La virtualisation

- Melchior Nicolas

http://publicdomainvectors.org/ko/%EB%AC%B4%EB%A3%8C%20%ED%81%B4%EB%A6%BD%EC%95%84%ED%8A%B8/%ED%83%B1%EA%B3%A0-%EC%BB%B4%ED%93%A8%ED%84%B0-%EC%95%84%EC%9D%B4%EC%BD%98-%EB%B2%A1%ED%84%B0-%EC%9D%B4%EB%AF%B8%EC%A7%80/12959.html

- Références
  - White Parer : Understanding Full Virtualization, Paravirtualization, and Hardware Assist, 2007, VmWare
  - Software and Hardware Techniques for x86 Virtualization, 2009, VmWare
  - Performance Aspects of x86 Virtualization, Mar 30, 2009 ,Ole Agesen http://www.vmware.com/pdf/Perf_ESX_Intel-EPT-eval.pdf
  - Ars Technica Guide to Virtualization, 2008-2010, Jon Stokes
    - http://arstechnica.com/hardware/news/2008/08/virtualization-guide-1.ars/
    - http://arstechnica.com/hardware/news/2008/12/virtualization-guide-2.ars/2
  - The Ars Technica Guide to I/O Virtualization, 2010, Peter Bright
    - http://arstechnica.com/business/guides/2010/02/io-virtualization.ars/3

Introduction

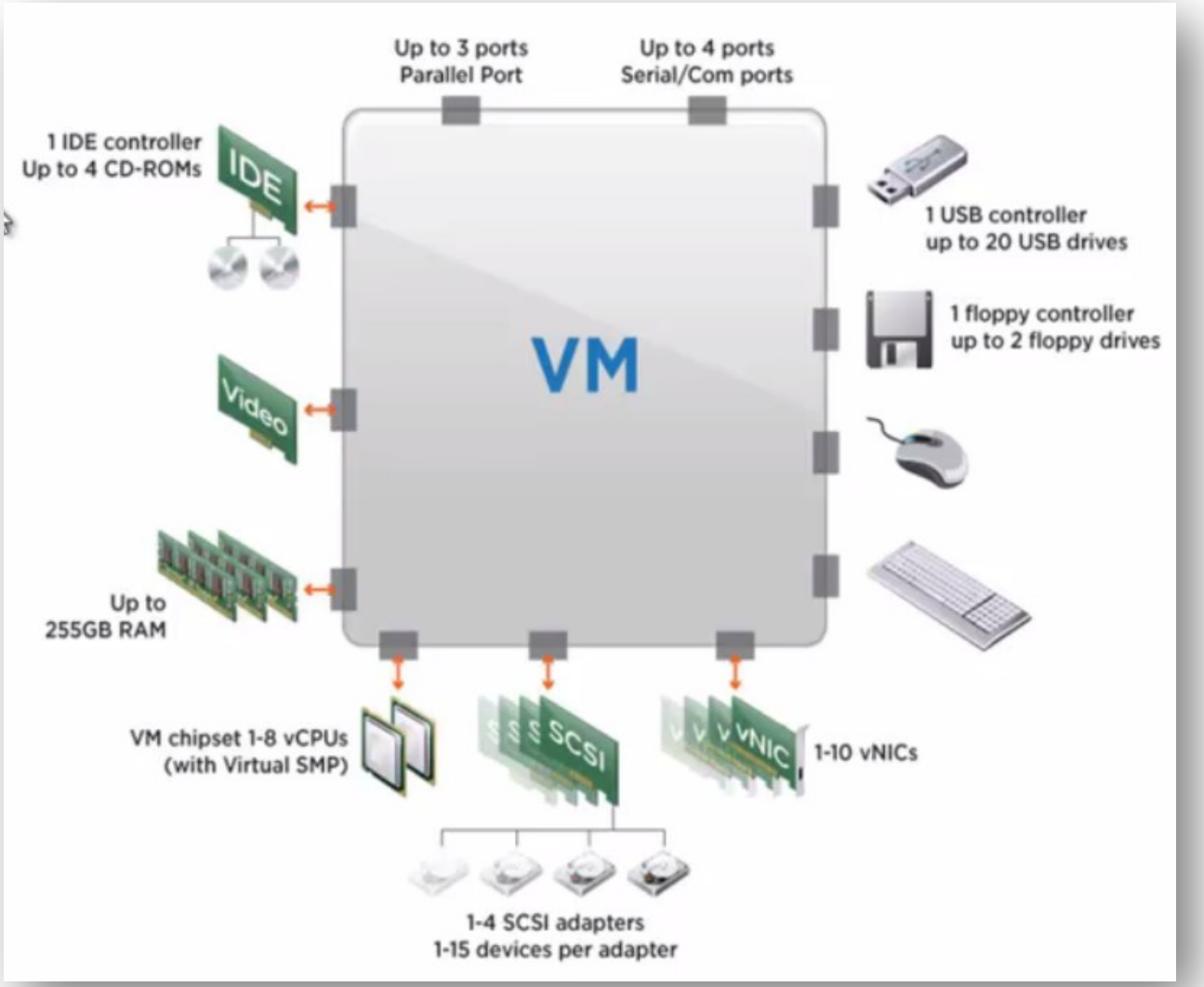
- La virtualisation est un ensemble de technologies hardware et software permettant de virtualiser du matériel (Ordinateur, stockage, etc.) et du logiciel (OS, applications, etc).
- Grands types de virtualisation :
  - Hardware
  - SDS : Software Defined Storage
  - SDN : Sofware Defined Network
  - SDDC : Software Defined Data Center : combinaison des 3
- Tendance en informatique à s'éloigner du matériel
  - Des techniques permettent de réutiliser le code binaire de la génération hardware précédente ce qui amène à une diminution des coûts
    - ISA + microcode
    - RISC
      - Compilateur entre ISA et programmeur
      - Augmentation performances
- Objectifs : réduction des coûts en optimisant l'emploi du matériel
- 2 familles de technologies :
  - Virtualisation de systèmes d'exploitations
  - Virtualisation d'applications
- La virtualisation de systèmes d'exploitations est un ensemble de techniques matérielles et logicielles qui permettent de faire fonctionner plusieurs systèmes d'exploitations sur un même machine.

Virtualisation d'applications

- Isolation
- Container

Une VM, c’est quoi ?

- Un ordinateur logiciel :
  - Simulation de ressources matériels
    - GPU
    - RAM
    - CPU’s
    - SCSI
    - IDE
    - SATA
    - NIC
    - USB
    - COM
    - …
  - + un OS



Virtualisation de systèmes d'exploitations

https://pixabay.com/en/fantasy-dream-reality-virtual-639115/

Objectifs

- Avantages :
  - Consolidation
    - Meilleure utilisation de CPU (un serveur utilise typiquement 5-10%)
    - Économie de surface et d’électricité en datacenter ($$ + environnement)
  - Load balancing
    - Une batterie de serveurs peut équilibrer la charge en déplaçant une machine virtuelle d'un système sur-utilisé à un système sous-utilisé
  - Tolérance de panne
    - Si le matériel d'un serveur se dégrade, les machines virtuelles de ce serveur peuvent être transférées (en direct) vers d'autres serveurs. Le serveur d'origine peut ensuite être arrêté pour maintenance, sans interruption de service.
  - Isolation
    - Les OS multi-utilisateurs ne suffisent pas toujours à isoler les utilisateurs les uns des autres
      - Utilisateur demandant beaucoup de ressources
      - Virus
    - Un administrateur peut facilement isoler une VM pour des raisons de performances et/ou de sécurité.

Objectifs

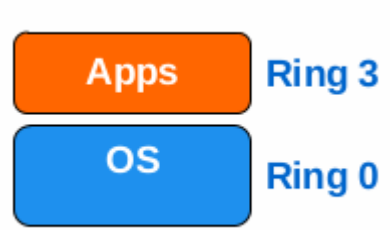
- Avantages :
  - Déboguer des programmes
    - Si un bug bloque complètement l'OS et nécessite à chaque fois un redémarrage pour pouvoir déboguer → plus rapide de redémarrer une VM.
  - Développement logiciel (ex : android)
    - Les développeurs utilisent également la virtualisation pour écrire des programmes pour un OS ou un ISA bien spécifique et ce sur n'importe quel OS hôte.
  - Compatibilité avec d'anciens systèmes (ex : win95/XP)
  - Virtualisation hétérogène
    - Linux, Windows, AX, MAC, Android,…
  - Administration simplifiée
    - Fini les problèmes hardware
  - HA, Migration, Clonage, affectation des ressources à chaud, Création de modèle…

Objectifs

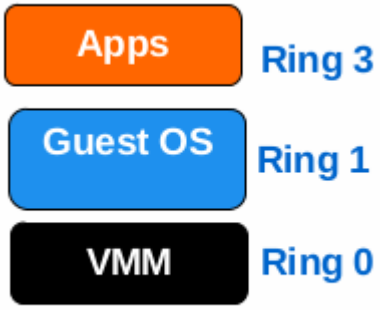
- Contraintes :
  - Pas de Self Service (Gestion par un admin)
  - Surcharge de travail CPU
    - logiciel « intermédiaire » en plus
    - Un kernel /VM
    - Solution : containers

Modèle de la Pile hard/Soft habituel

- Système d'exploitation au dessus du matériel
- Logiciels au dessus de l'OS
- L'OS
  - a un accès exclusif avec tous les privilèges au matériel
  - donne un accès sélectif aux ressources matérielles



- Rôles de l'OS :
  - Isoler les processus
  - Gérer le partage des ressources entre les processus
- → OS doit avoir accès complet au matériel
- Virtualisation : Faire croire que l'OS a cet accès

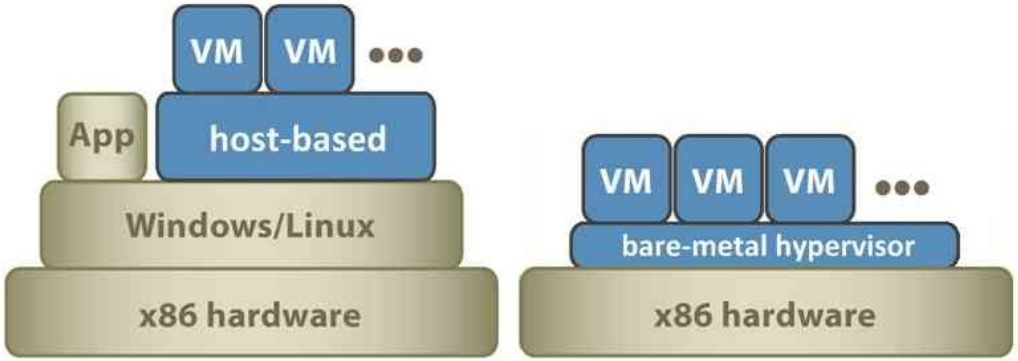


Notions

- OS Hôte
- OS Invité (Guest)
- VMM : Moniteur de machine virtuelles
  - Logiciel qui fait croire aux OS invités qu'ils tournent directement sur le matériel avec tous les privilèges sans couche logiciel intermédiaire
    - en exécutant un VMM au-dessus d'un système d'exploitation hôte qui hébergera plusieurs machines virtuelles
    - en installant un VMM entre le matériel et les OS invités, dans ce cas, le VMM est appelé hyperviseur
- ISA x86
  - Pas prévue pour → compliqué
  - Ajout des extensions à l'ISA pour la virtualisation (~ 2007)
    - Intel Vanderpool : VT-x
    - AMD : AMD-V
  - Amélioré à chaque nouvelle génération de CPU par de nouvelles extensions
- Machine virtuelle
  - Image créée par un VMM d'un ordinateur "idéal" présentée à un OS

Méthode de virtualisation

- 1. Hyperviseur
  - Intermédiaire direct entre le matériel et les OS Virtualisés
  - Tourne en espace kernel
- 2. Modèle hôte invité
  - VMM exécuté en espace utilisateur → VM dans l'espace utilisateur
    - Plus lent
    - Plus facile à mettre en place
- La VMM doit donner l'illusion à l'OS invité qu'il a l'accès exclusif à
  - CPU
  - Mémoire
  - Stockage
  - I/O



http://www.virtualisation-news.com/wp-content/images/virtualisation_arch_fig1.jpg

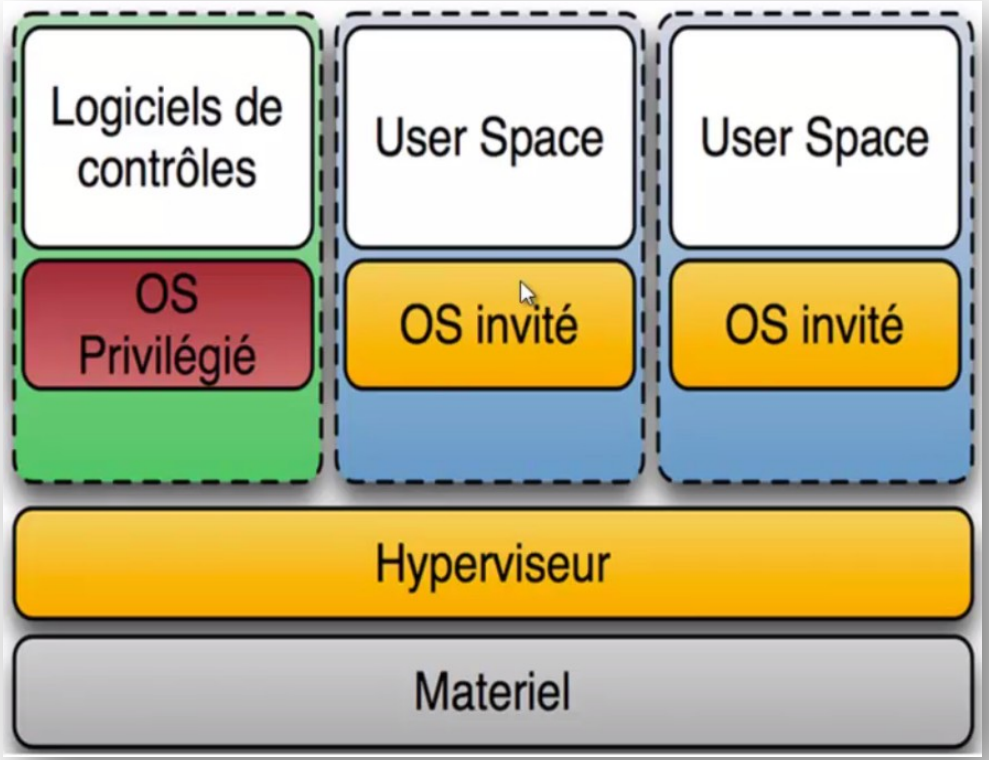
Technologies de Virtualisation

- Supportées au moins en partie par la majorités des VMM's actuels
- Matérielles (assistée par le « hardware »)
  - le matériel fournit un support architectural qui facilite la construction d'un VMM et permet aux systèmes d'exploitation invités de fonctionner de manière isolée
    - CPU
      - VT-x / AMD-V
    - Mémoire (MMU)
      - Nested paging : Intel EPT et AMD RVI
    - Mémoire de masse (HD)
    - I/O (carte réseau etc.)
      - Chipset I/O MMU intel VT-d / AMD-Vi
    - Périphériques
      - GPU : intel GVT (Graphics Virtualization Technology)

Technologies de Virtualisation

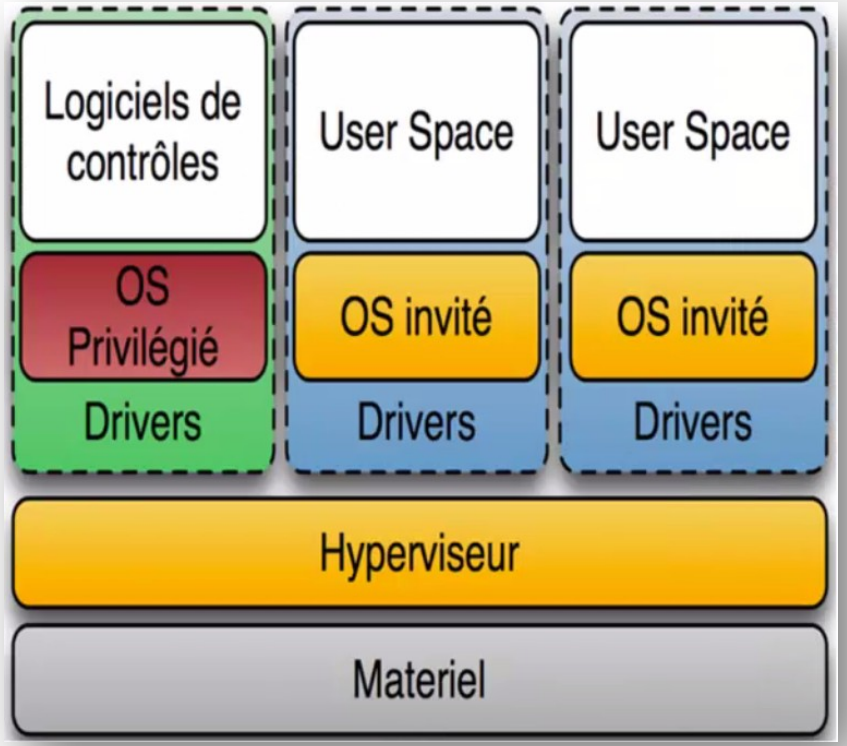
Logicielles

- Full Virtualisation
  - L'hyperviseur crée un environnement virtuel complet en simulant du « faux » matériel. L'OS invité n'a accès qu'à ces ressources simulées, et non aux ressources matérielles réelles. Ce type de virtualisation est limité aux systèmes d'exploitation prévus pour la même architecture matérielle (x86, x64, ARM, ...) que le processeur physique de la machine hôte.
- Émulation
  - Pour dépasser les limites ci-dessus, l'hyperviseur créé un environnement virtuel complet. Il simule un microprocesseur qui peut avoir une architecture matérielle différente de celle du CPU hôte. Le principal inconvénient est le niveau de performances, souvent assez bas. (Vmware Server, Virtualbox, Microsoft Virtual PC, ...)



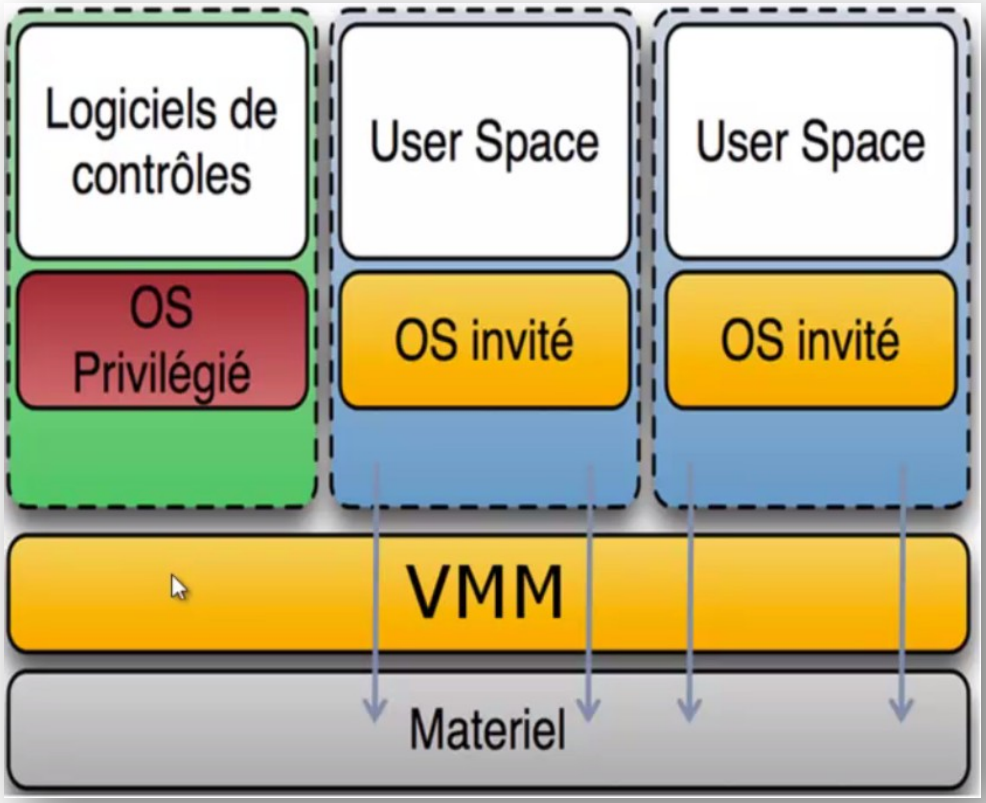
Technologies de Virtualisation

- Paravirtualisation (PVM)
  - Le système d’exploitation invité est conscient de s’exécuter dans un environnement virtualisé → modifications logicielles (par exemple l'installation de pilotes ou d'une surcouche logicielle). En contrepartie, il devient capable d’interagir avec l’hyperviseur et de lui demander de transmettre directement les appels systèmes au matériel du serveur hôte. Les performances « virtuelles » sont alors théoriquement proches de celles qu’il serait possible d’atteindre avec le matériel réel. (Xen, KVM, ...)
  - PCI PassThrough / DirectPass I/O



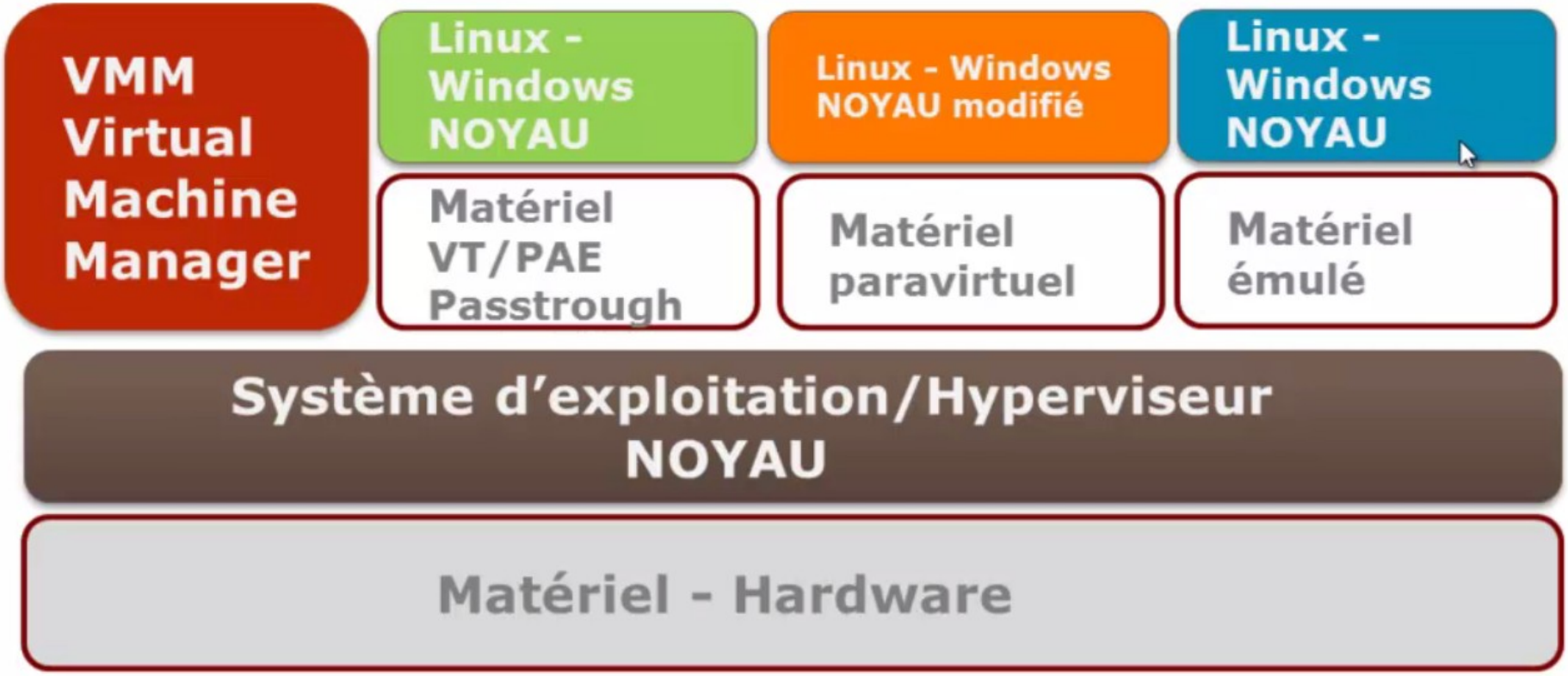
Technologies de Virtualisation

- Virtualisation assistée par le matériel (HVM)
  - Ajout d'extension de virtualisation au processeur
    - Intel-VT et AMD-V
  - Les VM's gèrent leurs propres interruptions et leurs changement de contexte
  - Plus d'émulations de zone mémoire
  - Accès direct au processeur



Technologies de Virtualisation

- La virtualisation complète
- La para virtualisation, La virtualisation assistée par le matériel

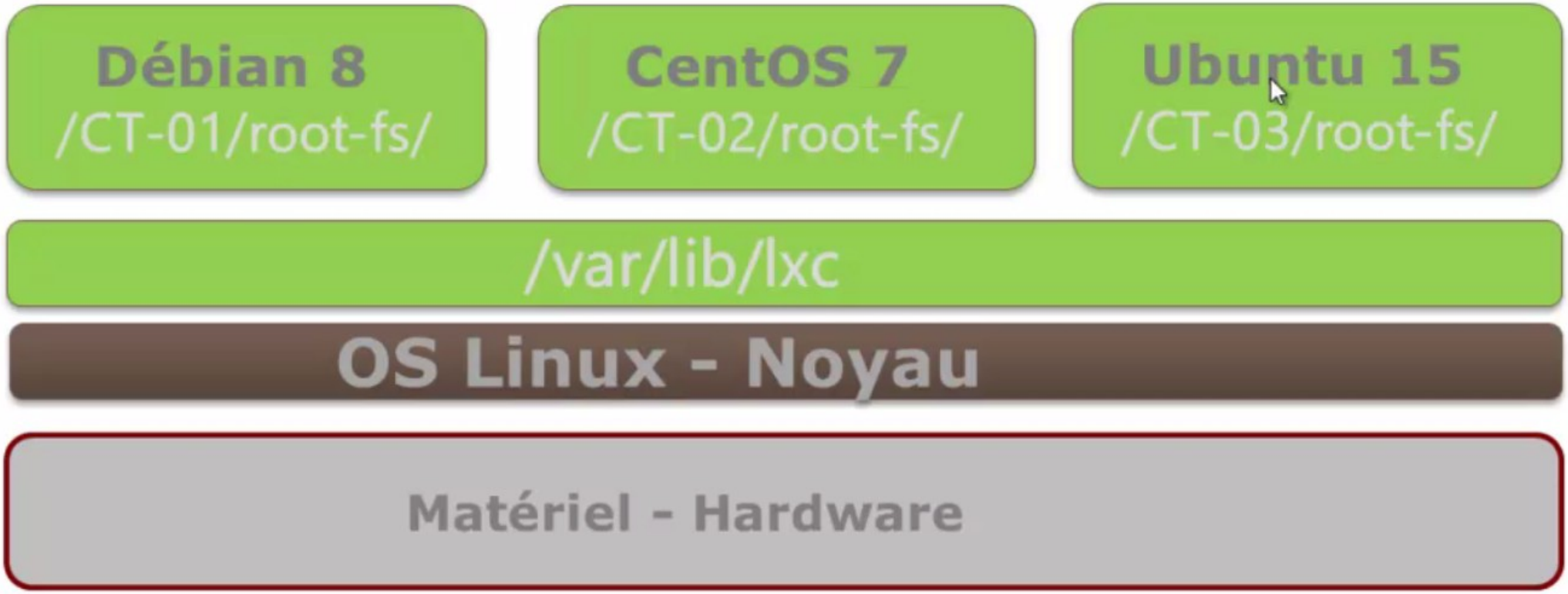


Technologies de Virtualisation

- Isolation
  - L’isolateur est une spécificité des systèmes Unix. Cette technique permet d’isoler une application du reste du monde. (VServer, chroot, bsd jail, ...)

Technologies de Virtualisation

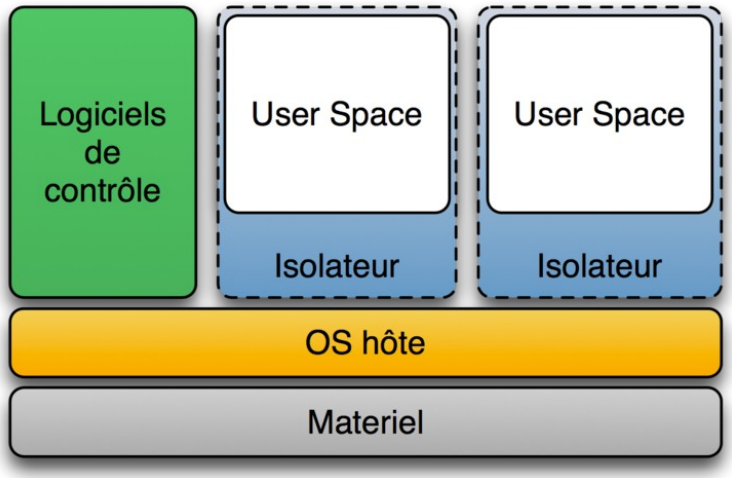
- Container
  - Pas de noyau dans les containers
  - Un répertoire = un OS
  - Pas d'émulation matériel
  - Accès direct au matériel



Technologies de Virtualisation

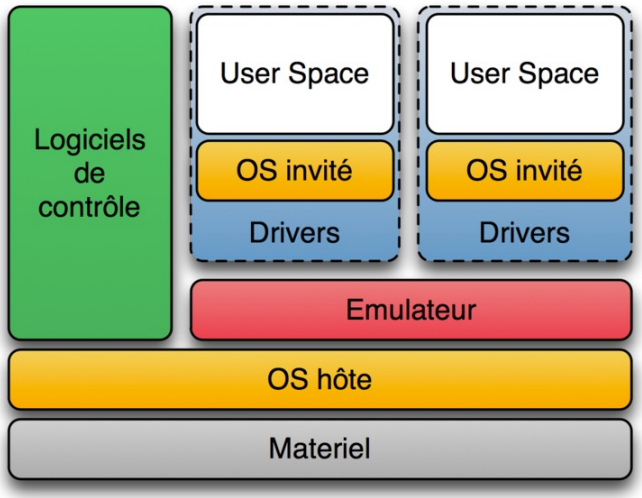
- Pour mettre en place ces techniques, il faut donc du « matériel virtuel » implémenté logiciellement ou dans le matériel:
  - CPU virtuel
    - Virtual Instruction Set
    - Virtual Memory Management Unit MMU
    - Virtual Programmable Interrupt Controller (PIC)
  - Mémoire virtuelle
  - Périphériques d'E/S virtuels
  - Timers virtuels
  - Autres périphériques

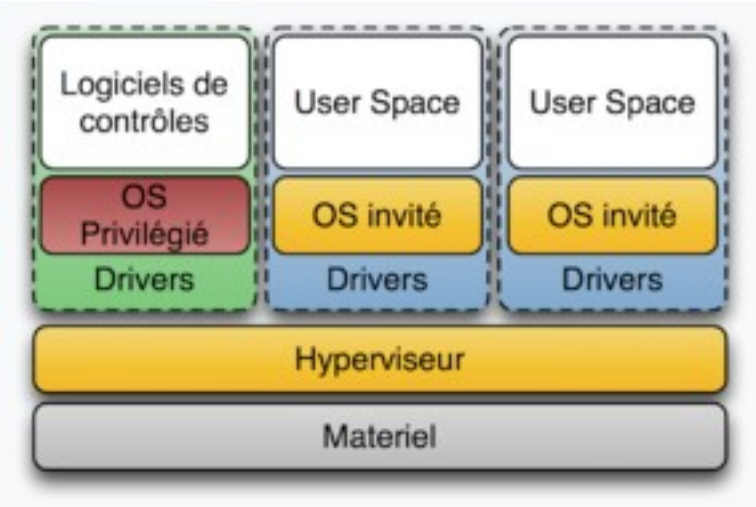
Technologies de Virtualisation



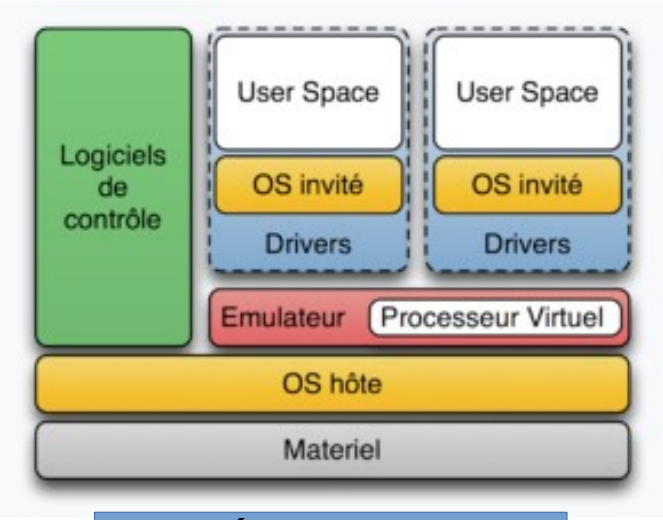
Isolateur

Hyperviseur Type II (User Space)





Hyperviseur Type I (Kernel Space « Barebone »)



Émulateur

Rappel sur les privilèges d'accès

- CPU n'accepte pas « d'office » d'exécuter toutes les instructions
  - L'OS peut accéder directement au matériel
  - Un processus normal ne peut pas
- Niveau de privilège
  - Mécanisme intégré au processeur qui empêche une application d'usurper l'accès hardware de l'OS
  - "Ring levels"
    - 0 le plus privilégié : accès complet (OS)
    - Nombre dépend de l'ISA
      - X86 : 4
      - 2 seulement utilisés par les OS : 0 et 3

Virtualisation et privilèges

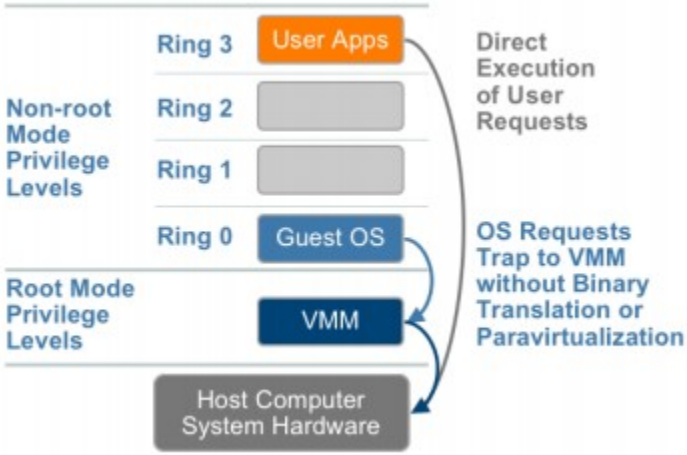
- Code user level (ring3)
  - Le code est exécuté directement sur le CPU
  - Le code d'exécution ne peut accéder directement au matériel ou à la mémoire
- Code kernel level (ring 0):
  - Seul un processus en ring 0 peut accéder **aux données et instructions** de l'état privilégié
    - L'état = Les variables et tableaux liés à un processus stockés en RAM et dans le CPU
    - Données à sauvegarder si l'on arrête le processus pour le relancer plus tard
      - Ex : Pages mémoires et flags associés
    - L'état privilégié : les données privées de l'OS à propos des processus
    - Hyperviseur gère l'accès à ces données
      - Lecture → ok
      - Ecriture / modification → interdits
  - → Virtualiser le jeux d'instructions ring0 pour cacher à la VM qu'elle
    - n'est pas dans le ring 0
    - Partage le matériel avec d'autres OS

Virtualisation et privilèges

- Mécanismes pour cacher à la VM qu'elle n'est pas dans le ring 0 et gérer l'accès aux **instructions** privilégiées (Full Virtualization) : virtualisation du jeux d'instructions
  - Trap & emulate
    - Un programme veut exécuter une instruction pour laquelle il n'a pas de privilège suffisant (pas le bon ring)
      - → une exception (trap) est générée
    - L'hyperviseur écoute les exceptions provoquées par l'OS virtualisé. Il
      - Intercepte (***traps***) l'exception: il reprend la main sur le CPU
      - Exécute l'instruction pour ***émuler*** le comportement attendu
        → Os invité ne voit voit pas qu'il est n'est pas en ring 0
- Mécanismes pour cacher et gérer l'accès aux **informations** privilégiées
  - Shadow structure
    - Un OS doit accéder et modifier certaines donnés liée au hardware, stockées dans
      - RAM
      - registres CPU
    - Un CPU ne supporte qu'un seul exemplaire de ces données → un seul OS
    - Shadow structure = une copie privée propre à une VM gérée par l'hyperviseur

Problèmes de l'ISA x86

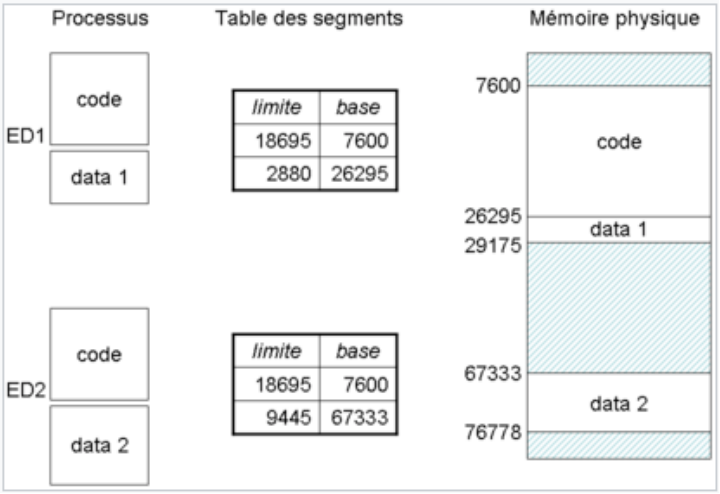
- L'ISA x86 permet l'accès au mode privilégié sans provoquer d'exception
  - Certaines instructions sont <> en ring 3 & 0 → L'hyperviseur n'est pas prévenu par manque de traps → trap & emulate impossible
  - Techniques de virtualisation "classique" du jeux d'instruction impossible
- Solutions possibles :
  - 1998 :Just in time Binary translation
    - « traduction » à la volée du code kernel non virtualisable (émulation)
      - Translation lors du 1er appel du code
      - Ensuite, stocké dans un cache pour accès plus rapide
    - Trap & emulate pour les instruction qui provoquent des traps
  - Paravirtualisation
  - Extension de l'ISA x86 → VT-x, AMD-V
    - Ajout « root mode » pour le VMM

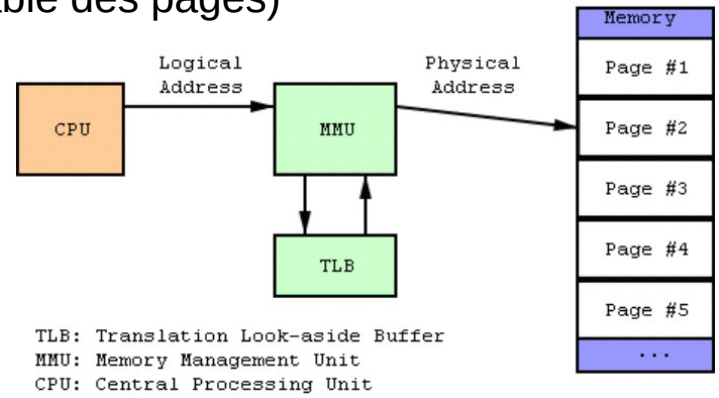


https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/techpaper/VMware_paravirtualization.pdf

Virtualisation de la mémoire

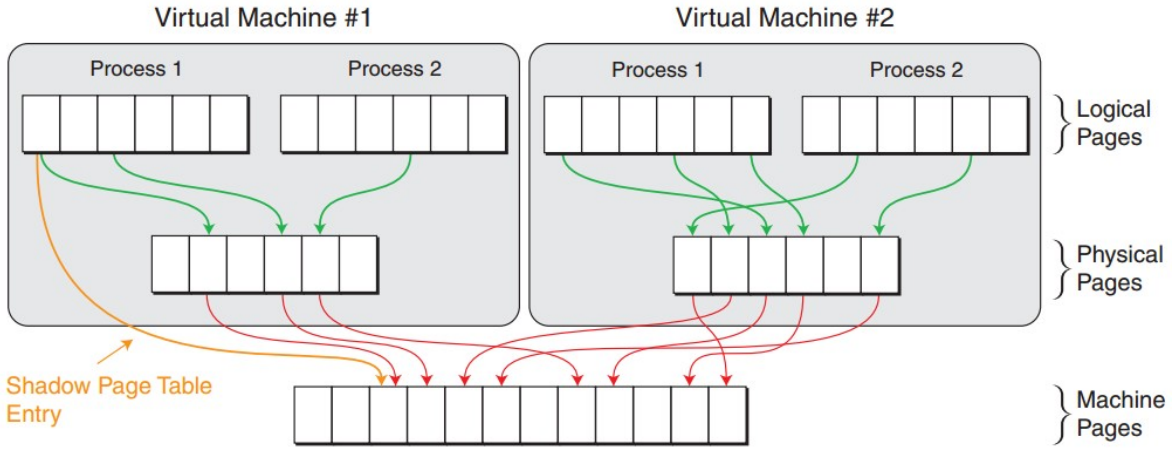
- VMM
  - Partage de la mémoire physique entre VM's
  - Allocation dynamique de la mémoire aux VM's
- Même principe que la présentation de la mémoire au processus par un OS
  - L'application voit un espace mémoire présenté par l'OS qui n'est pas lié à mémoire physique
  - L'OS gère une table de correspondance en RAM entre les pages de mémoire virtuelle et les pages physiques de la table des pages physiques
  - Le but est d'améliorer les performances de la mémoire virtuelle
    - MMU :Memory Management Unit assure la gestion matérielle de
      - Adresse virtuelle (VA) → adresse linéaire (LA) (table des segments)
      - Adresse linéaire (LA) → adresse physique (PA) (table des pages)
    - TLB
      - Mémoire cache du CPU dédiée à la translation d'adresse virtuelle (+ rapide que la RAM)





Virtualisation de la mémoire

- Virtualisation de la MMU
  - Shadow page table
    - Une shadow page table associée à chaque table de mémoire virtuelle
    - 2 associations dans une shadow page table :
      - Crée par le guest : LA VM → PA VM
      - Crée par le VMM : PA VM → adresse machine (MA)
    - une seule table pour associer la mémoire virtuelle de la VM à la MMU de la machine → presque aussi rapide qu'en « natif », sauf dans certains situations



http://www.control.lth.se/media/Staff/JohanEker/introduction_to_cloud_course/HW%20Virtualization.pdf

Virtualisation de la mémoire

- Hardware
  - 2ème génération extensions x86 pour virtualisation
    - AMD RVI et Intel EPT
    - MMU table avec 2ème table :
      - LA → PA modifiable par la VM
      - PA → MA modifiable par le VMM
    - TLB rempli avec correspondances LA → MA
    - VMM ne doit plus intervenir lors d'une modification
    - Parcours hardware des tables
  - Gain de performance : de rien ou légère perte à +- 40 %

Solutions

- Émulateurs
  - MAME (borne arcade)
  - BOCHS
  - Android ARM : intégré à Android Studio et MS Visual Studio
  - Consoles : emu pour NES, PS, etc.
- Simulateurs
- Isolateurs / containeurs
  - Chroot
  - Docker
  - LXD
  - BSD Jail
  - Linux Vserver
  - OpenVz
  - Solaris container
- Hyperviseurs Type I
  - VmWare ESXi
  - Xen
  - HyperV
  - KVM
- Hyperviseurs Type II
  - VirtualBox
  - VmWare player/Workstation
  - HyperV server (sur base de Windows Core)
  - Qemu
  - Parallels (mac)

Cloud

- Définition : Mise à disposition à la demande de ressources informatiques partagées (CPU, mémoire, stockage, réseau) via un réseau ou Internet. Ressources en général situées dans un datacenter.
- Types
  - Iaas
    - Infrastuture as a service
    - Fourniture de VM's
    - Stockage
    - Réseau
    - Load balancers
    - Ex : OpenStack
  - Paas
    - Platform as a service
    - Hébergement d'application
    - Environnement de développement (Middelware) avec DB, réseaux, etc.
    - Ex : Azure, Google Engine
  - Saas
    - Software as a service
    - Ex : Google Docs, Office 365
  - Publique
    - Via un fournisseur Cloud
    - !!sécurité !!!
    - Ex : Amazon, Rackspace, SalesForce
  - Privé
    - Cloud interne
    - Vsphère, OpenStack, Citrix
  - Hybride
    - Mix de privé et publique

Containers

- Virtualisation au niveau de l'OS
  - Plusieurs instances du même OS qui tournent avec le même Kernel
- Chaque container contient tout ce qui est nécessaire pour tourner:
  - Application
  - Dépendances de l'espace utilisateur (JVM, shell, etc.)
  - Librairies
- Éléments de l'espace kernel fournis par l'OS hôte
- OS hote
  - Isole les ressources des containers
  - Alloue les ressource CPU et RAM
- → Container doit être prévu pour un OS
- Avantage
  - Déploiement facile de plusieurs instances d'un application
  - Isolation
  - Performance ÷ Machine Virtuelle
- Inconvénient
  - Dépendant de l'OS → pas Cross platform
  - Isolation moins forte que avec une VM