Virtualizzazione e Macchine Virtuali

Gabriele D'Angelo, Ludovico Gardenghi

{gda, garden}@cs.unibo.it

http://www.cs.unibo.it/~gdangelo/

http://www.cs.unibo.it/~gardengl/

Università di Bologna

Corso di Laurea in Scienze dell'Informazione

Dicembre, 2006

Scaletta della lezione

- Virtuale? In che senso?
- Breve storia
  - IBM VM/370
  - Gli anni ‘90
  - Anni recenti
- Metodologie di virtualizzazione
- Macchine virtuali complete
- Paravirtualizzazione
- Virtualizzazione del sistema operativo
- Bibliografia

Virtuale? In che senso?

- Partiamo da una definizione di **virtuale**: "*oggetto non realmente esistente, non avente una sua fisicità*"
- Ma un software non è un'entità fisica, quindi di cosa stiamo parlando?
- **Macchina virtuale**: un software che crea fra il computer e l'utente finale un ambiente all'interno del quale è possibile eseguire software

Breve storia: IBM VM/370

- Il concetto di macchina virtuale nasce negli anni ’60 da parte di IBM con il suo prodotto VM/370
- La spinta viene dalla necessità di **condividere** (costose) risorse tra più utenti
- Il sistema proposto da IBM è composto da più componenti, ciascun utente ha a disposizione un sistema operativo mono-utente denominato CMS (Conversational Monitor System)
- Il CMS anziché essere direttamente eseguito sull’hardware si appoggia ad uno strato di virtualizzazione chiamato CP (Control Program) che, a sua volta, ha il controllo della macchina reale. Il suo scopo è quello di mostrare al livello superiore un’astrazione la cui **interfaccia sia uguale a quella dell’hardware sottostante**
- In questo modo le applicazioni non devono essere riscritte per la macchina virtuale ma rimangono **totalmente compatibili**

Breve storia: IBM VM/370

- La complessità del CP risiede nell'offrire questa astrazione in modo **efficiente** e **sicuro** (caratteristiche comuni anche in seguito)
- In questo caso per sicuro intendiamo che deve evitare che le singole istanze di CMS possano "uscire" dallo spazio a loro riservato e causare, potenzialmente, danni di vario tipo agli altri utenti
- In principio quindi per "virtualizzazione" si intendeva: un meccanismo per **suddividere** una macchina reale in più "copie" della stessa, garantendo la compatibilità del software, isolamento e, per quanto possibile, **prestazioni simili a quelle originali**
- Il passare del tempo, con l'evoluzione dell'hardware, la diffusione delle macchine personali e i sistemi operativi multi-utente hanno reso questo approccio piuttosto raro

Breve storia: gli anni ‘90

- I motivi, che negli anni ’90, hanno portato nuovo interesse alle tecniche di virtualizzazione sono completamente diversi
- La ricerca di un meccanismo per rendere un programma **portabile su più architetture**, con il numero minimo di interventi e con l’invasività minima possibile (es. senza la necessità di ricompilare)
- La soluzione di maggior diffusione è legata al linguaggio di programmazione Java e in particolare all’utilizzo di una **macchina virtuale simulata**: la Java Virtual Machine (JVM)
- Il software Java è basato sull’utilizzo di bytecode che viene interpretato ed utilizza le funzionalità offerte dalla JVM invece di quanto offerto direttamente dal sistema operativo
- A sua volta la JVM è responsabile della **traduzione delle richieste** nei confronti del sistema reale di esecuzione

Breve storia: anni recenti

- Negli anni recenti c'è stato un rinnovato interesse per la **virtualizzazione dell'hardware**, in parte a discapito dell'interesse per la virtualizzazione di singole applicazioni o linguaggi, **non solo su sistemi server ma anche desktop**

- Gli scopi sono molteplici, ad esempio:
  - **Compatibilità** delle applicazioni con architetture diverse da quelle disponibili od obsolete
  - **Testing** e sviluppo di nuovi sistemi
  - **Prototipazione** di architetture complesse
  - **Isolamento** di dati ed utenti

Non tutte le virtualizzazioni sono uguali

Esistono vari approcci per la virtualizzazione, molto diversi come filosofia e di conseguenza come implementazione

- **Virtualizzazione dell'architettura del processore**:

  La macchina virtuale ricrea un **ambiente hardware completo** o, comunque, **reimplementa** (appoggiandosi eventualmente all'hardware reale per migliorare le prestazioni) le istruzioni eseguite dal sistema operativo e dalle applicazioni sotto il suo controllo. Un approccio di questo tipo permette di realizzare macchine complete; portando la macchina virtuale su più architetture si può ottenere la portabilità in ambienti eterogenei di interi sistemi operativi e di tutte le applicazioni

Non tutte le virtualizzazioni sono uguali

- **System call trapping**:

  Questi metodi si basano **sull'intercettazione delle chiamate di sistema dei processi interni** alla virtualizzazione. Manipolando i risultati è possibile far eseguire al sistema reale (detto anche sistema host) le operazioni, mostrando poi ai processi interni della macchina virtuale (guest) **effetti potenzialmente diversi da quelli ottenuti**

  Questo approccio rende le macchine virtuali solitamente più efficienti ma anche meno portabili. I sistemi per intercettare le system call sono **strettamente dipendenti dal sistema operativo e dall'architettura della macchina reale**

  Normalmente richiedono versioni ad-hoc del sistema operativo da virtualizzare

Macchine virtuali complete (QEMU)

Con **macchine virtuali complete** intendiamo quelle che permettono la *virtualizzazione di un completo calcolatore* con una determinata architettura. Questa tipologia di macchina è usata per eseguire un sistema operativo completo e le relative applicazioni

- **QEMU:** permette di emulare diverse architetture fra cui Intel x86 / x86-64, PowerPC e Motorola M68K. Di conseguenza supporta un'ampia gamma di sistemi operativi, consente la presenza simultanea su una singola macchina ospite di **più architetture e sistemi operativi**. QEMU virtualizza l'hardware e il processore, ha un'invasività minima ed è eseguibile anche con i **soli permessi utente**. Per migliorare le prestazioni QEMU fa uso di **traduzione dinamica delle istruzioni dei processi virtualizzati**

Macchine virtuali complete (VMware)

- **VMware:** si tratta di un progetto proprietario. Permette la virtualizzazione di macchine complete su architetture x86, i sistemi host e guest possono essere Microsoft oppure Linux
  - Non si tratta di un singolo prodotto ma piuttosto di una famiglia di prodotti (es. GSX server ed ESX server)
  - La maggior differenza tra i vari prodotti risiede nella **tecnica per l'astrazione della macchina virtuale**
  - Nel primo caso (GSX server) è necessaria la presenza di un sistema operativo host che esegua la macchina virtuale
  - Nel secondo caso (ESX server) la macchina virtuale è in grado di lavorare **direttamente sull'hardware senza bisogno di alcun sistema operativo host**
  - Attraverso l'uso di varie tecniche di ottimizzazione, le prestazioni che vengono ottenute sono in linea di massima piuttosto buone

Macchine virtuali complete (VMware)

- È possibile personalizzare l'hardware della macchina virtuale, decidendo il numero delle CPU e varie tipologie di periferiche
- La presenza di un Virtual Infrastructure Layer permette di unire più macchine host, mostrandosi al sistema guest come una macchina unica
- Fra gli host partecipanti all'infrastruttura virtuale è possibile avere mobilità delle macchine virtuali **a caldo e senza interruzioni**

Macchine virtuali complete (Microsoft)

- **Microsoft Virtual PC e Virtual server:** sistema di virtualizzazione Microsoft. Sono due versioni di un singolo prodotto, una orientata all'uso personale e una pensata per l'utilizzo server
  - Il prodotto fornisce virtualizzazione dell'hardware (solamente architettura x86)
  - Come altri prodotti o progetti anche questo consente la creazione di macchine virtuali che possono comunicare con l'esterno tramite un'architettura di rete virtuale
  - La differenza tra i due prodotti riguarda la scalabilità e la presenza di emulazione per l'hardware usato in ambito server

Paravirtualizzazione

Con il termine **paravirtualizzazione** ci si riferisce ad una particolare forma di virtualizzazione: la macchina non traduce alcune istruzione, **si limita a creare uno strato minimale di software per assicurare la gestione delle singole istanze di macchine virtuali e il loro isolamento**

Un esempio è **Xen**, un prodotto patrocinato da IBM che risulta particolarmente efficiente. **Non supporta direttamente tutto l'hardware della macchina reale ma si appoggia ai driver del sistema operativo della prima macchina virtuale ad essere eseguita**

Supporta la migrazione a caldo degli ambienti virtuali

Virtualizzazione del sistema operativo (UML)

In questa categoria le macchine virtuali si applicano **a livello di processo**, mostrando quindi come interfaccia quella di un sistema operativo

- **User-Mode Linux (UML):** nella pratica si tratta del progetto maggiormente sviluppato e diffuso di questa categoria. Il suo obiettivo è rendere possibile l'esecuzione di un **kernel Linux in modalità utente**.
- Utilizzi classici:
  - Kernel debugging
  - Isolamento di processi;
  - L'uso di ambienti diversi senza avere più macchine reali, per la creazione di sandbox (ad esempio per operazioni pericolose)

Virtualizzazione del sistema operativo (UML)

- UML realizza l'ambiente virtuale modificando il kernel, il suo obiettivo è riuscire ad eseguirlo come un normale processo user-mode
- L'interfaccia messa a disposizione da UML è quella delle system-call, che vengono redirette al kernel reale
- Questo procedimento viene reso possibile dalla system call ptrace. UML controlla i processi interni al suo ambiente virtuale tramite tracciamento delle chiamate di sistema: un processo che esegue una system call viene intercettato da UML, che reagisce modificandone il comportamento

Virtualizzazione del sistema operativo (Virtuozzo)

- **Virtuozzo e OpenVZ:** il suo scopo è quello di consentire la presenza di "Virtual Private Server" (VPS) Linux su uno o più host
  - Ciascuno dei VPS è una **collezione di applicazioni che condividono lo stesso spazio virtuale**
  - Ambienti virtuali diversi possono differire per visione del file system, della rete, e per le zone di memoria accessibili
  - A differenza di UML, Virtuozzo **non prevede l'esecuzione di copie multiple del Kernel**: l'unica copia attiva è quella dell'host e tutti i VPS si appoggiano direttamente a quella
  - Le prestazioni di questo meccanismo sono molto buone, offre funzionalità avanzate come la migrazione a caldo di VPS
  - Recentemente una parte del prodotto è stata rilasciata con licenza libera sotto il nome di OpenVZ

Bibliografia

- **Virtual Machines: Versatile Platforms for Systems and Processes** (The Morgan Kaufmann Series in Computer Architecture and Design). Jim Smith, Ravi Nair
- **Comparison of Virtual Machines,** http://en.wikipedia.org/wiki/Comparison_of_virtual_machines
- **What is virtualization?** http://www.vmware.com/virtualization/
- **User Mode Linux (UML),** http://user-mode-linux.sourceforge.net/
- **OpenVZ**, http://openvz.org/