



# La virtualisation

## Motivations

Environnements traditionnels :

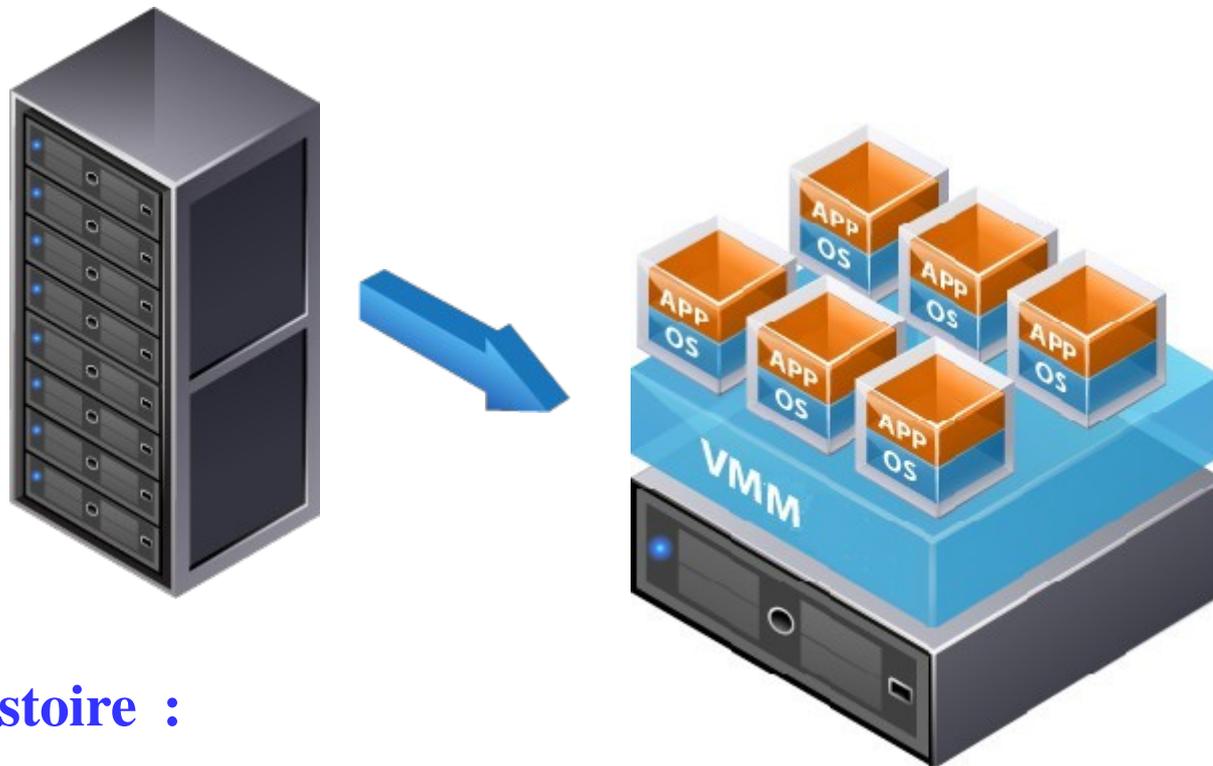
Les applications utilisent entre 10 % et 15 % des ressources en moyenne (sources VMware)

- L'agrandissement de l'environnement est très coûteux
- L'administration de l'environnement est complexe

# Virtualisation

## Définition :

Ensemble des techniques matérielles et/ou logicielles qui permettent de faire fonctionner simultanément sur une seule machine plusieurs systèmes d'exploitation (appelés machines virtuelles (VM)). Ex.: Xen, VMware, KVM, HyperV, etc.



## Un peu d'histoire :

La virtualisation est utilisée pour première fois en 1967 par IBM pour optimiser le partage de grosses machines entre différents processus.

# Virtualisation

## Définition :

**Selon le site Wikipédia, La virtualisation consiste à faire fonctionner un ou plusieurs systèmes d'exploitation ou applications comme un simple logiciel, sur un ou plusieurs ordinateurs ou serveurs et système d'exploitation, au lieu de ne pouvoir en installer qu'un seul par machine. Ces ordinateurs virtuels sont appelés serveur privé virtuel (Virtual Private Server ou VPS) ou encore environnement virtuel (Virtual Environment ou VE).**

# Virtualisation

## Définition :

**La virtualisation permet d'ajouter une couche d'abstraction qui sépare le système d'exploitation du matériel afin de délivrer une meilleure utilisation et flexibilité des ressources de traitement.**

**La virtualisation repose sur trois éléments importants :**

- **L'abstraction des ressources informatiques ;**
- **La répartition des ressources par l'intermédiaire de différents outils, de manière à ce que celles-ci puissent être utilisées par plusieurs environnements virtuels ;**
- **La création d'environnements virtuels.**



# Les domaines de la virtualisation

## La virtualisation d'applications

**La virtualisation d'application est une technologie logicielle qui va permettre d'améliorer la portabilité et la compatibilité des applications en les isolant du système d'exploitation sur le quel elles sont exécutées. Elle consiste à encapsuler l'application et son contexte d'exécution système dans un environnement cloisonné.**



# Les domaines de la virtualisation

## La virtualisation d'applications

**La virtualisation d'application va nécessiter l'ajout d'une couche logicielle supplémentaire entre un programme donné et le système d'exploitation ; son but est d'intercepter toutes les opérations d'accès ou de modification de fichiers ou de la base de registre afin de les rediriger de manière totalement transparente vers une localisation virtuelle.**

# Les domaines de la virtualisation

## La virtualisation d'applications

la couche virtuelle va ajouter des avantages au système virtualisé en permettant d'exécuter des applications conçues pour d'autres systèmes.

### **Exemple :**

Wine est un logiciel qui permet d'exécuter *certain*[http://doc.ubuntu-fr.org/wine#fn\\_\\_1](http://doc.ubuntu-fr.org/wine#fn__1) programmes Windows sous Ubuntu. <http://www.winehq.org/>

On peut aussi citer l'avantage gagné au niveau de la protection du système d'exploitation hôte en s'assurant que l'application virtualisée ne viendra pas interagir avec les fichiers de configuration du système.

# Les domaines de la virtualisation

## La virtualisation de réseaux

De manière générale, la virtualisation des réseaux consiste à partager une même infrastructure physique (débit des liens, ressources CPU des routeurs,...) au profit de plusieurs réseaux virtuels isolés. Un VLAN (Virtual Local Area Network) est un réseau local regroupant un ensemble de machines de façon logique et non physique. Puisqu'un VLAN est une entité logique, sa création et sa configuration sont réalisées de manière logicielle et non matérielle.

# Les domaines de la virtualisation

## La virtualisation de réseaux

On distingue plusieurs types de réseaux virtuels :

Les réseaux virtuels de niveau 1, appelés réseaux virtuels **par port** (port-based VLAN): ils définissent un réseau virtuel en fonction des **ports** de raccordement sur le commutateur (switch). Ainsi, chaque port du commutateur est associé **à un réseau virtuel**, indépendamment de la machine qui y est physiquement raccordée. Le principal inconvénient d'un VLAN de niveau 1 est sa rigidité : si une station se raccorde physiquement au réseau par l'intermédiaire d'un autre port du commutateur, alors il est nécessaire de reconfigurer ce commutateur afin de réintégrer la station dans le bon réseau virtuel.

# Les domaines de la virtualisation

## La virtualisation de réseaux

Les réseaux virtuels de niveau 2, appelés réseaux virtuels par adresse MAC (MAC address-based VLAN) : ils consistent à définir un réseau virtuel sur base des adresses MAC des stations. Une adresse MAC est un identifiant unique implémenté dans chaque adaptateur réseau. Ce type de VLAN est beaucoup plus souple que le précédent car il est indépendant de la localisation de la machine.

# Les domaines de la virtualisation

## La virtualisation de réseaux

Les réseaux virtuels de niveau 3. On distingue principalement deux types de VLAN de niveau 3 :

Les réseaux virtuels par adresse de sous-réseau (Network address-based VLAN) : ils déterminent les réseaux virtuels sur base de l'adresse IP source des segments. Ce type de réseau virtuel est très flexible puisque les commutateurs adaptent automatiquement leur configuration lorsqu'une station est déplacée.

Les réseaux virtuels par protocole (Protocol-based VLAN). Dans ce cas, les réseaux virtuels sont créés sur base des protocoles utilisés (TCP/IP, IPX,...) et les stations sont regroupées en réseaux virtuels suivant le protocole qu'elles utilisent.

# Les domaines de la virtualisation

## La virtualisation de réseaux

Les avantages qu'offrent les réseaux virtuels sont les suivants :

Une réduction du trafic de diffusion, puisque celui-ci est à présent contenu au sein de chaque réseau virtuel ;

Une sécurité accrue puisque l'information est encapsulée dans une couche supplémentaire ;

Une meilleure flexibilité puisqu'une modification de la structure des réseaux peut être réalisée en modifiant la configuration du commutateur.

# Les domaines de la virtualisation

## La virtualisation de stockage

**Dans une machine virtuelle, les données sont stockées sur un disque dur virtuel. Ce disque dur se présente sous forme de fichier dans le système de fichiers de l'hôte :**

- VHD chez Microsoft**
- VDI chez Oracle**
- VMDK chez VMWare**
- OVF pour le format ouvert**



# Les domaines de la virtualisation

## La virtualisation de stockage

**Les disques virtuels peuvent être statiques ou dynamiques. Dans le cas où le disque est statique, si on crée un disque de 50 Go, le fichier de disque virtuel fera 50 Go sur le système hôte. Avec un disque dynamique, le fichier de disque virtuel se remplit au fur et à mesure qu'il est utilisé. Un disque de 50 Go dans lequel il n'y a pas de données ne pèsera dans le système de fichiers hôte grande chose.**

# Les domaines de la virtualisation

## La virtualisation de stockage

**La virtualisation de stockage permet :**

**d'adjoindre un périphérique de stockage supplémentaire sans interruption des services;**

**de regrouper des unités de disques durs de différentes vitesses, de différentes tailles et de différents constructeurs ;**

**de réallouer dynamiquement de l'espace de stockage. Ainsi, un serveur nécessitant un espace de stockage supplémentaire pourra rechercher des ressources non allouées sur le disque logique.**

**Inversement, un serveur nécessitant moins d'espace de stockage pourra libérer cet espace et le rendre disponible pour d'autres serveurs.**



# Les domaines de la virtualisation

## La virtualisation de serveurs

**D'une manière générale, la virtualisation de serveur est un principe permettant de faire fonctionner simultanément, sur un seul serveur physique, plusieurs serveurs virtuels. Cette technique permet aux entreprises d'utiliser des serveurs virtuels en lieu et place de serveurs physiques. Si cette virtualisation est faite au sein de la même entreprise, le but est de mieux utiliser la capacité de chaque serveur par une mise en commun de leur capacité.**

# Les domaines de la virtualisation

## La virtualisation de serveurs

**La virtualisation de serveur permet de :**

**Regrouper plusieurs serveurs physiques sous-employés sur un seul hôte qui exécute des systèmes virtuels ;**

**Réduire la surface au sol, la consommation électrique, le besoin de climatisation et le nombre d'administrateurs ;**

**Réaliser des économies (locaux, consommation électrique, personnel).**

## Différents types de systèmes de virtualisation (VMM)

- Virtualisation complète
- Virtualisation niveau OS
- Para-virtualisation
- Virtualisation assistée par le matériel

# Virtualisation

Différents types de systèmes gestion de virtualisation (VMM)

## Virtualisation complète

Un OS de base qui exécute des logiciels parmi lesquels la VMM. Cette dernière exécute des VMs dans l'espace user. Toutes les instructions des VMs sont émulées par la VMM. L'OS de la VM est non modifiés et peut être de différent types (Linux, Windows, etc.).

**Ex.** : VirtualBox

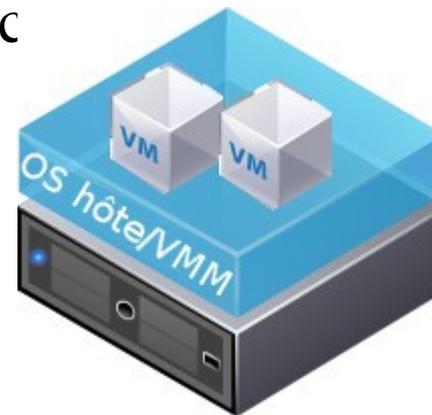


Différents types de systèmes de virtualisation (VMM)

## Virtualisation niveau OS

L'OS hôte dispose des mécanismes pour construire des containers isolés (VMs). Ces derniers partagent le même OS (l'hôte). La VMM fait partie de l'OS hôte.

**Ex.** : openVZ, chroot, LXC, etc



# Virtualisation

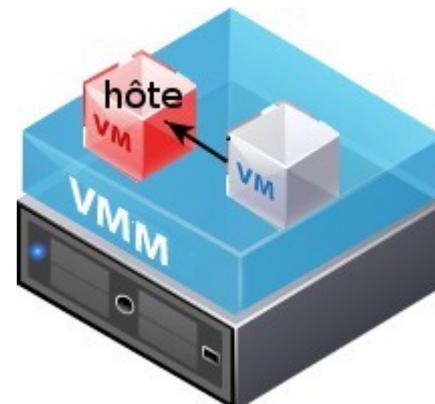
Différents types de VMMs

## Para-virtualisation

La VMM remplace l'OS hôte et sert d'intermédiaire pour discuter avec le matériel. L'OS hôte est lui même considéré comme une VM (privilégiée). Il est utilisé par la VMM pour assurer certaines tâches.

**Contrainte** : Les OS de VMs doivent être modifiés à chaque release (afin d'être au courant de leur situation de machine virtuelle).

Ex. : Xen, VMware, etc.



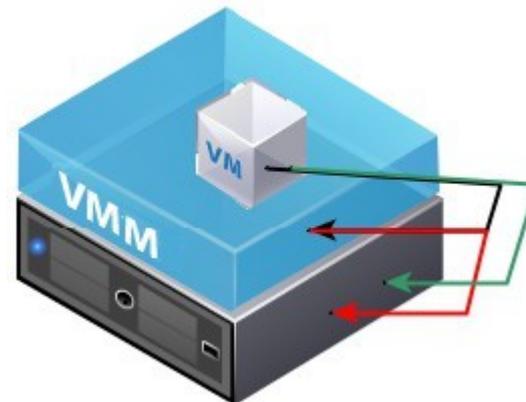
Différents types de VMMs

## Virtualisation assistée par le matériel (HAV)

Para-virtualisation sans intervention de l'OS hôte. Le matériel est au courant de la virtualisation.

Les OS de VMs ne sont pas modifiés.

Ex. : Xen, VMware, etc.



# Virtualisation

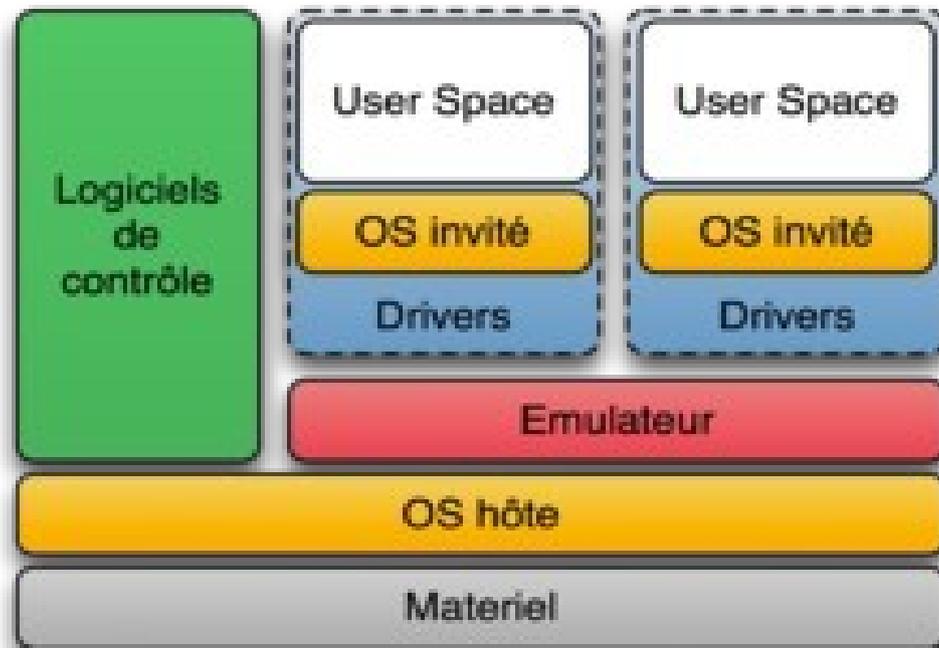
## Hyperviseur

Wikipédia

Un hyperviseur de Type 2 est un logiciel qui s'exécute à l'intérieur d'un autre système d'exploitation. Un système d'exploitation invité s'exécutera donc en troisième niveau au-dessus du hardware (matériel). Les OS invités n'ont pas conscience d'être virtualisés, il n'a donc pas besoin d'être adapté.

## Exemples

- VMWare Player
- QEMU
- VirtualPC
- VirtualBox
- Emulateurs NES, Atari, Amiga ...



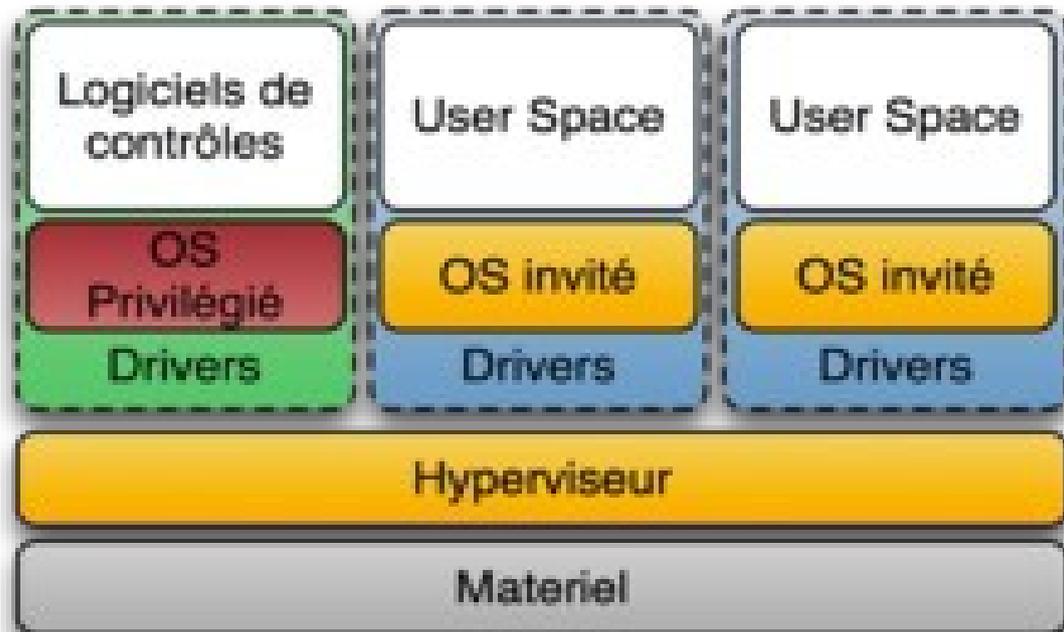
**L'hyperviseur est donc le logiciel qui permet et gère la virtualisation.**

# Virtualisation

Un hyperviseur de Type 1 (ou natif, bare-metal) est un logiciel qui s'exécute directement sur une plateforme matérielle. Un système d'exploitation secondaire peut de ce fait être exécuté au-dessus du hardware. L'hyperviseur type 1 est un noyau hôte allégé et optimisé pour ne faire tourner initialement que des noyaux d'OS invités adaptés et optimisés pour tourner sur cette architecture spécifique, les OS invités ayant conscience d'être virtualisés. Ils doivent être adaptés.

## Exemples

- VMWare ESX
- Xen / Oracle Vm
- Hyper-V
- Linux KVM
- PowerVM
- z/VM



# Virtualisation

## Para-Virtualisation

- Un hyperviseur de type 1 peut être para- virtualisé, dans ce cas, les invités sont modifiés pour ne pas intégrer d'instructions critiques et s'appuient sur l'API de l'hyperviseur.
- On note alors un gain de performance fort car l'hyperviseur n'a pas à intercepter toutes les instructions.
- Par ailleurs, au travers de ces API et de la confiance que l'on a dans cet OS adapté spécifiquement pour la virtualisation, il est possible de lui donner des accès directs à certains périphériques (disque, usb, réseau..)

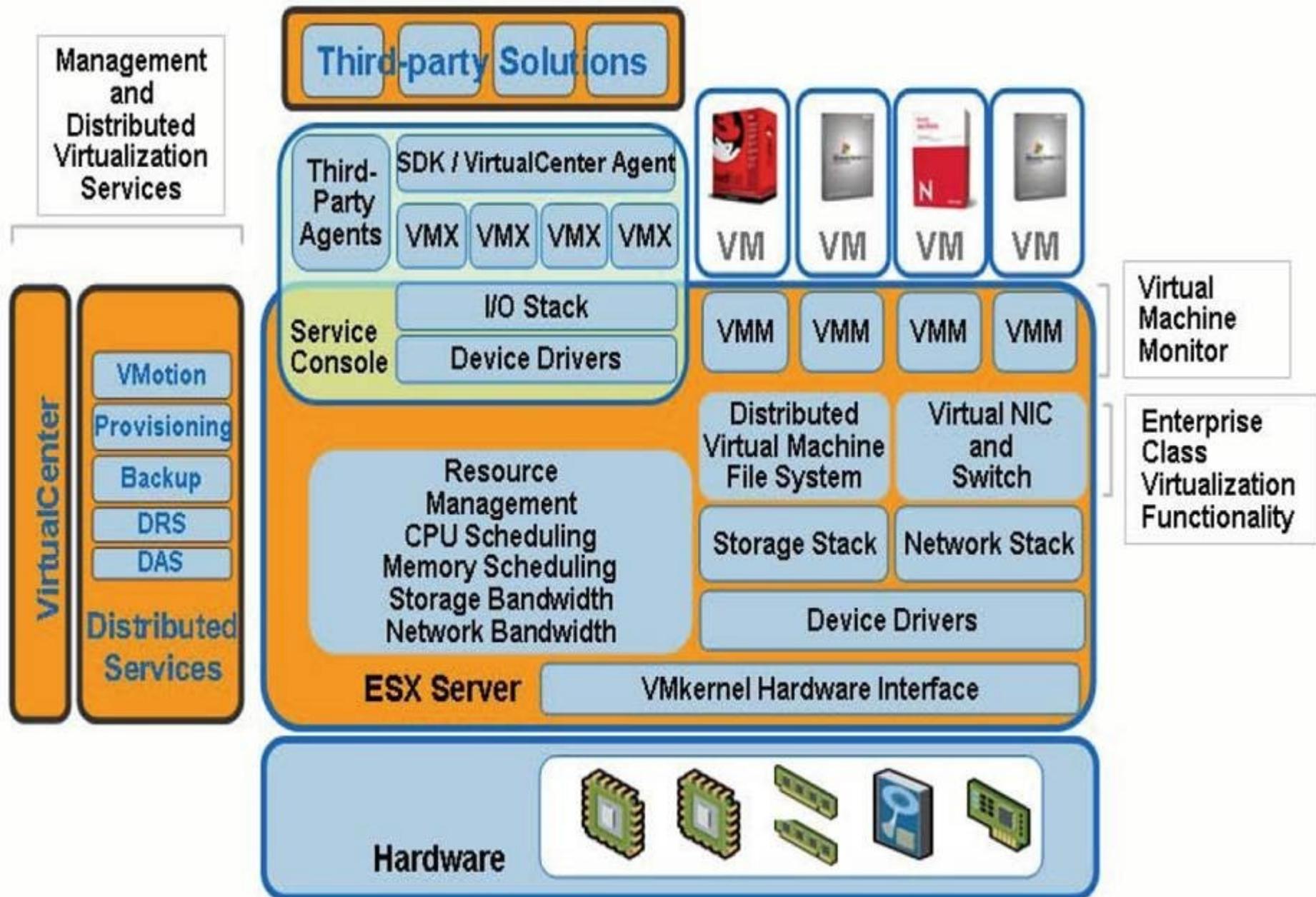
## Systemes para-virtualisés

- XEN
- KVM
- PowerVM
- ESX

## VMWare ESX

- Le VMKernel qui est l'hyperviseur et contrôle les périphériques.
- Au dessus on trouve les piles de drivers partagés entre les différentes VM
- La Vmware Console OS permet l'administration.
- Les VM n'ont pas à être modifiées, elles n'ont pas d'accès direct au matériel.
- Cette solution comprend beaucoup de composants d'administration et outils tiers.
- L'hyperviseur pèse à peu près 200.000 lignes de code.

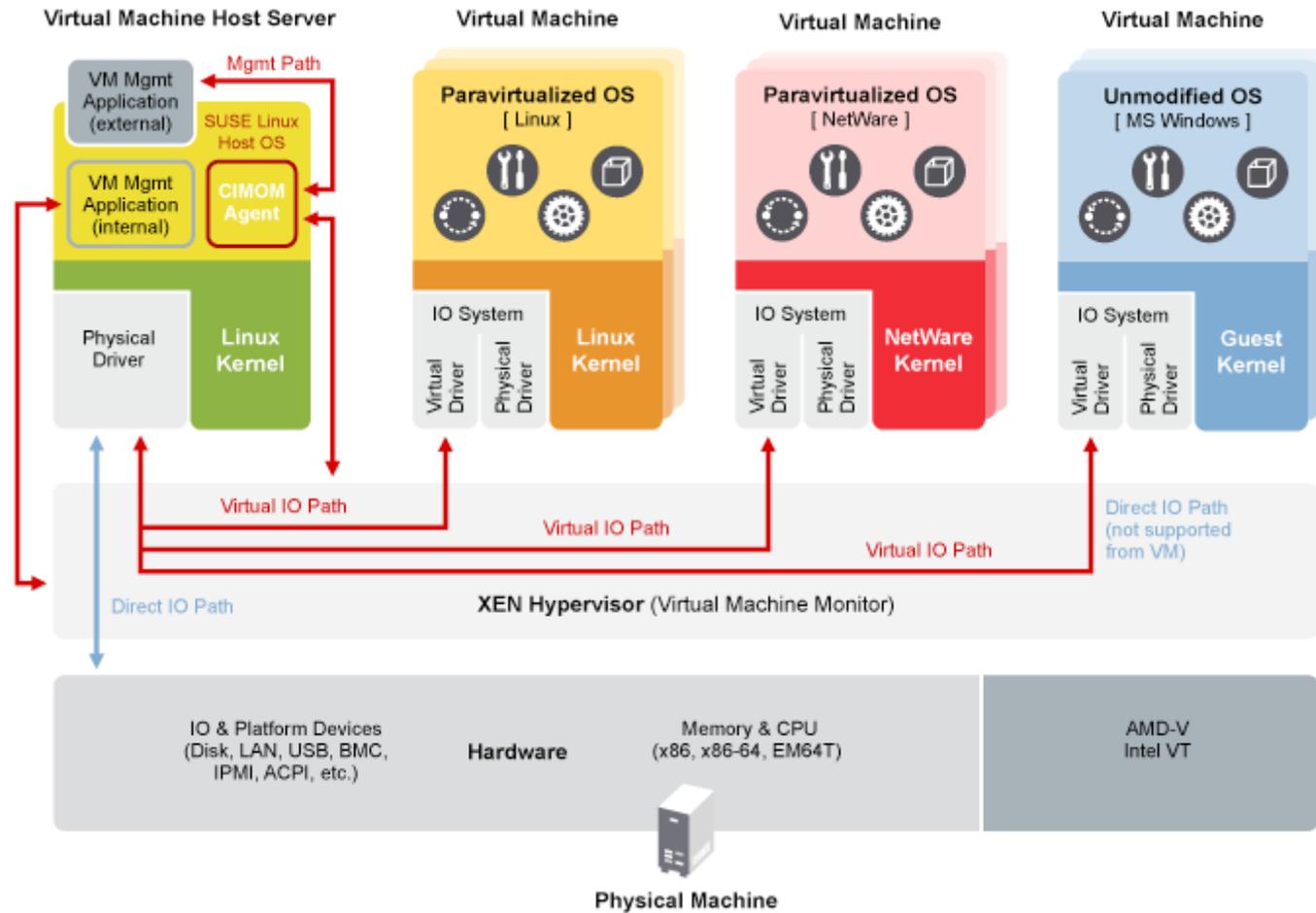
# Virtualisation



## Citrix Xen

- On retrouve la couche hyperviseur entre les matériel et les VM
- Une VM particulière appelée Dom0 permet l'administration des autres VM.
- Les systèmes ont besoin d'être portés pour être utilisés sur Xen
- Les VM peuvent directement accéder au matériel ce qui en fait un système des plus performant
- Un système Windows peut être lancé si le processeur supporte les technologies Intel-VT ou AMD-V. La stabilité est moyenne.
- L'hyperviseur pèse quelques millier de lignes de code.

# Virtualisation



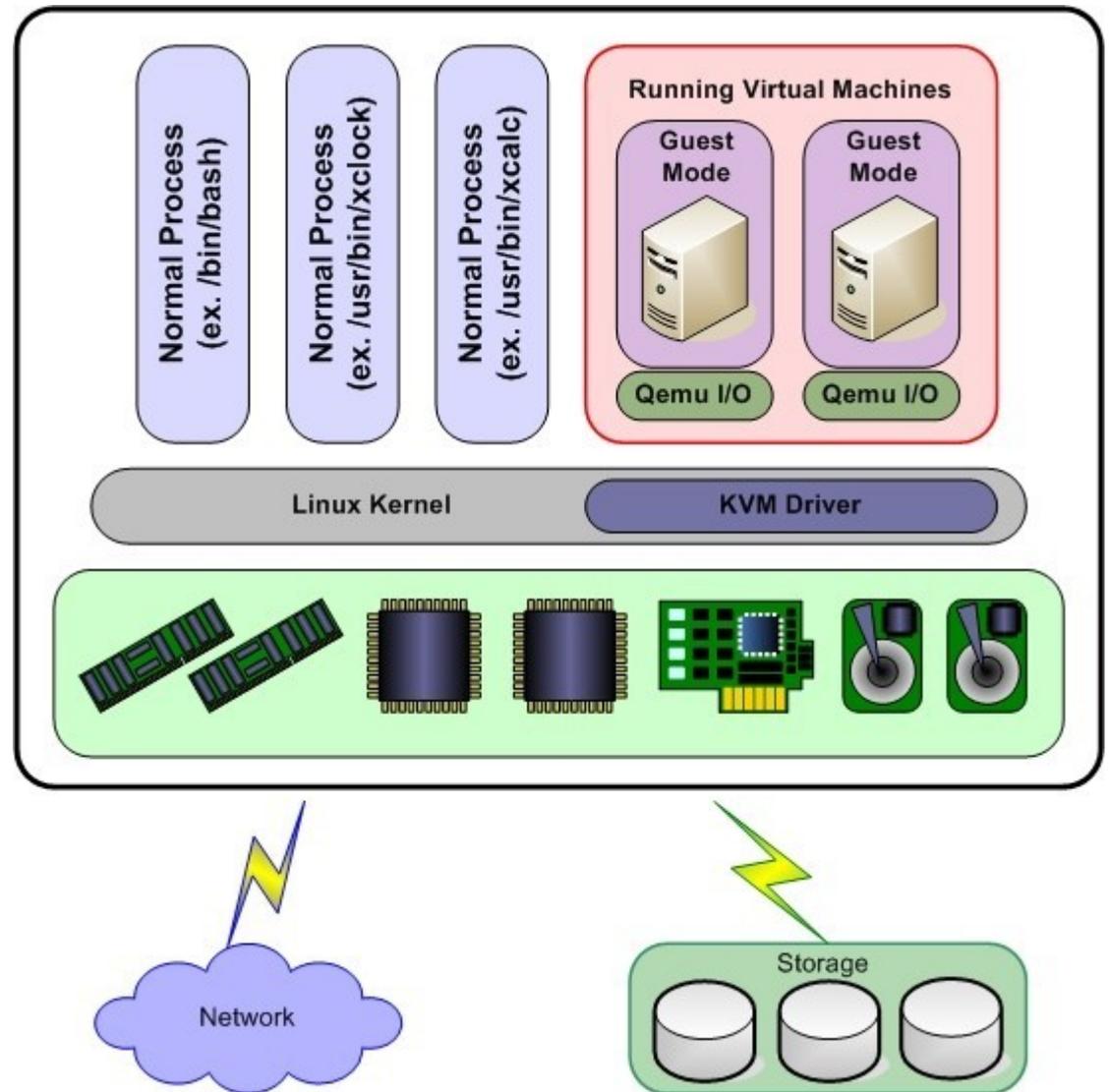
## Linux KVM-QEMU

- › KVM – Kernel-Based VM Module de virtualisation intégré dans le noyau Linux
- › Chaque VM est vue comme un processus du système hôte.
- › KVM ajoute un mode d'exécution en plus de kernel et user : le mode « Guest »

Se compose d'un périphérique (/dev/kvm) pour gérer la virtualisation et d'un composant émulant les périphériques d'un PC : QEMU.

- › La « LibVirt » permet la gestion de l'ensemble. KVM-QEMU profite de tout ce que le noyau linux apporte.

# Virtualisation



## Rôle d'un OS

- Partage le processeur entre plusieurs processus
  - S'appuie sur des mécanismes HW comme les Interruptions
- ➔ Partage la mémoire entre plusieurs processus
  - S'appuie sur des mécanismes HW comme la MMU
- Partage les I/O des périphériques entre plusieurs processus
  - S'appuie sur des mécanismes HW comme les Interruptions

## Rôle de l'hyperviseur

- ➔ Permettre ces mécanismes pour chaque OS invité, lui permettant de croire qu'il est le seul sur la machine.
- ➔ Intercepter les communications avec les interruptions, la MMU...  
et simuler de façon logicielle ces interfaces.

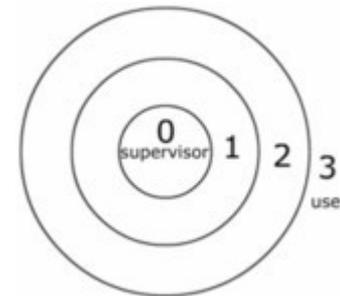
# Rôle de l'hyperviseur

## Niveau d'exécution

Un processeur a plusieurs niveau d'exécution (Ring)

- Ring 0 – Kernel – accès au HW
- Ring 1,2 - Dispo
- Ring 3 – User

- Le kernel en Ring0 peut contrôler les processus utilisateurs en Ring3, l'inverse n'est pas possible
- Normalement, les instructions assembleur pouvant changer la configuration HW ne devraient s'exécuter qu'en Ring0

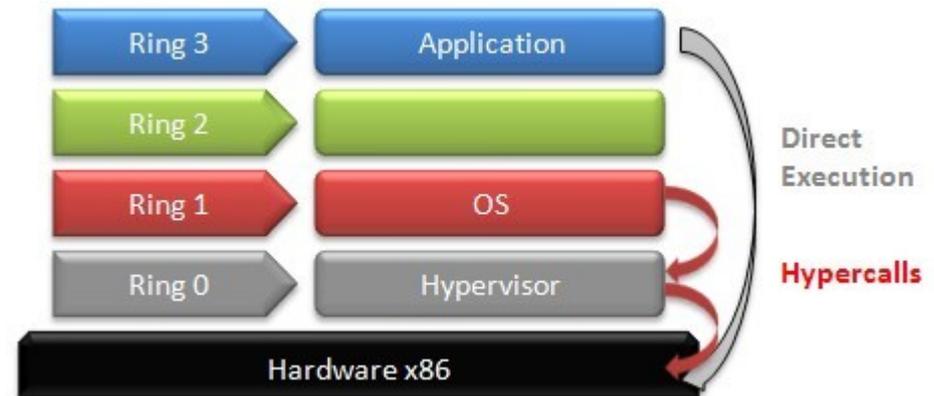


- Instructions critiques
  - Permettent le changement du registre d'horloge ou des interruptions
    - SGDT, SIDT, SLDT, SMSW, PUSHF, POPF
  - Protection du système, des accès mémoire
    - LAR, LSL, VERR, VERW, INT, STR...
- Sur x86, 17 instructions critiques parmi celles ci-dessus s'exécutent en Ring 1,2,3
- Lors de la virtualisation, il est impératif que l'hyperviseur les intercepte et les traite manuellement.
  - Pour les instructions critiques de Ring0 l'interception est simple car leur exécution dans un autre Ring génère une erreur qu'il est possible d'intercepter.
  - Pour les 17 instructions critique de Ring3, il n'y a que deux solutions :
    - Les éliminer (demande une ré-écriture du kernel invité)

# Virtualisation

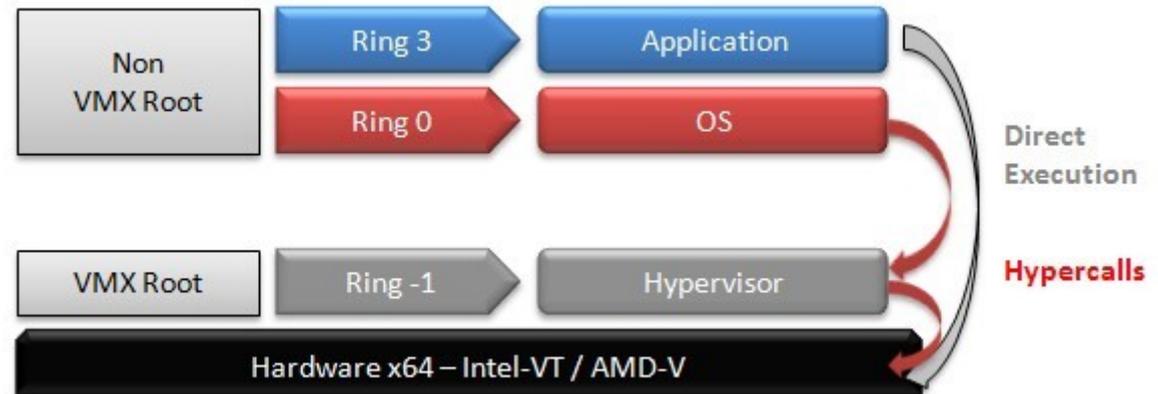
## Application a la virtualisation

- L'hyperviseur est exécuté en Ring0
- Les OS Invités sont exécutés en Ring1
  - Ils doivent passer par l'hyperviseur pour l'exécution des instructions critiques.
  - L'OS ne doit pas découvrir qu'il ne s'exécute pas en Ring0
- Les applications sont toujours exécutées en Ring3



# Virtualisation

- Technologies de virtualisation
  - HardWare
  - Intel-VT
  - AMD-V
- L'hyperviseur est exécuté dans un nouveau Ring -1
- Les VM sont exécutées en Ring0, mais ce Ring 0 est “Non Root”, il peut donc être intercepté par le ring -1.
- Ce Ring-1 embarque de nouvelles instructions facilitant la virtualisation.
- **La Para-Virtualisation n'est plus nécessaire.**



# Virtualisation

- Gestion mémoire
  - -La MMU (Memory Management Unit) est un composant matériel permettant de gérer la mémoire et sa sécurité. Elle permet de faire croire à chaque processus qu'il dispose de toute la mémoire, commençant à l'adresse 0 et de façon continue en effectuant des translations d'adresses automatiques (sans intervention de l'OS)
  - Elle assure la sécurité en empêchant une application de lire ou écrire au delà de la zone mémoire qui lui est attribuée (Segment Violation)
- Les appels à la MMU doivent être interceptés et la MMU doit être simulée au niveau de l'Hyperviseur ce qui crée rapidement une surcharge et une perte de performance.  
L'hyperviseur doit en effet maintenir une table de correspondance intermédiaire (Shadow Page Table)
- Intel-VT et AMD-V implémentent matériellement ces tables intermédiaires (Extended Page Table) ou (Rapid Virtualization Indexing)

# Virtualisation

## Gestion des périphériques

- Les périphériques peuvent être émulés, dans ce cas, le système hôte voit des périphérique standard du marché (carte video Cirrus Logic, carte réseau Realteak) alors que le matériel est tout autre. L'hyperviseur se charge de la conversion. En général, ces périphériques implémentent peu de fonctionnalités (pas de 3D par exemple) et sont peu performants.
- Les périphérique USB sont faciles à partager car le protocole USB peut être rerouté directement vers le système Guest.
- Intel et AMD ont mis en oeuvre des système d'IOMMU permettant un accès direct par les Guest aux périphériques.
  - Cette technologie permet un accès direct et exclusif à un périphérique par un hôte.
  - Cette technologie permet de profiter de toutes les fonctionnalités d'un périphérique et de sa performance nominale.

Le choix du processeur est donc important pour la virtualisation

## Avantages

- Consolidation et utilisation optimale des ressources
- Isolation (ressources, défaillance, espace utilisateur, etc.)
- Migration à chaud (déplacement d'un OS d'une machine à une autre sans interruption de service)
  - Optimisation énergétique
- Gestion améliorée de l'énergie électrique
- Réduction des coûts en refroidissement
- Réduction des coûts d'expansion de la plate-forme Sauvegarde et Reprise après pannes
- Checkpointing d'OS

La VM sert donc d'unité de facturation et d'allocation dans le cloud de type IaaS

## Inconvénients

- Surcout

- 3 % d'overhead avec la VMM Xen par rapport à un OS non virtualisé
- Coût de la migration à chaud
- LAN
- Surtout WAN
- 

## Hétérogénéité

- Coût du checkpointing
- Reprise à partir d'un état antérieur
- Granularité des images de VMs
- VMs à la carte (rapide à construire et à démarrer)

## Inconvénients

- Si la virtualisation simplifie les opérations, elle ne demande pas néanmoins un certain niveau d'administration

Il est nécessaire de suivre les évolutions de consommation des VM et de les déplacer le cas échéant.

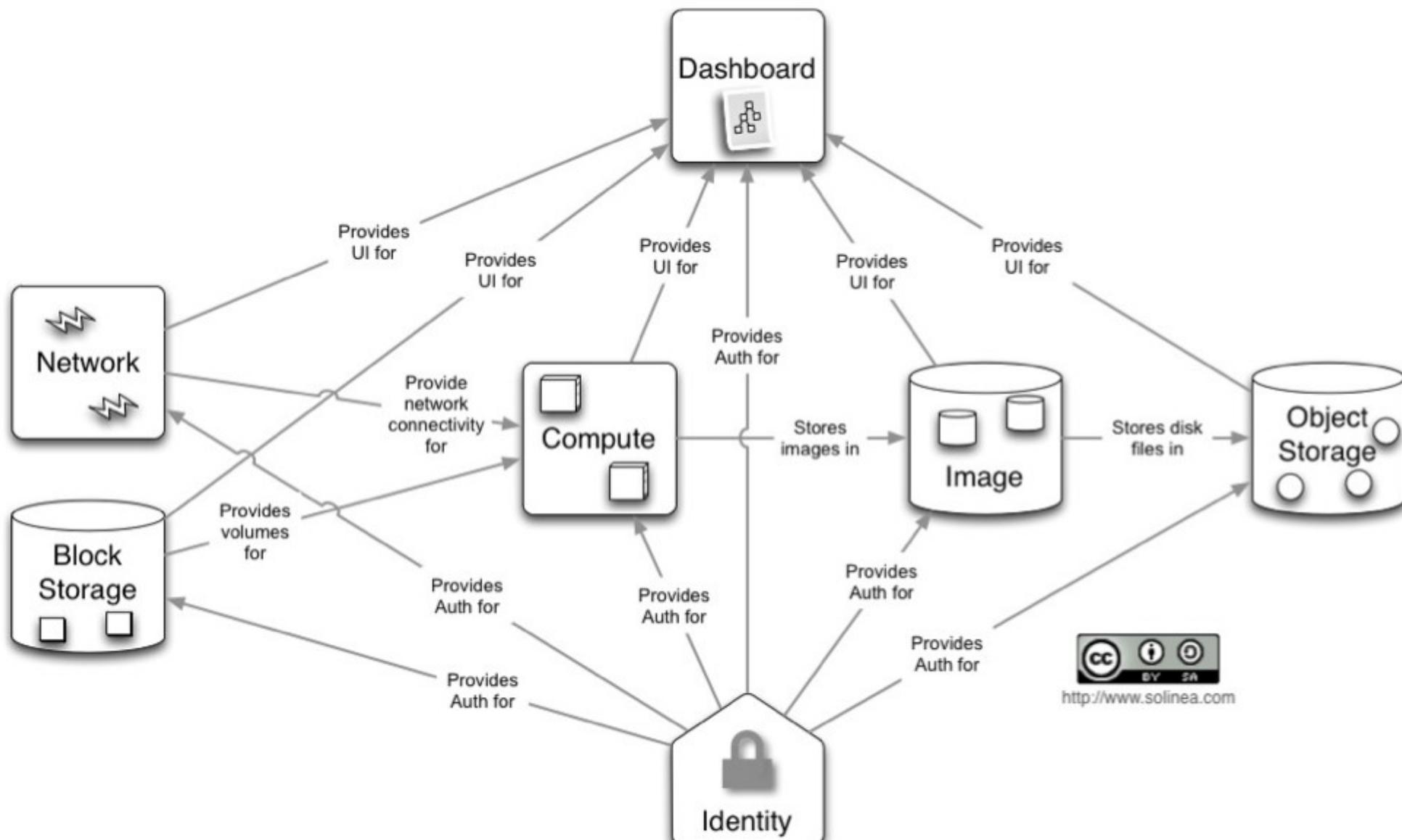
Il est nécessaire de les mettre à jour (sécurité) et plus il y en a, plus il faut automatiser le processus.

Il est nécessaire de gérer la sauvergarde.

Tout ne peut pas être virtualisé

Oracle par exemple refuse la virtualisation (autre que Oracle VM) et demande ainsi de payer les licences pour l'ensemble des

processeurs physiques.





Les principaux composants logiciels d'OpenStack sont les suivants :

Nova qui permet de gérer la partie « Compute » du cloud – à ce titre Nova gère la fourniture et le fonctionnement des VM sur le cloud

Swift, qui fournit les services de stockage en mode objet

Glance, qui fournit la gestion du catalogue d'images système

Cinder, qui fournit les services de stockage en mode bloc

Neutron, qui fournit la gestion des services de virtualisation de réseau

Keystone, qui fournit les services de gestion d'identité – et notamment les mécanismes d'authentification des utilisateurs

Horizon, qui fournit le portail d'administration de l'ensemble.



D'autres composants s'ajoutent chaque semestre à cette liste et viennent progressivement enrichir le framework OpenStack. Parmi les plus importants en cours de développement on peut noter Heat (un moteur d'orchestration à base de templates), Ceilometer (un système de compteur pour la facturation des services OpenStack à la consommation), Ironic (pour le provisioning de serveurs en mode bare-metal) ou Trove (un service de base de données MySQL à la demande pour OpenStack). Côté stockage, EMC, Mirantis et NetApp travaillent conjointement au développement du projet Manila pour permettre le support de services de stockage NFS – et à terme CIFS – par OpenStack



OpenStack ?  
Écrit en Python  
Communication entre les composants  
API Rest et RPC  
AMQP (RabbitMQ ou QPid)  
KVM comme premier choix d'hyperviseur  
LXC pour des tests rapides  
Xen/XenServer (dont XCP)  
VMware/ESX  
Docker  
Baremetal



Les principaux composants

Nova : Gestion des traitements

Responsable de la gestion des instances (lien fort avec la virtualisation)

La gestion des ressources (par des quotas)

Possède la notion de zones de disponibilité

Personnalisables

Pour une réduction des risques

Nova regroupe de nombreux "projets internes" pour assurer ces tâches comme novacompute



Les principaux composants

Swift : le stockage dans le cloud

Permet de gérer un stockage de plusieurs pétaoctets en 1 seul système

Assure la réplication et l'intégrité des données

Stockage de blobs bruts accompagnés d'attributs (nécessité de xattr)

Maintenance d'un mapping entre nom et localisation (the ring)

Notion de zones (similaires à nova)

Personnalisables

Réplication (par défaut) dans 3 zones différentes

Subit la concurrence de Ceph (projet non OpenStack mais très utilisé)



Les principaux composants

Glance : La gestion des images

Stockage et manipulation d'un ensemble d'images (personnalisées)  
utilisables

Lancement rapide

Déjà configurées

Intégrité vérifiée

Nova supporte plusieurs systèmes de virtualisation => plusieurs  
formats d'images supportés par Glance

Stockage multiple

Disques locaux

NAS/iSCSI

Swift



Les principaux composants  
Keystone : La gestion des identités  
Keystone est le point d'entrée dans son cloud  
centralise l'authentification  
liste des services et endpoints  
gère les différents rôles



Les principaux composants

Neutron : La gestion avancée du réseau

Neutron constitue la nouvelle génération de gestion du réseau sur OpenStack

Propose une API unifiée qui est capable de piloter plusieurs architectures réseaux

Linux Bridge

Open vSwitch

Cisco (matériel Nexus)

NSX (VMWare) ...

Permet la création de réseaux complets et autonomes par tenant

Base de fonctionnalités avancées

LBaaS

DNSaaS



Les principaux composants

Cinder : La gestion des volumes

Nouveau paradigme pour les données :

Ephémère si non précisé

Persistant à la demande, on parle de volume

C'est le rôle de Cinder

Cinder est l'équivalent de EBS pour AWS

Permet de stocker sur

Systèmes de fichiers (sur plusieurs nodes)

SAN / NAS

Ceph



Les principaux composants

Heat : L'orchestration

Equivalent de Amazon Cloud Formation

Permet la description d'un déploiement (une stack) d'une infrastructure

Lancement et provisionning d'instances

Gestion des stockages associés

Gestion des IP

Haute disponibilité

Autoscalling (couplé à Ceilometer)

2 Approches

Compatibilité Cloud Formation

Langage de templates HOT



Les principaux composants  
Ceilometer : La gestion des mesures  
Gère les mesures d'utilisation  
=> Le socle pour le billing (non inclu)  
De nombreux agents capable de donner des données ou des cumuls  
La base de l'alarming / autoscalling



Les principaux composants  
Et les autres (à différents stades)  
Horizon (Dashboard web)  
Trove (Base de données as a Service)  
Marconi (Messaging as a Service)  
Oslo (Librairies de code communes)  
Manilla (Systèmes de fichiers partagés)  
Devstack (Déploiement mononode pour prise en main rapide)  
Ironic (Driver Bare Metal)  
TripleO (Déploiement)  
Sahara (Couplage avec Hadoop)



Pourquoi choisir OpenStack ?

Création de cloud privé

Proche d'une solution de virtualisation sur mesure (avec tous les composants et possibilités offertes)

Multi hyperviseurs dans un seul environnement

Gestion fine des lieux d'exécution (ou pas)

Haute disponibilité

Scalable par conception

Load balancing natif

Souplesse/facilité des migrations pour permettre une bonne gestion des hotes

Ajout/intégration simplifié de nouveaux hosts

Décomposition native en projets simplifiant les permissions

Possibilité d'avoir une gestion avancée et fine du réseau

OpenStack s'impose progressivement comme l'un des standards open source les plus populaires pour la constitution de clouds d'infrastructure publics ou privés. La technologie, désormais pilotée par une fondation indépendante, est avec Apache Cloud Stack – à l'origine développé par Citrix et désormais piloté par la fondation Apache – l'une des principales alternatives libres aux technologies de cloud vCloud de VMware et Cloud O

OpenStack est une pile logicielle libre – sous licence Apache 2.0 – qui permet de créer et d'administrer automatiquement de vastes groupes de serveurs virtuels privés ainsi que les ressources réseaux et stockage associées. La technologie s'articule autour de plusieurs composants logiciels majeurs chacun responsable de fournir une fonction précise.



