiarsi temporaneamente ad un disco esterno. Inoltre, la fase di shrink, blocca completamente ogni attività del sistema virtualizzato rendendolo del tutto inutilizzabile, nonostante venga lanciata dall'interno del sistema stesso.

5.5 Conclusioni

Sulla rete sono presenti opinioni discordanti su chi possieda la maggior qualità e quota di mercato: com'è prevedibile anche gli stessi *XEN* [40] e *VMware* [41] rilasciano documenti in cui ognuno sostiene la propria posizione dominante. Nonostante l'anzianità del secondo, come ripetuto più volte, *XEN* ha raggiunto un'ottima maturità e se per quanto riguarda alcune funzionalità è ancora in svantaggio, per altre è lui il prodotto di riferimento.

Attualmente credo che tra i due sia ancora in vantaggio *VMware*, sia per la facilità con cui è possibile creare nuovi ambienti virtuali, sia per il supporto ai sistemi operativi *Microsoft Windows* come guest (non imponendo modifiche software o primitive hardware per la virtualizzazione) che, nonostante la mia simpatia sia notoriamente rivolta al mondo *GNU/Linux*, rappresentano al giorno d'oggi ancora un larga maggioranza nel panorama informatico.

Tabella riassuntiva

Operazione	XEN	VMware
Installazione host	10	8
Creazione VM	7	9
Installazione guest OS	6	10
Supporto guest OS	6	10
Costi implementazione	9	7
Manutenzione/aggiornamento	8	9
Problema timing	10	7
Console	7	10
Conversione sistemi	6	8
Aggiornamento	8	8
Totale	77	86

Capitolo 6

Conclusioni e prospettive

Il mercato della virtualizzazione è uno tra quelli che hanno avuto il maggior sviluppo nell'ultimo anno e sicuramente nel futuro più immediato riserverà grosse novità sia a livello aziendale che personale. Al giorno d'oggi sono già presenti numerose soluzioni, alcune simili, altre meno, ma ognuna con buone porzioni di mercato.

La maggior parte delle volte si tende ad associare al concetto di “virtualizzazione del sistema operativo” solo la possibilità di utilizzare un altro sistema all'interno del proprio computer. Tutto ciò è ovviamente estremamente riduttivo, in quanto ormai esistono già data center completamente virtualizzati che gestiscono dati, applicazioni e banda internet al pari dei loro equivalenti fisici. Inoltre, per quanto possano essere utili le virtualizzazioni lato client, il mercato si sta orientando sempre più sul lato opposto, a livello server.

Nei prossimi mesi è probabile che ci sarà un aumento nel settore virtualizzazione client intesi come terminali che sfruttano un server centrale per qualsiasi loro attività, fornendo un ambiente terminal server virtualizzato. Si sfrutterà quindi la virtualizzazione nell'ambito dei terminali, una volta detti “stupidi” o “diskless”, arrivando ad avere la propria postazione di lavoro ovunque nel mondo. Essendo poi VM, si potranno sfruttare tutti i relativi vantaggi, dalla semplicità di deploy al backup/restore semplificati ed in tempi brevissimi. Tutto ciò probabilmente si concretizzerà fondamentalmente in due modi: virtualizzazione dell'intero sistema client ed accesso remoto ad esso, oppure, una singola macchina virtuale ed un'istanza di essa per ogni client connesso via rete. Quest'ultima è di gran lunga la soluzione meno onerosa in termini di prestazioni, ma comporta una maggior complessità a livello di progettazione e realizzazione dell'ambiente virtuale stesso. Tutto ciò comporta inoltre problemi legati al licensing di ogni singola istanza della macchina virtuale, dovuti al fatto che non si sa o non si è ancora deciso se debba essere considerata equivalente ad un sistema fisico o semplicemente un “accesso remoto” ad una

risorsa.

Dal mio punto di vista entro i prossimi 5 anni si assisterà ad un incremento consistente della virtualizzazione lato server che andrà a sostituire gran parte dei sistemi attuali. Inoltre, lato utente, come accennato in questo ultimo capitolo, vi saranno grandi novità riguardo la centralizzazione dei sistemi, scaricando sempre più i client e spostando le funzionalità lato server, un po' come sta succedendo con tutti i servizi e le applicazioni fruibili via web.

Bibliografia

- [1] Virtualizzazione,
<http://it.wikipedia.org/wiki/Virtualizzazione>,
Wikipedia
- [2] Virtualizzazione x86,
http://it.wikipedia.org/wiki/Virtualizzazione_X86,
Wikipedia
- [3] Virtual machine,
http://it.wikipedia.org/wiki/Virtual_machine,
Wikipedia
- [4] Emulatore,
<http://it.wikipedia.org/wiki/Emulatore>,
Wikipedia
- [5] Hypervisor,
<http://it.wikipedia.org/wiki/Hypervisor>,
Wikipedia
- [6] Kernel,
<http://it.wikipedia.org/wiki/Kernel>,
Wikipedia
- [7] Exokernel,
<http://en.wikipedia.org/wiki/Exokernel>,
Wikipedia

-
- [8] MIT Exokernel Operating System,
<http://pdos.csail.mit.edu/exo.html>
- [9] Tecniche di virtualizzazione, teoria e pratica,
http://www.mondodigitale.net/Rivista/07_numero_1/Boari_p._38-49.pdf,
Mondo Digitale (01/2007), 38–49
- [10] BABEL,
<http://www.babel.it/virtualizzazione.htm>
- [11] La virtualizzazione,
<http://www.andreabeggi.net/2007/03/19/la-virtualizzazione/>
Andrea Beggi
- [12] Server consolidation tramite virtualizzazione,
<http://www.andreabeggi.net/2007/03/22/server-consolidation-tramite-virtualizzazione/>
Andrea Beggi
- [13] VMware,
<http://www.vmware.com/>
- [14] Microsoft Virtual Server,
<http://www.microsoft.com/italy/server/virtualserver/>,
Microsoft
- [15] Microsoft Virtual PC,
<http://www.microsoft.com/windows/products/winfamily/virtualpc/>,
Microsoft
- [16] Xen,
<http://www.cl.cam.ac.uk/research/srg/netos/xen/>,
University of Cambridge Computer Laboratory
- [17] XenSource,
<http://www.citrixserver.com/>,
Citrix

- [18] Xen introduction,
<http://en.wikipedia.org/wiki/Xen>,
Wikipedia
- [19] User-mode Linux Kernel,
<http://user-mode-linux.sourceforge.net/>
- [20] User-mode Linux,
http://en.wikipedia.org/wiki/User-mode_Linux,
Wikipedia
- [21] UML wiki,
<http://uml.jfdi.org/uml/Wiki.jsp>
- [22] Introduction to User Mode Linux,
<http://www.informit.com/articles/article.aspx?p=481866>,
InformIT
- [23] Introduzione a UML,
<http://www.blogin.it/uml-grolug.pdf>,
Antonio Anselmi
- [24] Storage Area Network,
http://it.wikipedia.org/wiki/Storage_Area_Network,
Wikipedia
- [25] Network Attached Storage,
http://it.wikipedia.org/wiki/Network_Attached_Storage,
Wikipedia
- [26] File system distribuito,
http://it.wikipedia.org/wiki/File_system_distribuito,
Wikipedia
- [27] Distributed file system,
http://en.wikipedia.org/wiki/Distributed_file_system,
Wikipedia

-
- [28] Fibre Channel,
http://it.wikipedia.org/wiki/Fibre_Channel,
Wikipedia
- [29] SCSI,
http://it.wikipedia.org/wiki/Small_Computer_System_Interface,
Wikipedia
- [30] iSCSI,
<http://en.wikipedia.org/wiki/ISCSI>,
Wikipedia
- [31] ATA Over Ethernet,
http://en.wikipedia.org/wiki/ATA_over_Ethernet,
Wikipedia
- [32] Network File System,
http://it.wikipedia.org/wiki/Network_File_System,
Wikipedia
- [33] Server Message Block,
<http://it.wikipedia.org/wiki/CIFS>,
Wikipedia
- [34] Andrew file system,
http://it.wikipedia.org/wiki/Andrew_file_system,
Wikipedia
- [35] ZFS,
[http://it.wikipedia.org/wiki/ZFS_\(file_system\)](http://it.wikipedia.org/wiki/ZFS_(file_system)),
Wikipedia
- [36] Da Oracle un server di virtualizzazione gratuito,
<http://punto-informatico.it/p.aspx?i=2113457>,
Punto Informatico

-
- [37] Virtualizzazione: SUN scende in campo,
<http://punto-informatico.it/p.aspx?i=2143315>,
Punto Informatico
- [38] Hyper-V,
<http://en.wikipedia.org/wiki/Hyper-V>,
Wikipedia
- [39] Timekeeping in VMware Virtual Machines,
http://www.vmware.com/pdf/vmware_timekeeping.pdf,
VMware Inc.
- [40] Hypervisor performance comparison with ESX data,
http://www.citrixserver.com/Documents/hypervisor_performance_comparison_1_0_5_with_esx-data.pdf
Citrix
- [41] Hypervisor performance,
http://www.vmware.com/pdf/hypervisor_performance.pdf
VMware Inc.

Ringraziamenti

I classici ringraziamenti in cui ci si dimentica sempre di qualcuno non mi sono mai piaciuti, per cui i miei “grazie” in informaticese vanno a:

famiglia.*

amici.*

uniud.scienze.informatica.*

colleghi.*

.

.

.

* (così nessuno può lamentarsi).