

proca

fede andrucci

September 2021

Contents

1 Virtualizzazione

Virtualizzare un sistema (hardware e software) significa presentare all'utente una visione delle risorse del sistema diversa da quella reale. Ciò è possibile introducendo un **livello di indirectione** tra la vista logica e quella fisica delle risorse.

Quindi l'obiettivo della virtualizzazione è quello di disaccoppiare il comportamento delle risorse di un calcolatore dalla loro realizzazione fisica. Quindi appaiono diverse da quelle effettive della macchina. Il software che si occupa di virtualizzare in parole semplici divide le risorse reali nel numero di macchine virtuali necessarie. Quindi ogni macchina virtuale avrà la sua CPU, GPU, RAM, ecc...

Esempi di virtualizzazione:

- **Virtualizzazione a livello di processo:** i sistemi multitasking permettono l'esecuzione contemporanea di più processi, ognuno dei quali dispone di una macchina virtuale dedicata. Questo tipo di virtualizzazione viene realizzata dal kernel del sistema operativo.
- **Virtualizzazione della memoria:** in presenza di memoria virtuale, ogni processo vede uno spazio di indirizzamento di dimensioni indipendenti dallo spazio fisico effettivamente a disposizione. Anche questa virtualizzazione è realizzata dal kernel.
- **Astrazione:** un oggetto astratto (risorsa virtuale) è la rappresentazione semplificata di un oggetto (risorsa fisica), quindi esibendo le proprietà significative per l'utente e nascondendo i dettagli realizzativi non importanti

1.1 Virtualizzazione di un Sistema di Elaborazione

Tramite la virtualizzazione una singola piattaforma hardware viene condivisa da più elaboratori virtuali, ognuno gestito da un proprio sistema operativo. Il

disaccoppiamento viene realizzato dal **Virtual Machine Monitor (VMM)**, il cui compito è quello di consentire la condivisione da parte di più macchine virtuali di una singola piattaforma hardware.

Quindi il **VMM** è il **mediatore unico** nelle interazioni tra le macchine virtuali e l'hardware, il quale garantisce: **isolamento tra le VM** e **stabilità del sistema**.

1.2 Tecniche del VMM

1.2.1 Emulazione

L'emulazione è l'insieme di tutti quei meccanismi che permettono l'esecuzione di un programma compilato su un determinato sistema di girare su un qualsiasi altro sistema differente da quello nel quale è stato compilato. Quindi vengono emulate interamente le singole istruzioni dell'architettura ospitata.

I vantaggi dell'emulazione sono l'interoperabilità tra ambienti eterogenei, mentre gli svantaggi sono le ripercussioni sulle performances.

Esistono principalmente due tecniche di emulazione: **interpretazione** e **ri-compilazione dinamica**.

Interpretazione:

L'interpretazione si basa sulla lettura di ogni singola istruzione del codice macchina che deve essere eseguito e sulla esecuzione di più istruzioni sull'host virtualizzante. Produce un sovraccarico elevato in quanto potrebbero essere necessarie molte istruzioni dell'host per interpretare una singola istruzione sorgente.

Compilazione dinamica:

Invece di leggere una singola istruzione del sistema ospitato, legge interi blocchi di codice, vengono analizzati, tradotti per la nuova architettura, ottimizzati e messi in esecuzione. Il vantaggio in termini prestazionali rispetto all'interpretazione è notevolmente maggiore.

Ad esempio parti di codice utilizzati frequentemente vengono bufferizzati nella cache per evitare di doverli ricompilare in seguito.

.. ..

1.2.2 Realizzazione del VMM

Requisiti di Popek e Goldberg del 1974:

- **Ambiente di esecuzione per i programmi sostanzialmente identico a quello della macchina reale:** Gli stessi programmi che eseguono nel sistema non virtualizzato possono essere eseguiti nelle VM senza modifiche e problemi.

- **Garantire un'elevata efficienza nell'esecuzione dei programmi:** Il VMM deve permettere l'esecuzione diretta delle istruzioni impartite dalle macchine virtuali, quindi le istruzioni non privilegiate vengono eseguite direttamente in hardware senza coinvolgere il VMM
- **Garantire la stabilità e la sicurezza dell'intero sistema:** Il VMM deve sempre rimanere sempre nel pieno controllo delle risorse hardware, e i programmi in esecuzione nelle macchine virtuali non possono accedere all'hardware in modo privilegiato

Parametri e classificazione - **Livello** dove è collocato il VMM: - **VMM di sistema**: eseguono direttamente sopra l'hardware dell'elaboratore (vmware esx, xen, kvm) - **VMM ospitati**: eseguiti come applicazioni sopra un S.O. esistente (parallels, virtualbox) - **Modalità di dialogo**: per l'accesso alle risorse fisiche tra la macchina virtuale ed il VMM: - **Virtualizzazione pura** (vmware): le macchine virtuali usano la stessa interfaccia dell'architettura fisica - **Paravirtualizzazione** (xen): il VMM presenta un'interfaccia diversa da quella dell'architettura HW