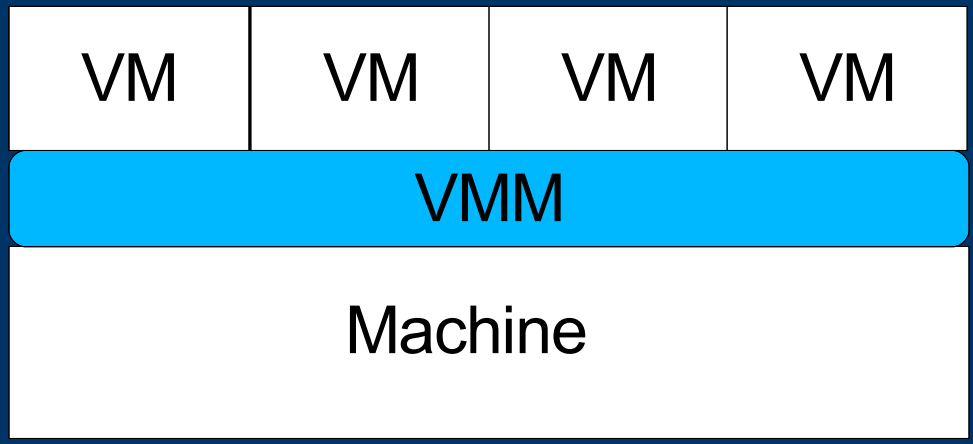
Tecniche di virtualizzazione di processori Intel-AMD

Gianluca Guida
glguida@gmail.com

Cos'e' la virtualizzazione.

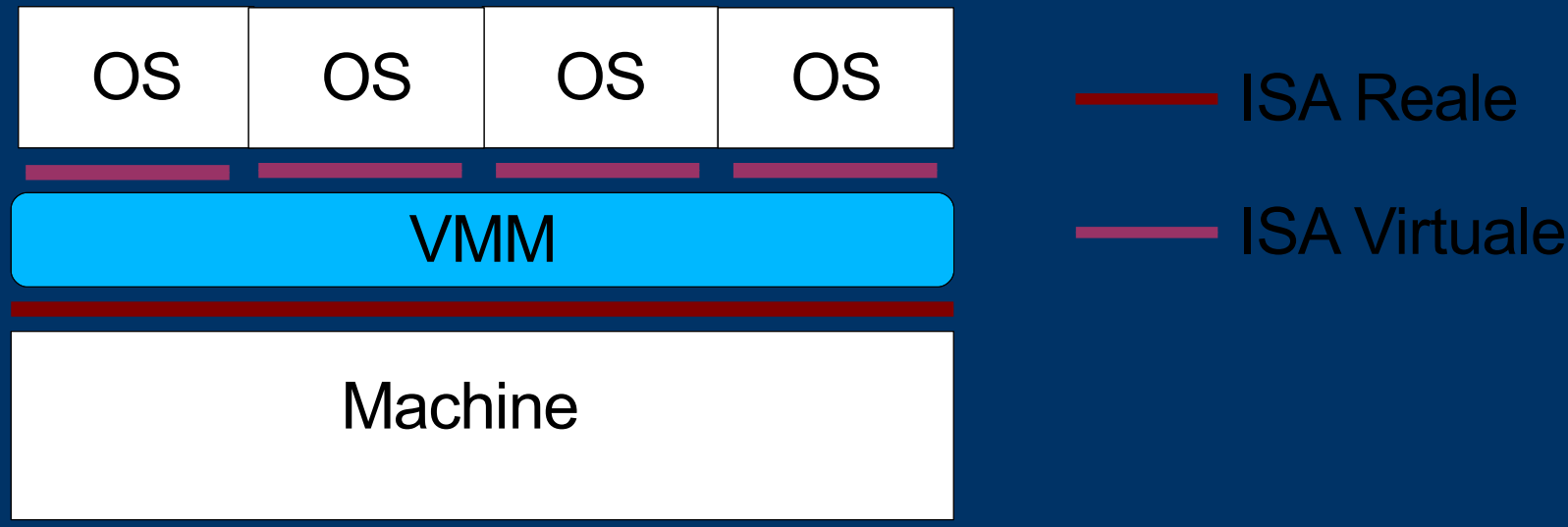
- “*efficient, isolated duplicate* of the real machine”;



Da ISA reale a ISA virtuale

ABI, actually.

- Instruction Set Architecture



- Es ISA: IA32, AMD64, MIPS;
- In questo talk:
  - ISA reale == ISA virtuale

Virtual Machine Monitor 1/2

Proprieta' di un VMM (*Popek–Goldberg, 1974*)

- *Fidelity.* “Ambiente” di esecuzione dei processi sostanzialmente identico alla macchina;
- *Performance.* Overhead della virtualizzazione realativamente basso (prestazione VM comparabile a prestazione macchina fisica)
- *Safety.* VMM controlla e gestisce le risorse fisiche;

Virtual Machine Monitor 2/2

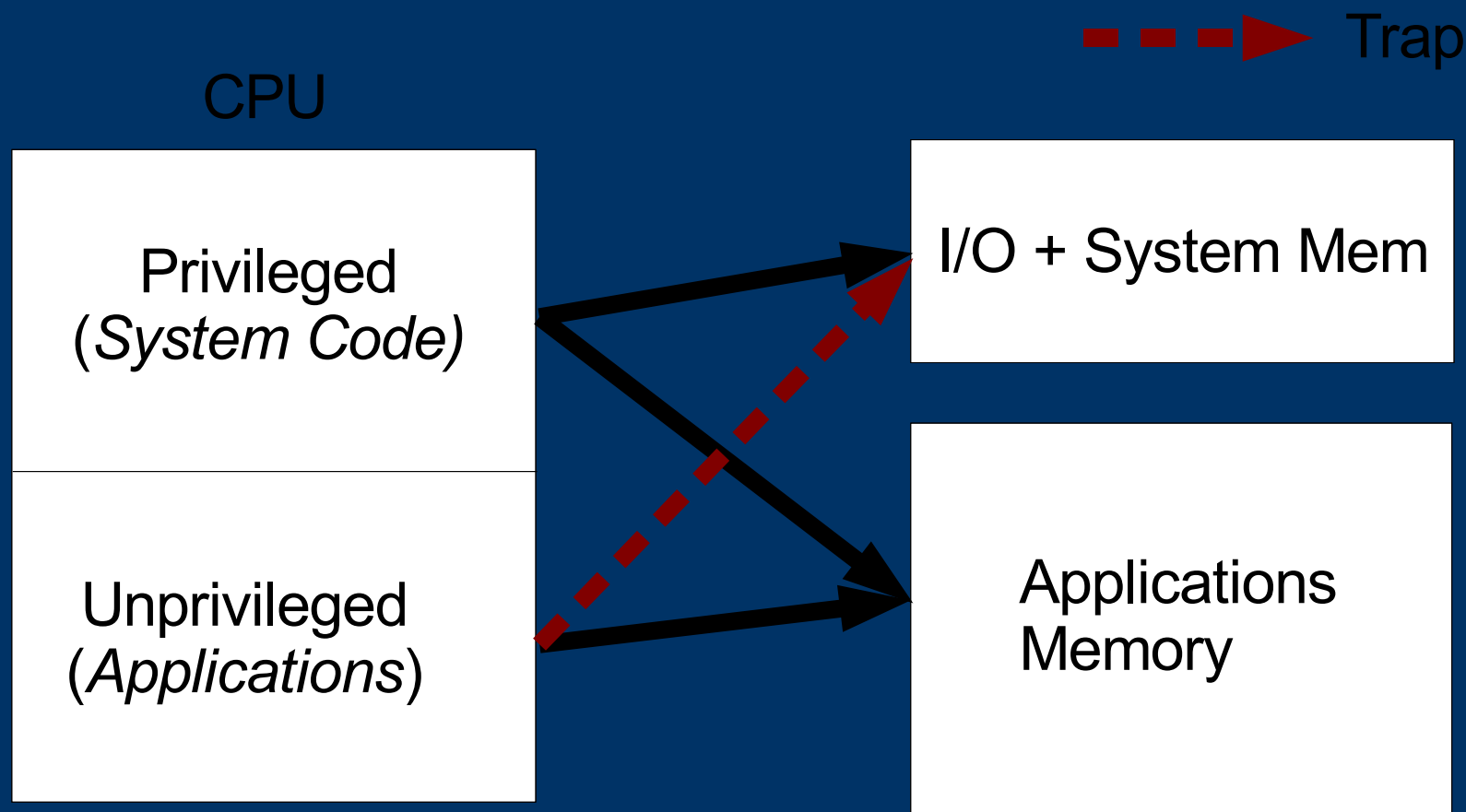
Architettura concettuale di un VMM:

- CPU virtualization (ISA)
- I/O virtualization (virtual devices)
- Gestione risorse fisiche

In questo talk:

CPU virtualization nell'architettura IA32/AMD64 (etc...)

Modello Ideale di CPU 1/2



Modello Ideale di CPU 2/2

Istruzioni interessanti ai fini della virtualizzabilita':

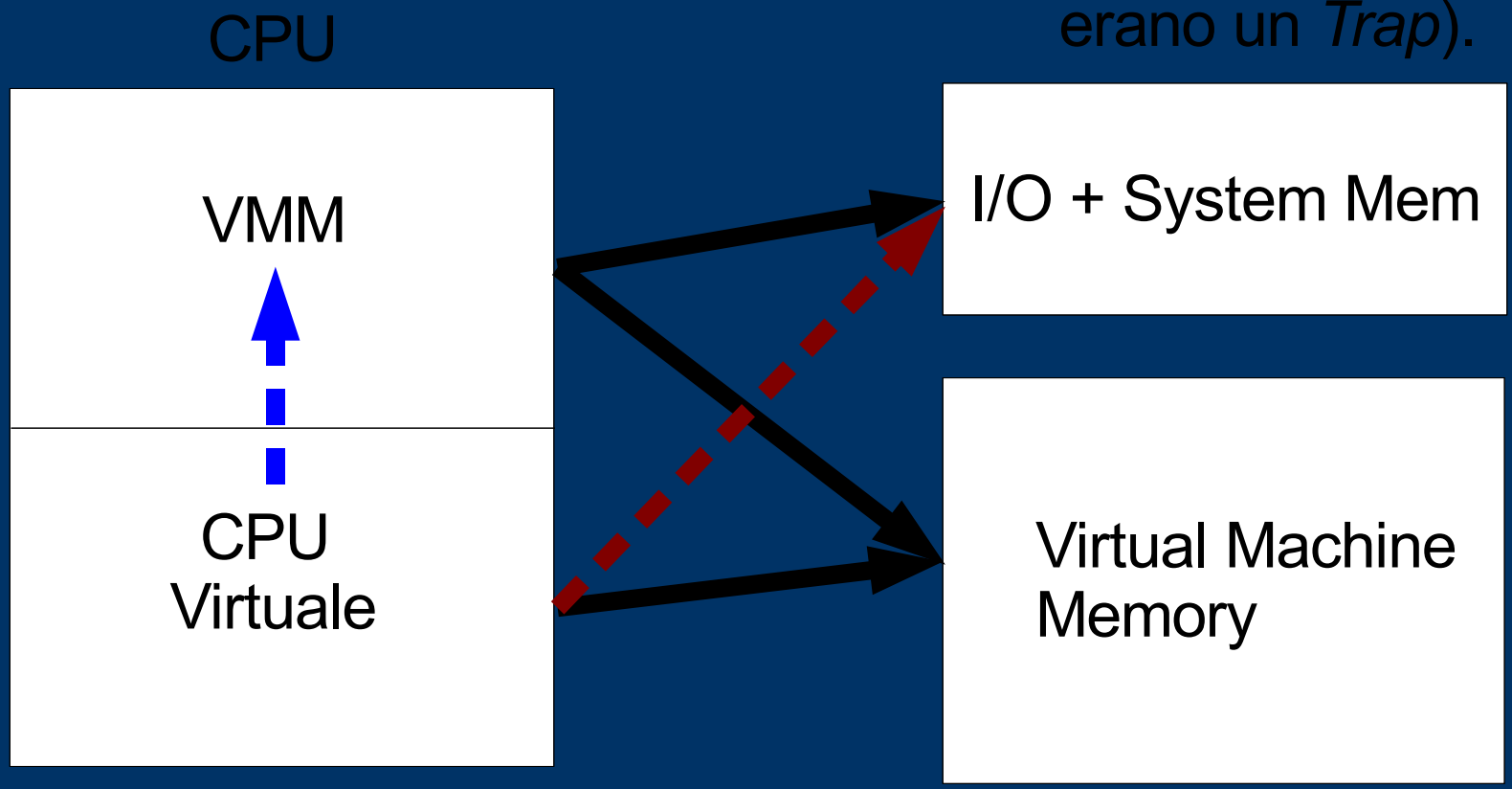
- *Sensitive Instructions*: eseguite in Privileged Mode cambiano lo stato del sistema.
- *Privileged Instructions*: possono essere eseguite solo in Privileged mode; in Unprivileged mode causano *traps*.

La classificazione dell'ISA in questo tipo di istruzioni e' ortogonale. (ci possono istruzioni *sensitive* non *privileged* e viceversa)

Tecnica classica di virtualizzazione

Trap and Emulate 1/2

VMM emula istruzioni privilegiate (che generano un *Trap*).



Tecnica classica di virtualizzazione
Trap and Emulate 2/2

- Il VMM viene eseguito in *Privileged Mode*;
- Le istruzioni delle CPU virtuali (sia in *Privileged* che in *Unprivileged Mode*) vengono eseguite sulla CPU fisica in *Unprivileged Mode*.

- Quando la CPU virtuale prova ad eseguire un'istruzione privilegiata viene generato il fault (*Trap)*.
- Il Trap viene gestito dal VMM che emula gli effetti dell'istruzione privilegiata sulla CPU virtuale (*Emulate*).

Tecnica classica di virtualizzazione

Condizioni per la virtualizzabilita' in senso classico 1/2

- Se esistono istruzioni *sensitive* ma non *privileged* allora la VCPU non generera' un fault.
  Succede quando la stessa istruzione ha comportamenti diversi a seconda che sia eseguita in modalita' *Privileged* o in modalita' *Unprivileged*.
- Tali architetture non sono *virtualizzabili in senso classico*.

Tecnica classica di virtualizzazione

Condizioni per la virtualizzabilita' in senso classico 2/2

Formalmente si ha che un sistema e' virtualizzabile in senso classico quando (teorema di *Popek-Goldberg*):

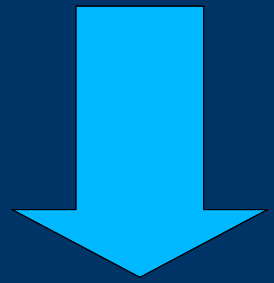
$$I_S \subseteq I_P$$

I_S = Insieme delle *sensitive instructions*.

I_P = Insieme delle *privileged instructions*.

x86 e virtualizzazione

- Solo un esempio: *the POPF problem*. (*sensitive* non *privileged*)

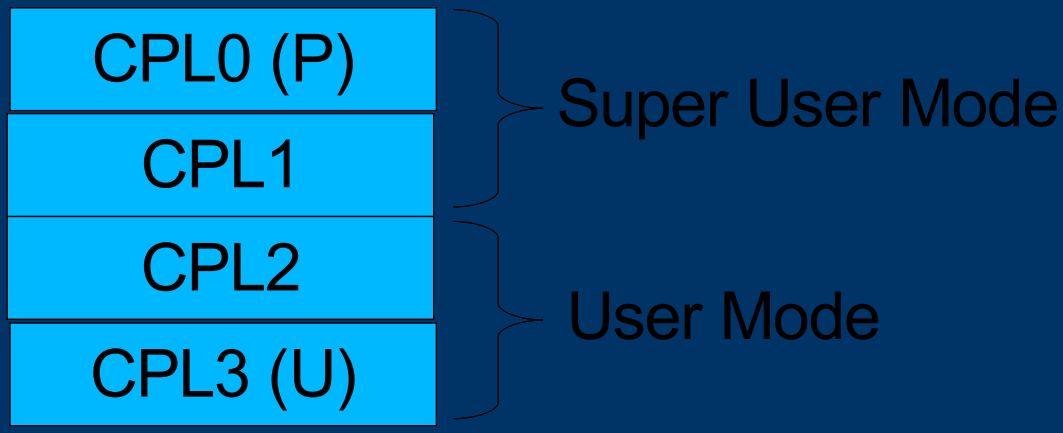


- Non virtualizzabile in senso classico

x86 e virtualizzazione
semplificando...

- Architettura incredibilmente complicata.

Livelli di privilegio



- Livello di privilegio per l'accesso all'I/O variabile (per *task*)
- Inoltre ci sono altre modalita' di funzionamento che non hanno alcun tipo di livelli di privilegio.

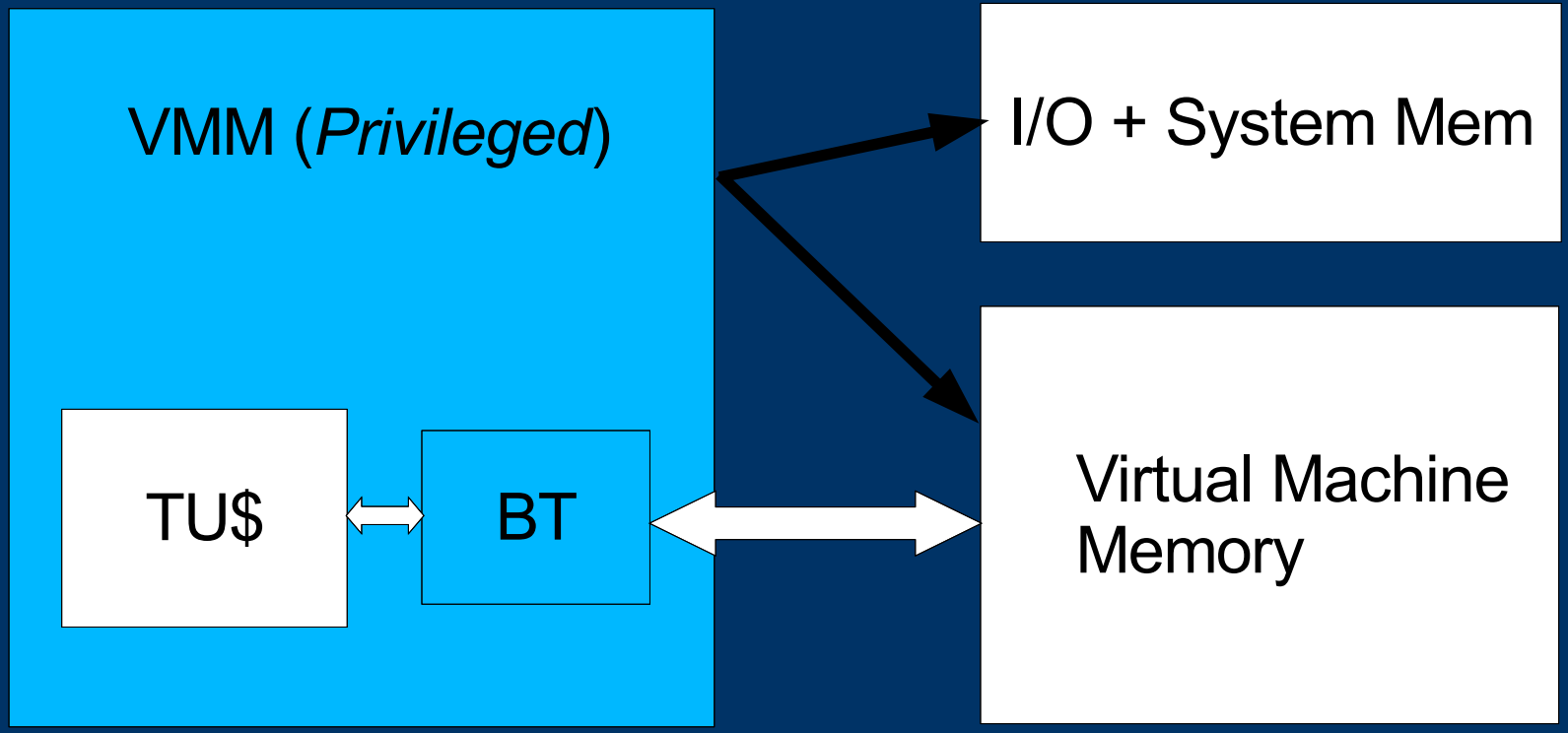
Software Virtualization (BT)

mai saltare l'ultimo paragrafo.

La virtualizzazione dell'x86 e' ancora possibile, ma richiede una traduzione binaria dinamica (Dynamic BT) del codice eseguito.

- *Dinamica*: il codice puo' essere automodificante, quindi la traduzione dev'essere Just In Time.
- *Binaria*: l'Input del traduttore e' codice binario x86, non codice sorgente
- L'output del traduttore e' ancora codice x86.

Software Virtualization (BT)



Software Virtualization (BT)

Come fatturare $180mln a trimestre.

- Solo le istruzioni *privileged* e *sensitive* richiedono il BT, quindi la maggior parte delle traduzioni sono di tipo 1:1 con l'input;
- Cache delle parti gia' tradotte (invalidate quando il codice viene modificato);

Supporto Hardware

ingegneri elettronici...

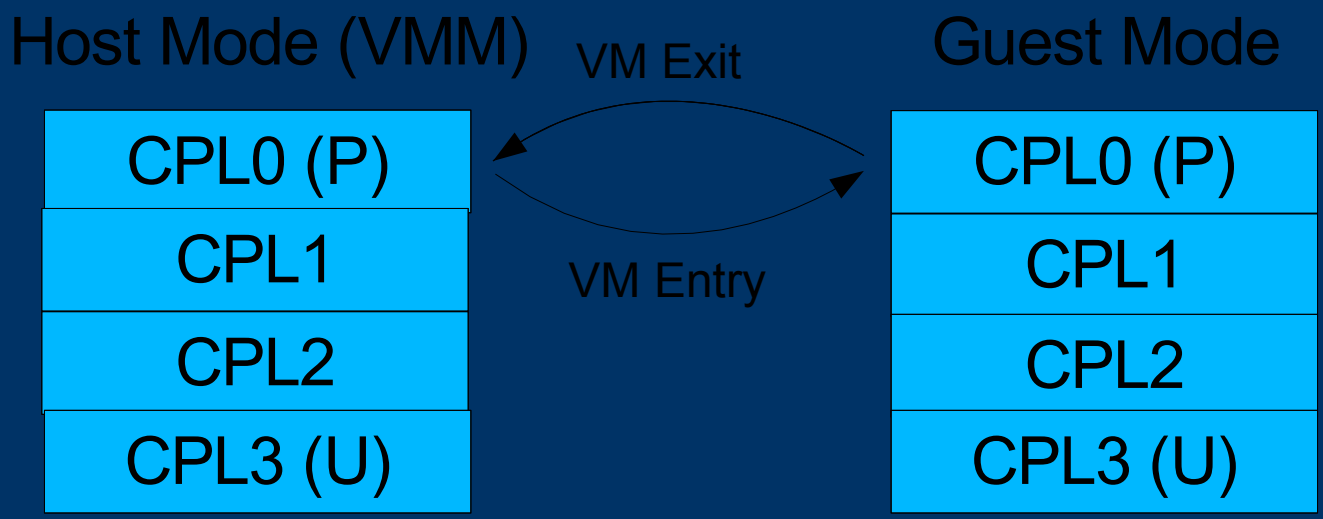
- Ritorno della richiesta per virtualizzazione.

- Ottima opportunita' per complicare ulteriormente l'x86 (*vende meglio*).
  - Intel: VT (Virtualization Technology)
  - AMD: SVM (Secure Virtual Machine)

  Soluzioni sostanzialmente uguali.

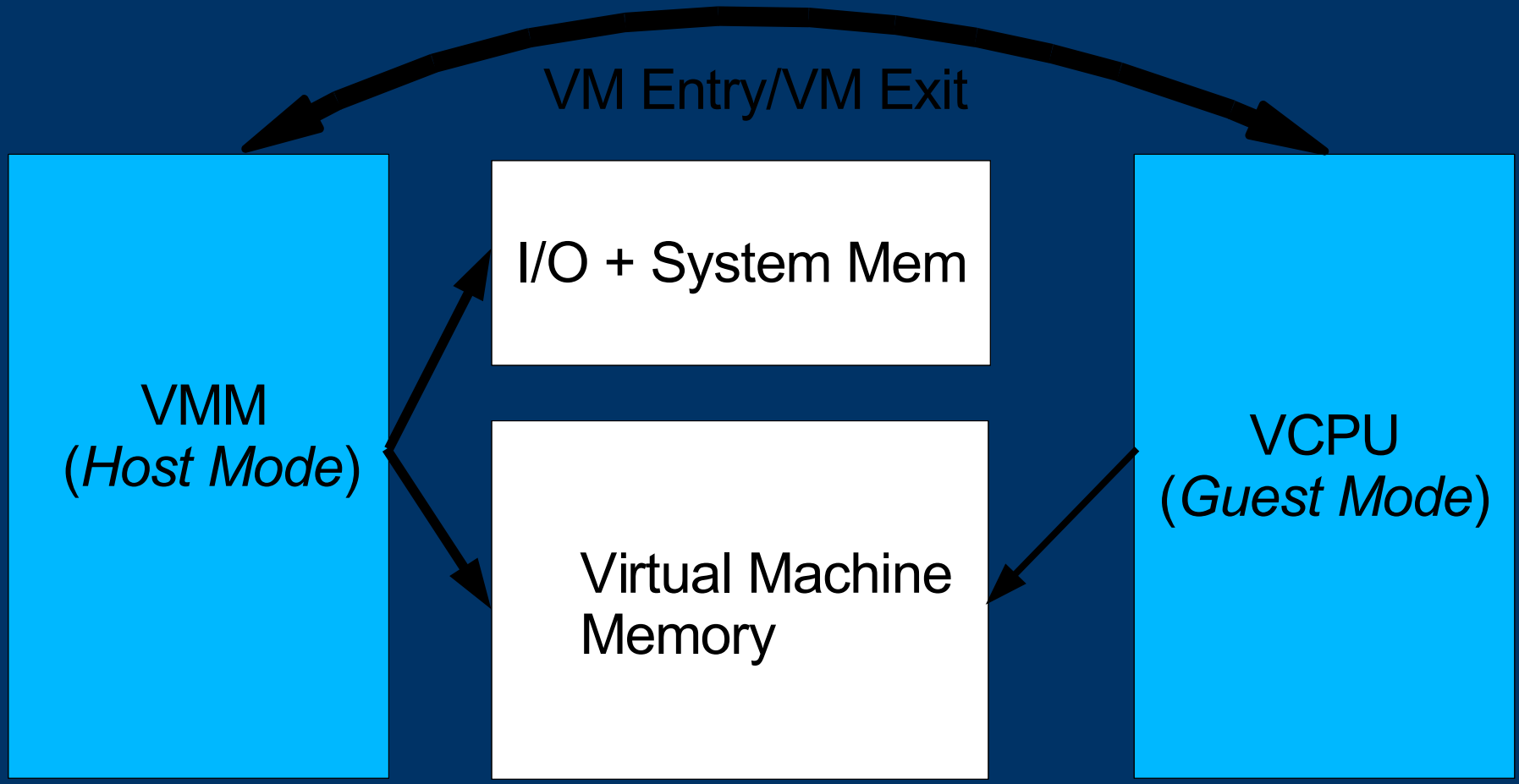
Supporto Hardware

Solo una dimensione di livelli di privilegio e' lame.



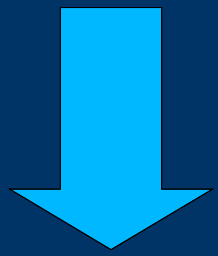
- L'Host Mode corrisponde al normale funzionamento (usato dal VMM).
- Il Guest Mode e' identico all'Host mode, ma puo' essere configurato per generare VM Exits a determinate condizioni.
- Lo stato di un processore virtuale e' salvato in un'area di memoria (VMCS/Intel o VMCB/AMD)

Virtualizzazione con supporto HW



Virtualizzazione con supporto HW

- Non c'e' *de-privileging*;
- La stragrande maggiorparte del codice e' eseguito in modalita' nativa;
- Non tutte le istruzioni *sensitive* in *privileged mode* generano VM Exits (Ex: POPF cambia gEFLAGS);



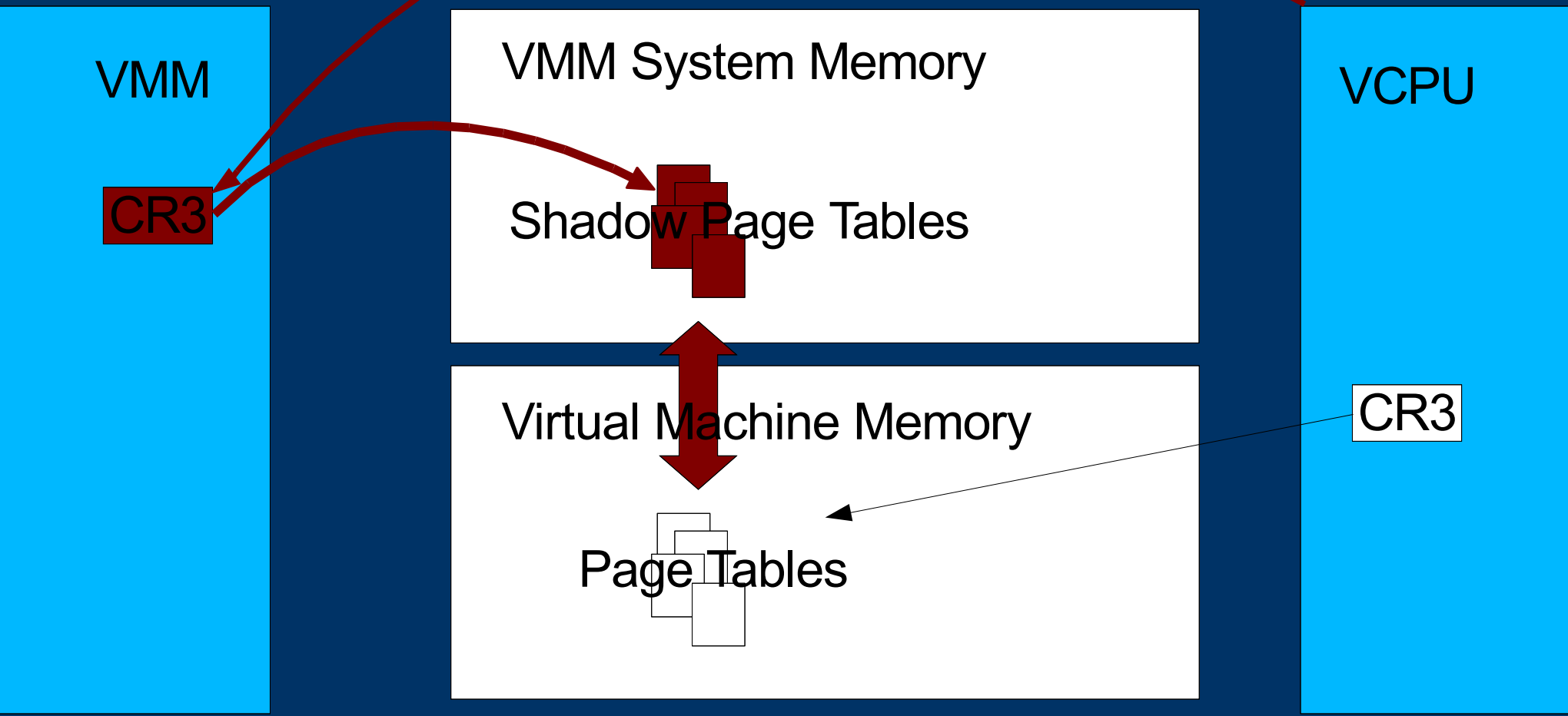
- Un VMM del genere *dovrebbe* avere prestazioni molto superiori ad uno che usa il BT;

L'altra “VM”
Virtual Memory

- Cos'e' la Virtual Memory;
- Lo stato della memoria virtuale non e' nella CPU ma in memoria (*Page Tables*);
- Il VMM deve proteggere le tabelle;
- Inoltre le Page Tables che crea la Virtual Machine non corrispondono alle Page Table reali.

L'altra “VM”

shadow structures.



L'altra “VM”

shadow structures explained.

- La traduzione tra indirizzo virtuale della macchina virtuale in indirizzo fisico avviene nelle Page Tables del VMM (*shadow page tables*);
- Il VMM mantiene le shadow page tables consistenti con le page tables della macchina virtuale;

- Uno dei motivi principali dell'overhead di virtualizzazione e' appunto dovuto ai continui interventi del VMM per la gestione della memoria virtuale.

BT vs HW

Ingegneri elettronici, convertitevi.

- Un VMM che sfrutta il supporto HW, come gia' visto, risolve la necessita di generare trap per eseguire istruzioni privilegiate (o di interpretarle dinamicamente);
- Non aiuta, pero', nei trap generati dalla Virtual Memory.

BT vs HW

inductance is futile.

- Costo del Trap (VCPU => VMM) e' molto maggiore nei VMM con supporto HW rispetto ai VMM software;
- Software e' piu' flessibile. Piu' ottimizzazioni possibili;

Un VMM basato su software tende ad essere migliore in applicazioni I/O-bound o che creano molti processi.

E l'x86-64?

- Le versioni di AMD (AMD64) e Intel (EM64T) differiscono lievemente nella gestione dei segmenti.
  - AMD64 supporta il Limit Checking dei segmenti, non EM64T
- Il BT usa il Limit Checking per proteggere la memoria del VMM, quindi su processori Intel la virtualizzazione a 64 bit richiede il supporto hardware.

E la paravirtualizzazione?

- Il sistema operativo viene modificato e reso cosciente della presenza del VMM.
- Il sistema operativo comunica direttamente con il VMM, semplificandone notevolmente la complessita' e aumentando le prestazioni del sistema in maniera significativa.
- Idea non nuova.
- Due implementazioni al momento: Xen (open source, alleanza con Microsoft) e VMware VMI.

Grazie!