

Cenni sulla virtualizzazione

Anno accademico 2017/18

Sistemi Concorrenti e Distribuiti

Tullio Vardanega

Virtualizzazione

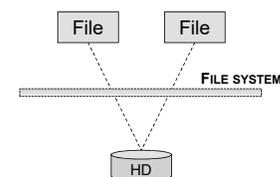
- Realizzare una vista logica su una risorsa indipendentemente dalla sua reale natura
 - La virtualizzazione rimpiazza il reale esponendo una interfaccia di astrazione
 - Esempio: il prerilascio virtualizza la modalità di esecuzione nell'architettura di von Neumann
- Parola chiave: **encapsulation**
- Punto di forza
 - La virtualizzazione usa l'astrazione ma ne rafforza il valore impegnandosi a garantire sempre la vista logica esposta all'utente

Astrazione

- Nascondere dettagli dell'implementazione per semplificare la vista logica offerta all'uso
 - Esporre un tipo di dato astratto (più semplice) invece della realtà sottostante (più complessa)
 - Esempio: in UNIX/Linux, l'astrazione base è il *file* che noi usiamo tramite una interfaccia ben definita
- Parole chiave: **information hiding, well-defined interface**
- Punto debole
 - L'interfaccia di astrazione è fragile rispetto a variazioni che intervengano sotto l'astrazione

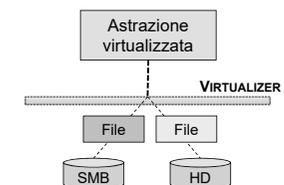
Esempio

Astrazione



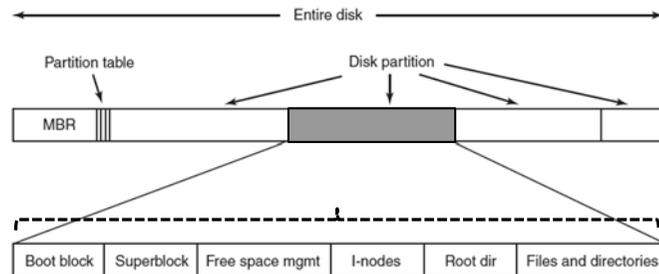
Permettere usi e contenuti diversi sulla stessa astrazione realizzandola nascondendo la struttura fisica sottostante

Virtualizzazione



Offrire una specifica astrazione "virtuale", possibilmente basata su diverse astrazioni di livello inferiore

Astrazione di Sistema Operativo



- **Boot block:** procedura di inizializzazione
- **Superblock:** descrittore del resto della partizione
- **I-nodes:** lista di tutti i descrittori (*i-node*)

Cenni storici /2

- Anni '80, passaggio ai *minicomputer* prima e ai PC poi
- Il problema della condivisione trasparente delle risorse di calcolo viene risolto in modo ricorrente («standardizzato») dai S/O multi-programmati
- Diminuisce l'interesse per lo sviluppo della virtualizzazione

Cenni storici /1

- Anni '60, epoca *mainframe*
- HW scarsamente disponibile e molto costoso
- La virtualizzazione permette la condivisione trasparente delle poche risorse fisiche disponibili
 - Il *time sharing* virtualizza l'accesso alla CPU
 - La memoria virtuale supera i limiti fisici
- La virtualizzazione diventa così uno dei principi fondanti dell'informatica

Cenni storici /3

- Primi anni '90: cresce l'attenzione per il calcolo a parallelismo massiccio per applicazioni specializzate
 - Esempio: Transputer (*transistor & computer*), componente *general-purpose* antesignano dei *massively parallel processors*
- L'interesse per la virtualizzazione rinasce per agevolare la preservazione di applicazioni destinate ad HW *special-purpose*
 - Nasce VMware Inc.

Cenni storici /4

- Seconda metà anni '90, enorme diffusione dell'IT a supporto delle attività aziendali
 - Al diminuire del costo unitario aumenta l'eterogeneità (classica legge della domanda)
 - Con server dedicati si ha maggior costo di gestione, meno portabilità e sotto-utilizzo di HW e SW eterogenei
- Rinascita della virtualizzazione
 - Condividere HW e risorse di calcolo inutilizzate aiuta a ridurre i costi

Laurea Magistrale in Informatica, Università di Padova 9/27

Dove interviene l'ABI: esempio

Laurea Magistrale in Informatica, Università di Padova 11/27

Architettura e interfacce /1

Laurea Magistrale in Informatica, Università di Padova 10/27

Architettura e interfacce /2

- Tre punti di connessione per servizi a valore aggiunto
 - API, ABI, ISA
- Le interfacce di astrazione sono alla base dell'architettura classica dei sistemi di calcolo
- Ma ogni astrazione è fragile rispetto a variazioni nella natura e nel comportamento del livello sottostante

Laurea Magistrale in Informatica, Università di Padova 12/27

Architettura e interfacce /3

- Cosa succede se cambia l'HW?
 - Se cambia l'ISA sotto il S/O sono costretto a cambiare S/O per la fragilità di quella astrazione
 - Potrebbe diventare necessario cambiare anche ABI e API
- Per preservare il valore aggiunto dei livelli alti dobbiamo rafforzare l'astrazione con la virtualizzazione
 - Ma dobbiamo scegliere a che livello realizzarla

Laurea Magistrale in Informatica, Università di Padova 13/27

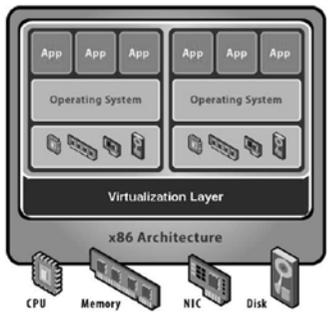
Principio base di virtualizzazione /1

- Dalla fine degli anni '60 il «modo» di esecuzione è diviso in livelli di privilegio
 - L'ISA è accessibile al SW in sottoinsiemi («**protection ring**») concentrici più vicini al core al crescere del privilegio
 - Ogni tentativo di accesso a istruzioni HW a livello di privilegio superiore di quello del chiamante solleva una eccezione (**trap** HW)
 - L'innalzamento di privilegio è ottenuto tramite una istruzione speciale (**trap** SW)

Laurea Magistrale in Informatica, Università di Padova 15/27

Punto di arrivo

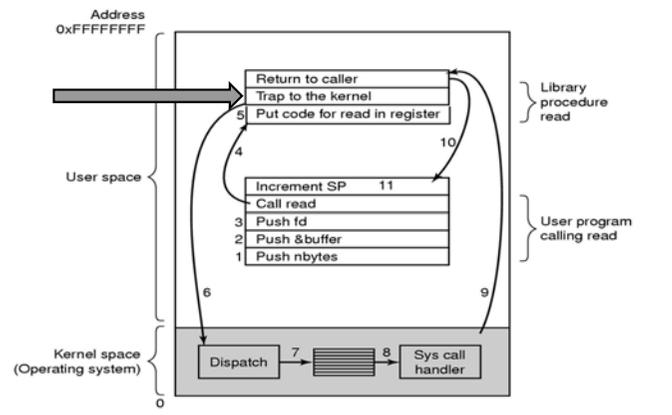
L'attenzione si sposta sull'isolamento



- Hardware-Level Process Abstraction: CPU, memory, chipset, I/O devices, etc.
 - Virtual NIC instead of sockets
 - Virtual disk instead of file system
 - Hardware state becomes software state
- Virtualization Software
 - Hardware and software decoupled

Laurea Magistrale in Informatica, Università di Padova 14/27

Principio base di virtualizzazione /2



Address 0xFFFFFFFF

User space

Kernel space (Operating system)

Laurea Magistrale in Informatica, Università di Padova 16/27

Virtualizzazione: obiettivi

Virtualization allows the creation of a secure, customizable, and isolated execution environment for running applications without affecting other users' applications

- various functions enabled by managed execution
 - sharing (e.g. server consolidation)
 - aggregation (e.g. cluster management software)
 - emulation (e.g. arcade-game emulator)
 - isolation ⇒ no interference between multiple guest

Virtual Resources
Virtualization
Physical Resources

Sharing, Aggregation, Emulation, Isolation

Laura Magistrale in Informatica, Università di Padova 17/27

Virtualizzazione: tassonomia /2

How it is done? Technique Virtualization Model

Execution Environment, Storage, Network, ...

Process Level: Emulation, High-Level VM, Multiprogramming

System Level: Hardware-Assisted Virtualization, Full Virtualization, Paravirtualization, Partial Virtualization

Application, Programming Language, Operating System, Hardware

Laura Magistrale in Informatica, Università di Padova 19/27

Virtualizzazione: tassonomia /1

■ In base al livello di interfaccia sotto al quale si realizza la virtualizzazione

Virtualization

Process, System

Classic-System, Hosted

Paravirtualization, Whole-System

Laura Magistrale in Informatica, Università di Padova 18/27

Process Virtualization /1

- L'*hypervisor* fornisce una specifica ABI per le applicazioni
- L'unione tra l'*hypervisor* e i programmi su di esso eseguiti viene detto Virtual Machine (VM)
- La VM a livello di processo più comune è il S/O stesso!

Application SW, Libraries

Process Hypervisor

O/S

API, ABI

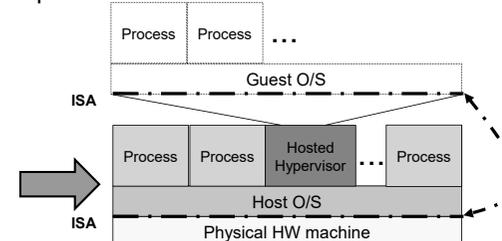
Laura Magistrale in Informatica, Università di Padova 20/27

Process Virtualization /2

- La *process VM* dispone di
 - Memoria virtuale
 - Strumenti di I/O astratti come *file* e *socket*
 - Tempo di CPU
- Una «capsula SW» esegue l' applicazione ospite istruzione per istruzione
 - Per esempio tramite «*instruction interpretation and translation*» → JVM (*bytecode*)
 - Oppure per esecuzione diretta → Wine (binario)

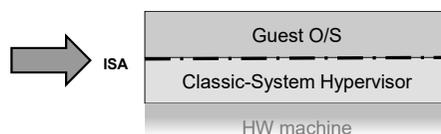
Hosted Virtualization

- L'*hypervisor* è un processo come tutti gli altri
 - Alloca le risorse di memoria e *storage* necessarie richiedendole al S/O ospitante
 - Medesimo ISA (esecuzione diretta), diverso ISA (interpretazione)
- Comporta alto costo di esecuzione



Classic-System Virtualization

- Nelle VM di tipo sistema la «macchina» esposta è una ISA con periferiche associate
 - *Storage, network, etc...*
- Questa virtualizzazione riproduce tutto quello che serve al S/O ospite esattamente come sarebbe l'HW fisico
 - «**Guest OS de-privileging**»



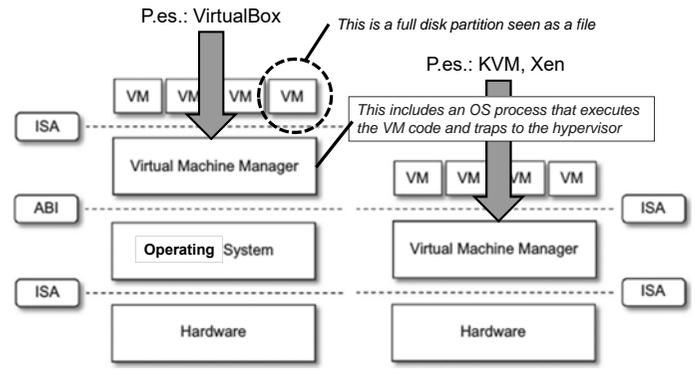
Whole-System Virtualization

- Questa tecnica permette di virtualizzare architetture (ISA) diverse da quella ospitante
- Variante del tipo «*Hosted*»
 - Nel caso «*hosted*» le istruzioni HW a basso privilegio emesse dall'applicazione virtualizzata eseguono direttamente
 - Perché l'ISA virtualizzata è la stessa di quella fisica
 - Nel caso «*whole system*» serve un emulatore di ISA all'interno dell'*hypervisor*

Para-virtualization /1

- Negli anni '80 scema l'interesse per la *system virtualization*
- Nelle architetture x86 vengono introdotte istruzioni macchina non virtualizzabili (!)
 - La loro esecuzione non genera *trap* HW
 - Conseguentemente l'*hypervisor* non si può accorgere del loro utilizzo

Visione d'insieme



Para-virtualization /2

- Viene allora definita una nuova interfaccia (*hypercall API*) che richiede l'adattamento del *S/O guest*
- Il beneficio è una bassa penalità di esecuzione (ca. 1%)

