MODULE :

Le Cloud Computing & DevOps

Pr. F. Benabbou
Master DSBD
Faculté des Sciences Ben M’Sik Casablanca

TABLE OF CONTENTS

01

LE CLOUD COMPUTING

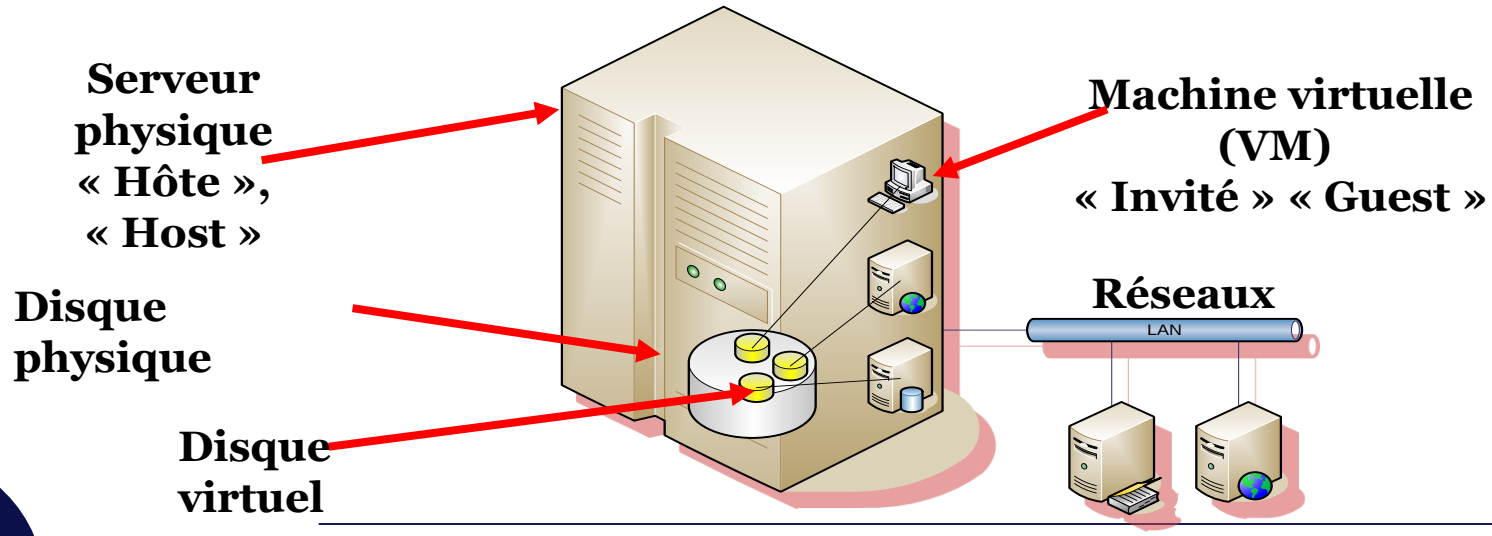
Définition

« La Virtualisation est un processus qui va permettre de masquer les caractéristiques physiques d'une ressource informatique de manière à simplifier les interactions entre cette ressource et d'autres systèmes, d'autres applications et les utilisateurs.
Elle va permettre de percevoir une ressource physique comme plusieurs ressources logiques et, inversement, de percevoir plusieurs ressources physiques comme une seule ressource logique ».

«La Virtualisation est la création d'une version virtuelle (non réel) de quelque chose, comme un système d'exploitation, un serveur, un dispositif de stockage ou les ressources du réseau »

C’est quoi une Machine virtuelle?

- Une machine virtuelle souvent abrégée VM est un environnement virtuel fonctionnant comme un ordinateur à partir de ressources virtuelles
- Une MV est doté de son propre système d’exploitation (système invité) qui est indépendant de l'architecture et des spécificités matériels de la machine physique (système hôte) sur lequel elle s’éxecute
- Le système invité est vu par l’hôte comme un simple programme, il n’a pas d’accès direct au matériel contrairement à l’hôte.

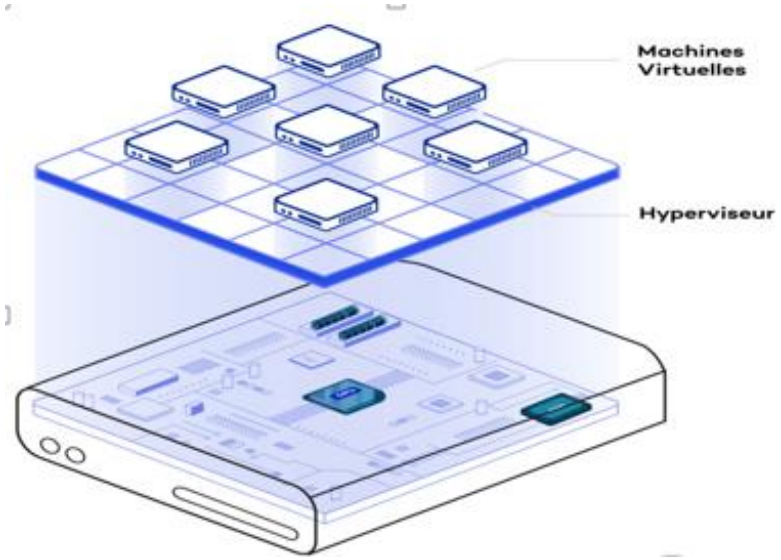


Machine Virtuelle/ hôte

- La VM exécute donc son propre système d'exploitation et s'appuie sur les mêmes équipements : Disque, RAM, processeur, carte réseau, écran, clavier, souris …
- L'utilisateur peut exécuter le système invité comme un système normal : installer des applications, naviguer sur Internet, exécuter un programme, etc.
- on peut d'utiliser une seule machine physique pour exécuter plusieurs MVs avec des systèmes d'exploitation, qui peuvent être totalement différents de celui installé sur la machine physique.

Couche de virtualisation

- Dans le processus de virtualisation nous avons besoin d'ajouter une couche d'abstraction entre les systèmes invités et les ressource physiques.
- Son rôle est de transformer les instructions envoyées au système invité en instructions exécutables par le système hôte.
- L'ajout de cette couche introduit évidemment un surcoût en performance qui varie selon les traitements effectués au sein de cette couche.



- le système virtualisé ne pourra pas accéder au matériel directement,
- l'accès se fera à travers la couche de virtualisation .

Architecture virtualisable

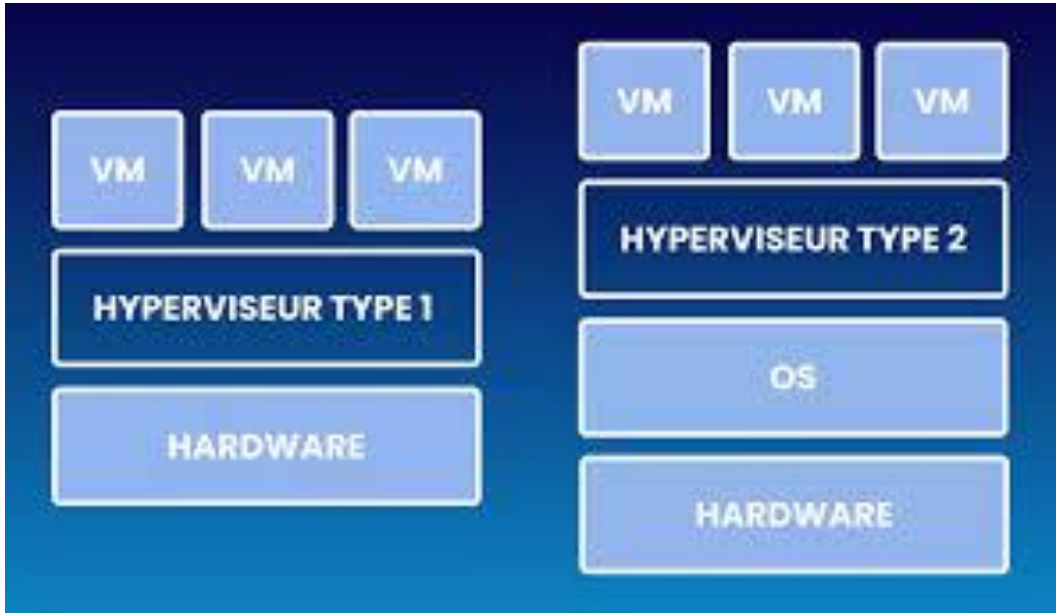
- D’après la théorie de Popek et Goldberg, pour qu’une architecture de processeur soit virtualisable efficacement, il faut que l’immense majorité des instructions envoyées au processeur virtuel peut être exécutée directement sur le processeur physique, sans intervention de l’hyperviseur
- Dans ce cadre, trois critères doivent être remplies pour qu'une architecture informatique soit virtualisable efficacement:
  - **Transparence de l'architecture** : l'hyperviseur doit être en mesure d'identifier et d'intercepter toutes les instructions sensibles et de les exécuter de manière sûre et efficace.
  - **Isolation des machines virtuelles** : L’accès d’une machine virtuelle à la mémoire doit pouvoir être restreint à ses propres données, afin de protéger l’hyperviseur et les autres machines virtuelles
  - **Équivalence des privilèges** : chaque instruction d'une machine virtuelle doit s'exécuter de la même manière et avec les mêmes résultats que si elle était exécutée directement sur la machine physique.

L’hyperviseur

- La couche de virtualisation est souvent nommée Hyperviseur
- Un **hyperviseur**, également appelé superviseur de machine virtuelle ou monitor de machine virtuelle (VMM), est un logiciel qui crée et gère des machines virtuelles (VM).
- Il permet à plusieurs systèmes d'exploitation et applications de partager un seul hôte matériel et ses ressources.
- Son rôle est de :
  - Fournir un environnement d’exécution virtuel aux systèmes invités
  - Allouer les ressources nécessaires à leur fonctionnement
  - Rediriger les requêtes d’entrées sorties vers les ressources physiques
  - Veiller au confinement des invités dans leur propre espace.

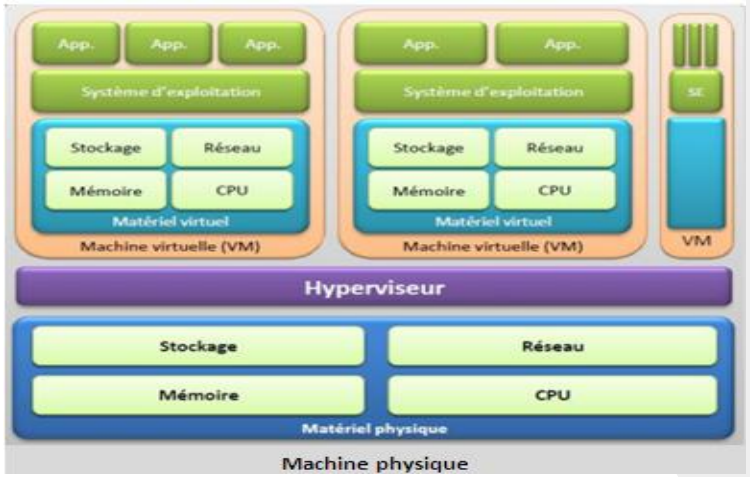
Type d'hyperviseur

- Hyperviseur de type 1(bare metal) :
  - Fonctionne directement sur le matériel physique de l'hôte.
  - Gère les machines virtuelles sans nécessiter de système d'exploitation hôte.
  - Exemples : VMware ESXi, Microsoft Hyper-V, Xen, Red Hat virtualisation
- Hyperviseur de type 2: (ou hébergé)
  - Fonctionne au-dessus d'un système d'exploitation hôte.
  - Les machines virtuelles sont gérées par le système d'exploitation de l'hôte.
  - Exemples : VMware Workstation, Oracle VM VirtualBox, Parallels Desktop.



Hyperviseur de Type 1

- L’hyperviseur de Type 1 est sous forme d’un noyau système très léger.
- Solution dédiée aux entreprises
- Permet d’optimiser les ressources physiques
- Fournit la haute disponibilité des serveurs
- Plusieurs éditeurs proposent des solutions logicielles de virtualisation avec hyperviseur comme:
  - ESXi Server de Vmware, Microsoft Hyper-V, TRANGO l'hyperviseur LPAR de IBM, KVM (Kernel-based Virtual Machine) pour Linux, Xen Citrix, etc,



Microsoft Hyper-V

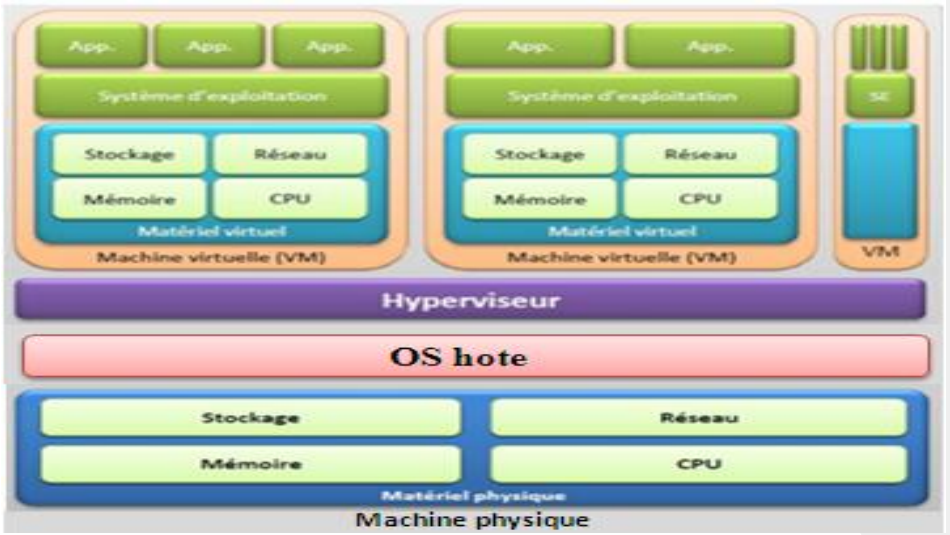
vmware ESXi

CITRIX XenServer

KVM

Hyperviseur de Type 2

- Un hyperviseur de type 2 consomme aussi les ressouces de la machine en plus du système d'exploitation qui le supporte.
- Toutes les opérations de l'invité sont interceptées et traduites pour être exécutées par l'environnement hôte, ce qui est une méthode très consommatrice en ressources.
- Solution adaptée à:
    - Pour l'entreprise et les particuliers
    - Permet de faire des tests un OS en toute sécurité
    - Permet d'exécuter plusieurs hyperviseurs simultanément vu qu'ils ne sont pas liés à la couche matérielle.
- Exemple de solution
    - Microsoft Virtual PC, VMware Player, QEMU, Oracle VirtualBox, VMware Workstation…



ORACLE VM VirtualBox

VMware Workstation Player

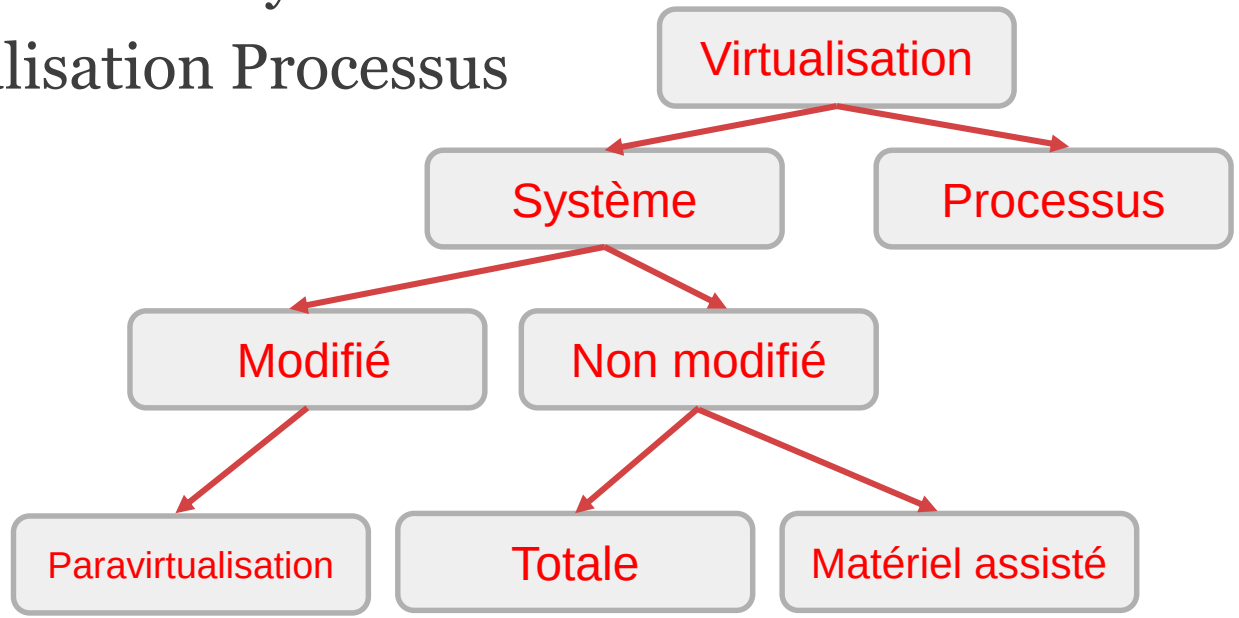
QEMU

vmware Workstation

Microsoft Virtual PC

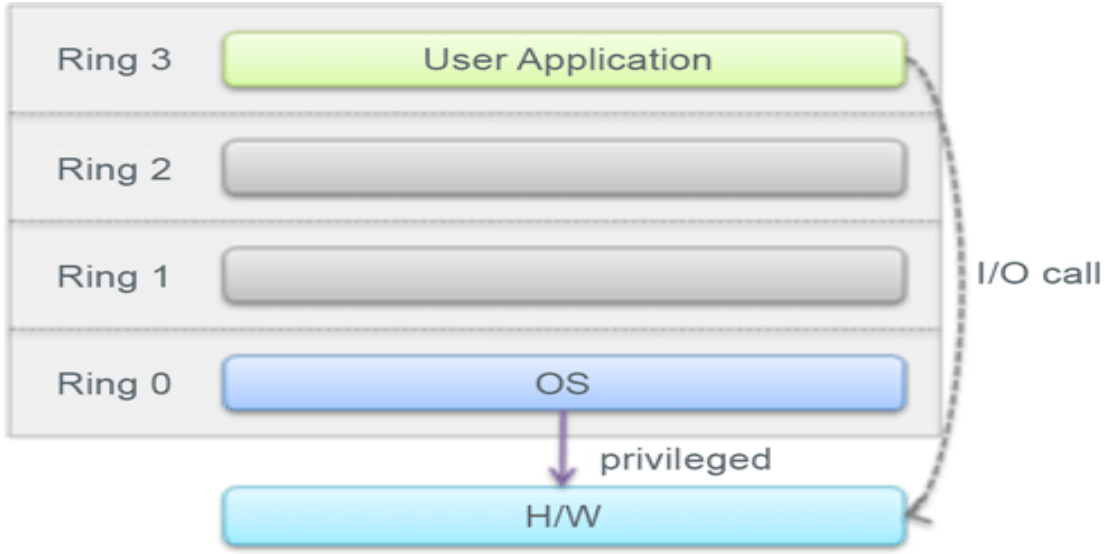
Techniques de Virtualisation

- Les techniques de virtualisation peuvent être classées en 2 catégories principales:
  - Virtualisation Système
  - Virtualisation Processus



L’architecture x86

- Lorsqu’AMD et Intel ont refondu l’architecture x86 (dont Linux et Windows) pour passer au 64-bit, ils ont décidé de supprimer les anneaux de privilège 1 et 2.
- Il ne reste que les anneaux 0 et 3.
- On parle couramment de ces anneaux avec les termes « mode noyau» et « mode utilisateur ».

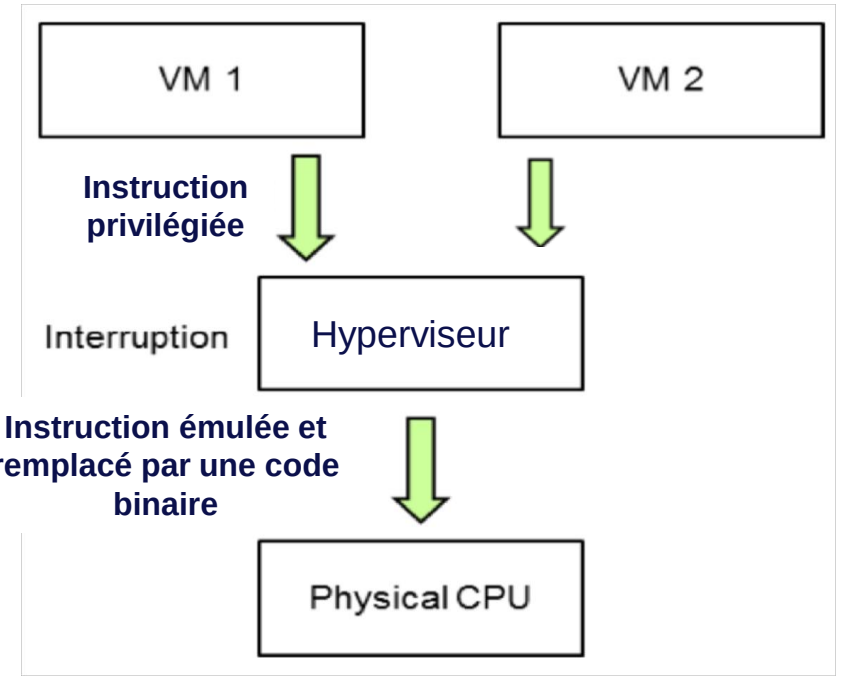


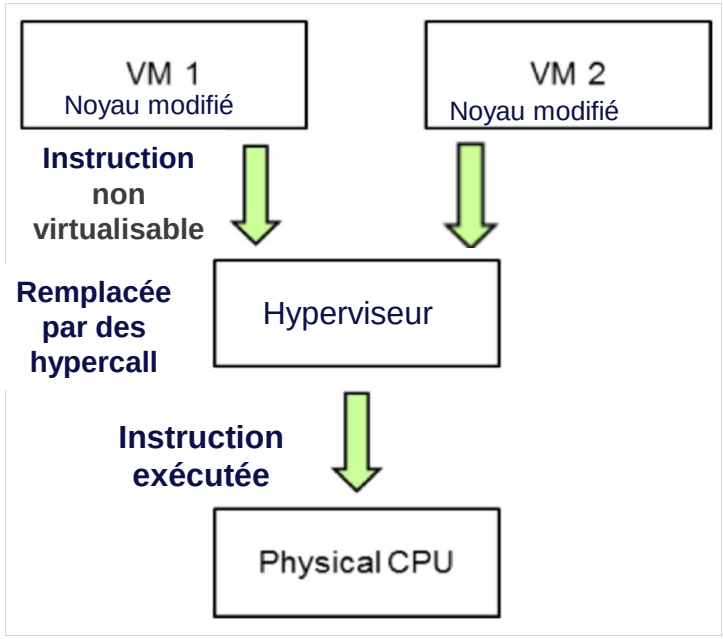
Type d’instructions

- Le jeu d’instruction d’un processeur est l’ensemble des instructions que ce processeur est en mesure de décoder.
- L'hyperviseur joue un rôle central pour gérer ces instructions et assurer une isolation complète et sécurisée entre les machines virtuelles.
- Deux types d’instruction sont problématiques
  - **Instructions privilégiées** :
    - Ce sont des instructions processeur qui ne peuvent être exécutées que par le système d'exploitation en mode noyau car elles ont un accès direct aux ressources matérielles comme la mémoire, les périphériques d'E/S et le processeur.
    - Exemples d'instructions privilégiées : gestion des interruptions, modifications des registres de contrôle, accès direct aux périphériques.
  - **Instructions non virtualisables** :
    - Ce sont des instructions qui, lorsqu'elles sont exécutées en dehors du mode privilégié (par exemple, dans un environnement utilisateur ou invité), ne génèrent pas les exceptions ou les interruptions nécessaires pour permettre à l'hyperviseur de les intercepter.
    - Cela empêche la virtualisation transparente, car ces instructions peuvent compromettre l'isolement des machines virtuelles ou fausser les résultats.

Techniques de virtualisation

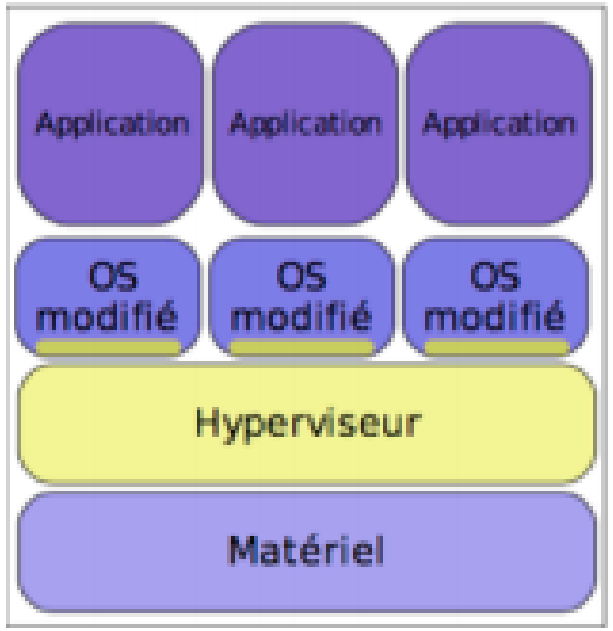
- Chacune des techniques de virtualisation proposent des solutions pour pouvoir exécuter ce genre d'instruction sans mettre en péril l'isolation des systèmes virtuels.





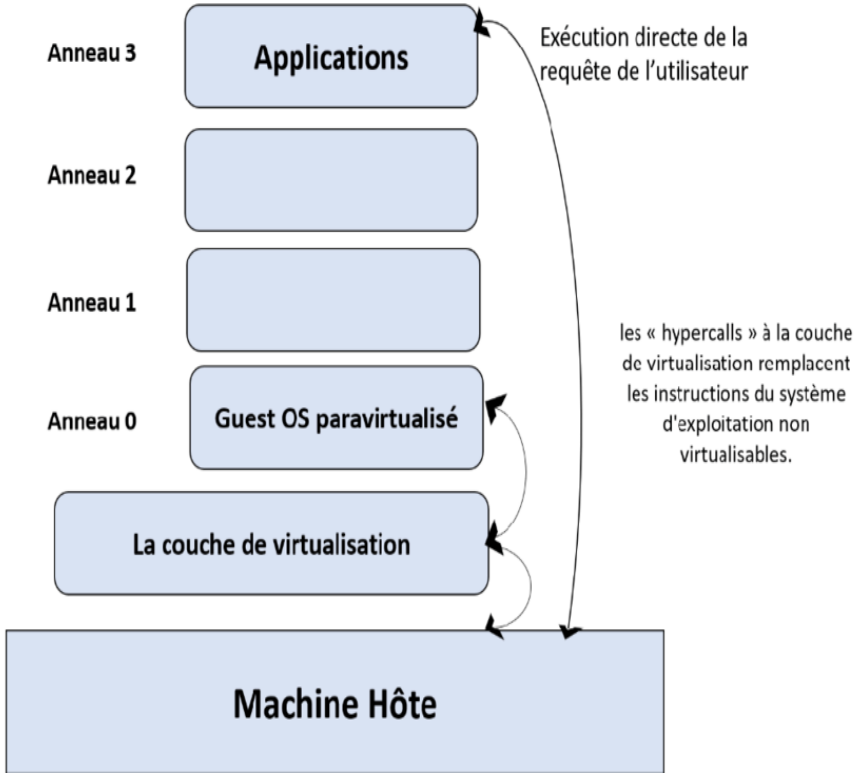
Paravirtualisation

- La paravirtualisation est la virtualisation de systèmes d'exploitation dont le noyau du **système invité a été modifié** pour simplifier la communication avec l'hyperviseur.
- Les modifications permettent de faciliter la communication avec l'hyperviseur
- L'hyperviseur gère les accès des systèmes invités à l'architecture matérielle sous-jacente.
- Les machines virtuelles (OS invités) peuvent être de type différents mais d'architecture identique (x86 par exemple).



Paravirtualisation

- Les hypercalls sont des appels spécifiques utilisés pour permettre au système d'exploitation invité d'accéder aux ressources virtuelles, tels que les processeurs virtuels, la mémoire virtuelle ou les périphériques virtuels de la machine physique.
- les instructions privilégiées sont interceptées et remplacées par des hypercalls, qui sont traités et exécutées directement sur le matériel par l'hyperviseur
- Ainsi, le système d'exploitation invité peut appeler directement des fonctions de l'hyperviseur pour réaliser certaines tâches, comme la gestion de la mémoire, les opérations d'E/S (Entrée/Sortie), ou la planification des processus.



Paravirtualisation

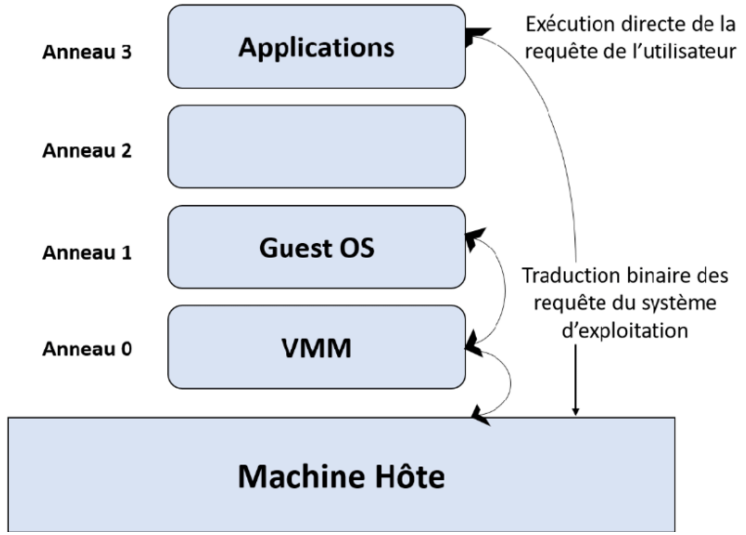
- La paravirtualisation apporte un gain de performances important
- La modification du noyau ne pose aucun problème pour un système libre comme GNU/Linux, mais il en est pas de même pour les systèmes propriétaires, tels que Microsoft Windows et Mac OS.
- L'usage de la paravirtualisation est donc généralement limité aux systèmes libres, ou à une solution de virtualisation propriétaire compatible avec un seul système d'exploitation invité

Virtualisation totale

- l'émulation est le principe qui consiste à remplacer un composant matériel par une application dont le comportement est similaire voire identique.
- La virtualisation totale consiste à émuler l'intégralité d'une machine physique pour le système invité.
- L'hyperviseur se charge de la traduction complète des instructions du système invité en instructions exécutables sur le système hôte.
- Le noyau de l'OS ne subi aucune modification

Virtualisation totale

- la virtualisation totale utilise la **traduction binaire** pour exécuter les instructions privilégiées dans l'environnement de la machine virtuelle
- Lorsqu'une instruction problématique est émise par la MV, elle est interceptée par l'hyperviseur et traduite en une série de instructions équivalentes qui peuvent être exécutées sur le matériel de la machine hôte.
- Un traducteur binaire remplace les instructions sensibles par des instructions non sensibles (op-code) équivalentes compatible avec le processeur de l'environnement cible



Virtualisation totale

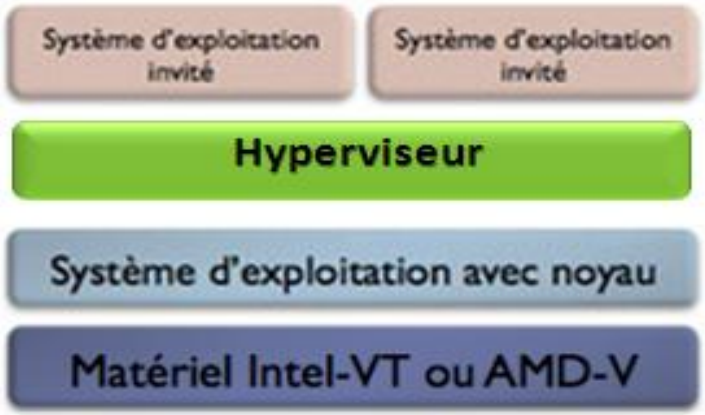
- Un des avantages de la virtualisation totale est de pouvoir émuler n'importe quelle architecture matérielle.
- On peut donc faire fonctionner les OS que l'on désire indépendamment de l'architecture du système hôte.
- L'inconvénient c'est la dégradation des performances liée au processus de traduction binaire, la cohabitation de deux OS invité et et celui du hôte

Translation binaire

- Le coût d'analyse et de traduction des instructions est important
- Mais il peut être fortement réduit par diverses optimisations, surtout dans le cas ou la traduction a lieu entre deux jeux d'instructions identiques (guest et host)
- Autres possibilités
  - **Caches de traduction** : On utilise des caches de traduction afin de ne traduire le code qu'une seule fois. Ainsi, le traducteur n'est appelé que lorsque le processeur virtuel arrive à une instruction qui n'a pas encore été analysée.
  - **Translation dynamique :**
    - ✓ inspiré du principe de la compilation à la volée (JIT Just-In-Time), une technique de compilation dans laquelle le code est compilé à la volée pendant l'exécution du programme, plutôt qu'à l'avance.
    - ✓ dans ce cas les instructions de la machine source sont traduites en instructions de la machine cible **au moment même de leur exécution** et le code traduit peut être adapté en fonction d'informations statistiques recueillies à l'exécution.
    - ✓ On traduit seulement les parties de code qui sont réellement utilisées, ce qui optimise l'exécution.

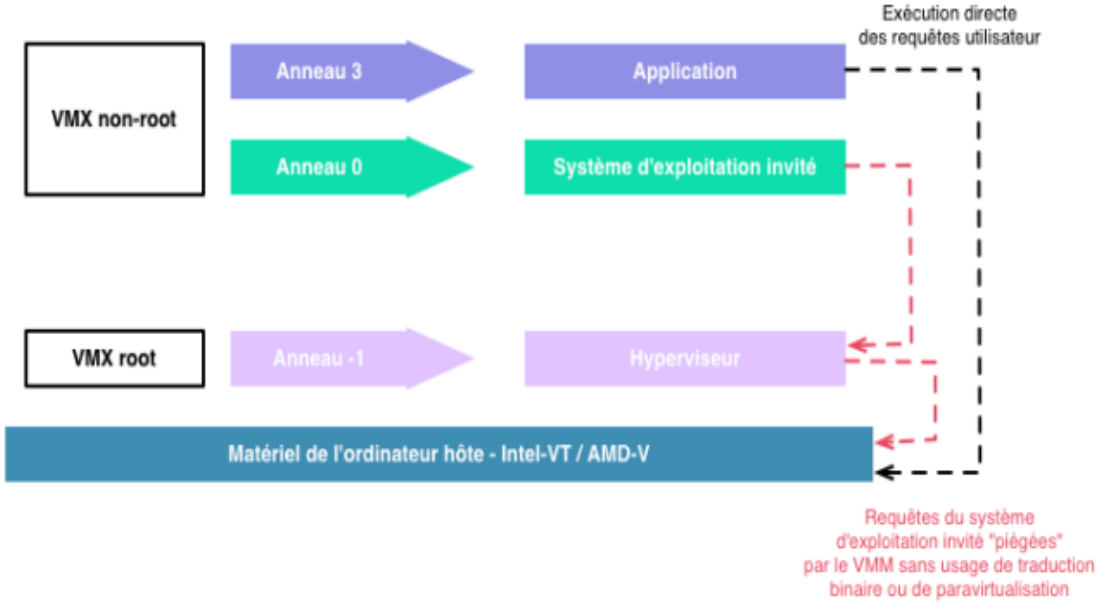
La virtualisation matérielle assistée

- Il s’agit d’une extension du principe de virtualisation totale.
- Elle utilise des fonctionnalités matérielles spéciales pour permettre à l'hyperviseur d'exécuter certaines instructions directement sur le matériel de la machine hôte, sans passer par la traduction binaire.
- Les principaux fabricants de processeurs comme Intel et AMD ont développé des extensions pour leurs architectures, notamment :
  - Intel VT-x (Virtualization Technology for x86) pour les processeurs Intel.AMD-V (AMD Virtualization) pour les processeurs AMD.
- Ces extensions ajoutent de nouvelles instructions et fonctionnalités au processeur qui facilitent la gestion des environnements virtualisés et réduire la charge de travail de l'hyperviseur.



La virtualisation matérielle assistée

- La technologie VT introduit un nouveau mode d'exécution au sein des processeurs concernés, connue sous le nom de VMX
- Ce dernier comporte un niveau racine (VMX root operation) et un niveau correspondant aux anciens anneaux 1 à 3 (VMX non-root operation).
- Le niveau VMX root operation est destiné à l'hyperviseur alors que le niveau VMX non-root operation fournit un environnement contrôlé par le VMM et conçu pour supporter une VM.
- Chaque niveau d'opération comporte les quatre niveaux de privilèges (ring0 à ring3).
- Le noyau du système d'exploitation invité peut donc évoluer au sein de l'anneau 0, les applications exécutées en son sein, au niveau 3, le VMM disposant de son propre jeu d'anneaux (parfois appelé l'anneau -1).



La virtualisation matérielle assistée

- On a une nette amélioration de performance et réduction de la charge de travail de l'hyperviseur.
- Cependant, il faut noter que la virtualisation assistée par le matériel ne permet pas d'exécuter toutes les instructions directement sur le matériel.
- Certaines doivent encore être traduites en code exécutable sur le matériel.
- Cette technique de virtualisation a été implantée dans les processeurs à base d'architecture x86 sous les noms de :
  - les processeurs Intel-VTx (32 bits) et Intel VT-i (64 bits) pour Intel
  - les processeurs AMD-V pour AMD, les processeurs POWER6, POWER7, POWER8, POWER9, et POWER10 d'IBM, les processeurs UltraSPARC T1, T2, T3, T4, T5 et M7 de Sun (actuellement Oracle)

Remarques

- Bien que les méthodes de virtualisation citées ont fait leur preuves, on remarque qu'il y a un gaspillage de la mémoire du hôte ce qui limite forcément le nombre de VM prises en charge par chaque serveur.
- En effet chaque VM à ses propres ressources virtuelles qui ont leur projection sur les ressources physique gérées par le système hôte
- Pourquoi pas une solution qui permet aux instances virtuelles de partager un système d'exploitation ?

La virtualisation des processus

- La virtualisation des processus est une technique qui a pour but de créer un environnement d'exécution isolé à l'intérieur d'un système d'exploitation hôte.
- l'objectif est de faire fonctionner plusieurs applications ou processus en même temps sur un même système d'exploitation, sans qu'ils ne puissent interférer les uns avec les autres.
- Cette technique le concept de « conteneurs » ou de « zones » isolées dans lesquelles chaque application ou processus dispose de ses propres ressources, tels que la mémoire, le processeur, le réseau et le système de fichiers.
- Chaque conteneur est ainsi capable de fonctionner de manière autonome, sans perturber les autres conteneurs ou le système d'exploitation hôte.

Exemples d’outils

- Docker : Docker est un outil de virtualisation des processus open source qui permet de créer, de déployer et de gérer des applications dans des conteneurs.
- LXC (Linux Containers) : LXC est une technologie de virtualisation des processus qui permet d'isoler et de gérer les processus Linux dans des conteneurs.
- OpenVZ : OpenVZ est une technologie de virtualisation des processus pour les systèmes d'exploitation Linux.
- FreeBSD Jails : FreeBSD Jails est une technologie de virtualisation des processus pour les systèmes d'exploitation FreeBSD.
- Windows Containers : Windows Containers est une technologie de virtualisation des processus pour les systèmes d'exploitation Windows.

Exemples d’applications

- Hébergement de sites web :
  - Permet d’isoler les sites web les uns des autres, afin de garantir leur sécurité et leur performance.
  - Chaque site web peut ainsi fonctionner dans son propre conteneur, avec ses propres ressources allouées, sans affecter les autres sites web hébergés sur le même serveur.
- Développement de logiciels :
  - Permet aux développeurs de logiciels d’utiliser des environnements de développement virtuels pour simplifier la configuration et la gestion des environnements de test et de production.
- Systèmes de gestion de bases de données :
  - Permet aux SGBD d’isoler les bases de données les unes des autres, et garantir leur sécurité et leur performance sans affecter les autres bases de données hébergées sur le même serveur.
- Virtualisation de postes de travail :
  - Chaque utilisateur peut fonctionner dans son propre environnement virtuel, avec ses propres applications et ses propres données, sans affecter les autres utilisateurs partageant le même matériel physique.
- Services cloud
  - Les fournisseurs peuvent créer des environnements d'exécution isolés pour leurs clients, afin de garantir la sécurité et la performance de leurs applications et de leurs données,
  - chaque client fonctionne dans son propre conteneur, avec ses propres ressources allouées, sans affecter les autres clients utilisant les mêmes ressources physiques.