

Infogérance et virtualisation

Hassen CHEFFI

Institut Supérieur des Études Technologiques de SFAX

November 28, 2024

Objectif du cours

- ▶ Maîtriser les principes de la virtualisation
- ▶ Comprendre les étapes d'une démarche de virtualisation, ses enjeux, ses contraintes
- ▶ Savoir déployer, sécuriser et maintenir une solution de virtualisation

Introduction à la virtualisation

Partage d'un serveur

Un **serveur** est un ordinateur **utilisé à distance** depuis **différents postes de travail**, ou autres serveurs. Il **possède** des **ressources matérielles**, principalement CPU, mémoire, disques et interfaces réseau. Ces ressources sont utilisées par des applications, non pas de manière directe, mais en s'appuyant sur un système d'exploitation.

Type de serveur



Serveur tour



Serveur rack



Serveur lame

Qu'est-ce que la virtualisation ?

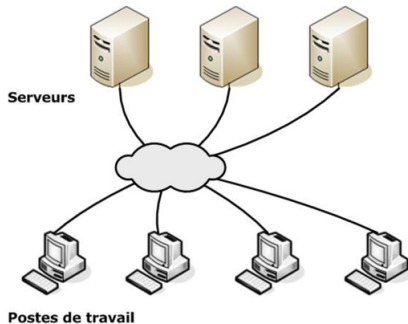
- ▶ La virtualisation de serveurs est un **ensemble de techniques** et d'outils permettant de **faire tourner plusieurs systèmes d'exploitation** sur un **même serveur physique** .
- ▶ Le principe de la virtualisation est donc un **principe de partage** : les différents systèmes d'exploitation se partagent les **ressources du serveur** .

Principes fondamentaux

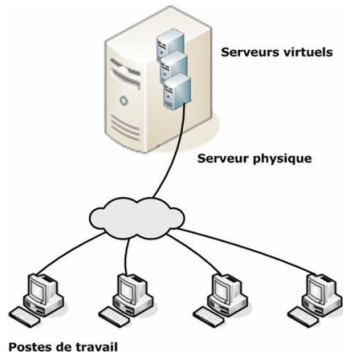
Pour être utile de manière opérationnelle, la virtualisation doit **respecter** deux **principes fondamentaux** :

- ▶ Le **cloisonnement** : chaque système d'exploitation a un **fonctionnement indépendant** , et ne peut interférer avec les autres en aucune manière.
- ▶ La **transparence** : le fait de fonctionner en mode virtualisé ne change rien au fonctionnement du système d'exploitation et a fortiori des applications.
- ▶ La transparence implique la **compatibilité** : toutes les applications peuvent tourner sur un système virtualisé, et leur fonctionnement n'est en rien modifié.

Architecture traditionnelle



Architecture traditionnelle



Architecture virtualisée

Architecture traditionnelle

Il existe depuis longtemps d'autres moyens de partager des ressources physiques. En fait, les applications tournant sur un même serveur, en l'absence de virtualisation, se partagent déjà les ressources du serveur. C'est l'une des missions du **système d'exploitation** que de **permettre** et d'administrer ce **partage** : plusieurs applications se partagent les disques, le processeur, la mémoire, les accès réseau, et le système d'exploitation est le chef d'orchestre, gérant les règles de ce partage.

- . Alors, pourquoi ce partage ne suffit-il pas ?
- . Pourquoi a-t-on besoin de virtualisation ?

Architecture traditionnelle

Les **limites** du partage d'applications au sein d' **un unique système d'exploitation** :

- ▶ Le premier aspect concerne la rigueur du cloisonnement au sein d'un même système entre les différents contextes de travail. Les systèmes natifs souvent **ne fournissent pas un cloisonnement**
- ▶ Des fois, les applications requièrent un **système d'exploitation particulier** , ou bien une **configuration particulière** du système, ou encore des composants logiciels majeurs qui ne peuvent pas cohabiter sur un même système d'exploitation.

Fonctionnement sans utilisation de la Virtualisation

- ▶ **Matériel Dédié**
 - ▶ Chaque application ou serveur fonctionne sur son propre matériel dédié.
 - ▶ Exige une allocation spécifique de ressources matérielles pour chaque service.
- ▶ **Inefficacité des Ressources**
 - ▶ Le matériel est souvent sous-utilisé, car chaque serveur ou application a ses propres ressources allouées
- ▶ **Complexité de la Gestion**
 - ▶ La gestion des serveurs physiques individuels peut devenir complexe, surtout dans des environnements avec de nombreuses applications.
- ▶ **Évolutivité Limitée**
 - ▶ L'ajout de nouvelles applications ou la modification des configurations peut nécessiter l'achat de nouveau matériel
- ▶ **Reprise d'Activité Limitée**

Fonctionnement sans utilisation de la Virtualisation

Inconvénients de l'Approche sans Virtualisation :

- ▶ Sous-utilisation des Ressources
- ▶ Coûts Élevés
- ▶ Difficulté de Gestion
- ▶ Rigidité et Manque de Flexibilité
- ▶ Complexité de la Reprise d'Activité

Bénéfices de la Virtualisation

- ▶ Usage optimale des ressources
- ▶ Installation, déploiement et migration facile
- ▶ Économie sur le matériel
- ▶ Sécurisation
- ▶ Isolation
- ▶ Allocation dynamique
- ▶ Diminution des risques (fonctionnalités telles que la **sauvegarde** et la **restauration rapides** à partir de **snapshots** , facilitant ainsi la reprise après incident)

Inconvénients de la Virtualisation

- ▶ **Complexité** : La virtualisation introduit une couche d'abstraction supplémentaire, ce qui peut rendre la gestion et la maintenance des infrastructures plus complexes.
- ▶ **Besoin de Ressources Élevé** : Les hyperviseurs consomment des ressources matérielles, ce qui peut entraîner une surcharge sur le serveur hôte.
- ▶ **Dépendance à l'Égard de l'Hyperviseur** : Les machines virtuelles dépendent de l'hyperviseur, et tout problème avec celui-ci peut avoir un impact sur l'ensemble des machines virtuelles.
- ▶ **Coût Initial** : Bien que la virtualisation puisse entraîner des économies à long terme, la mise en place initiale peut nécessiter des investissements importants

Historique

- ▶ Idée développée au centre IBM de Cambridge et de Grenoble en 1972 (VM/CMS) (pseudo-machine.)
- ▶ Mi-90's émulateurs d'Atari, Amiga, NES, SNES,...
- ▶ Début des années 2000 : VMware
- ▶ Logiciels libre : Xen, Qemu, Bochs,...
- ▶ Propriétaire (mais gratuits) : VirtualPC, VirtualBox, etc.

Terminologie

- ▶ **Hyperviseur** (ou VMM – Virtual Machine Monitor) : Logiciel qui permet la création et la gestion de machines virtuelles.
- ▶ **Machine Virtuelle** (VM) : Une instance virtuelle d'un ordinateur complet, comprenant un système d'exploitation, des applications et des ressources matérielles émulées ou virtuelles.
- ▶ **Hôte** (ou Serveur Hôte) : Le serveur physique sur lequel l'hyperviseur est installé et qui exécute les machines virtuelles.
- ▶ **Invité** (ou Machine Virtuelle Invitée) : Une machine virtuelle qui fonctionne sur un hôte. Elle est souvent utilisée pour exécuter des applications ou des services spécifiques.
- ▶ **Snapshot** : Une copie instantanée de l'état actuel d'une machine virtuelle. Les snapshots permettent de revenir à un état précédent en cas de besoin.

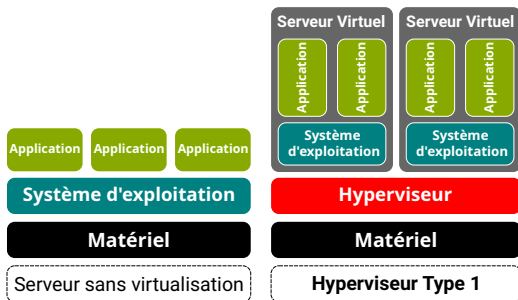
Type d'hyperviseur

Il existe deux principaux types d'hyperviseurs, également appelés superviseurs de virtualisation. Ces types sont classés en fonction de leur relation avec le matériel physique et la manière dont ils gèrent les machines virtuelles (VM) :

- ▶ Hyperviseur de Type 1 (ou Bare-Metal)
- ▶ Hyperviseur de Type 2 (ou Hosted)

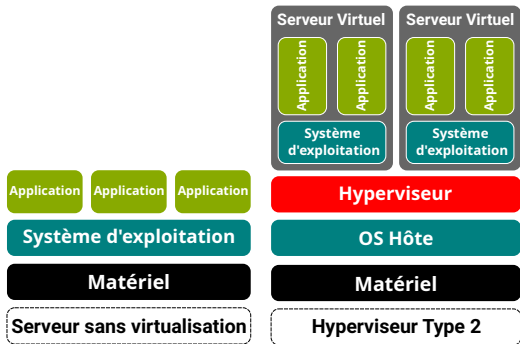
Hyperviseur de Type 1 (ou Bare-Metal)

- ▶ **S'exécute** directement sur le **matériel physique** sans nécessiter un système d'exploitation hôte.
- ▶ Offre des **performances optimales** et une **isolation** plus **robuste**.
- ▶ Exemples : VMware ESXi, Microsoft Hyper-V Server, KVM (Kernel-based Virtual Machine).



Hyperviseur de Type 2 (ou Hosted) :

- ▶ **S'exécute** comme une **application sur** un **système d'exploitation hôte** existant.
- ▶ Peut être installé sur un système d'exploitation traditionnel. Convient généralement pour le développement, les tests...
- ▶ Exemples : VMware Workstation, Oracle VirtualBox.



Les différents types de la Virtualisation

Il existe différents types de virtualisation:

- ▶ **Virtualisation complète** (ou Virtualisation matérielle)
- ▶ **Paravirtualisation**
- ▶ **Les Isolateurs**

Virtualisation complète (ou Virtualisation matérielle)

- ▶ **Description** : Permet à un hyperviseur (ou moniteur de VM) de créer et de gérer plusieurs VMs, chacune exécutant un système d'exploitation complet.
- ▶ **Fonctionnement** : L'hyperviseur crée une couche d'abstraction entre les VMs et le matériel physique. Chaque VM dispose d'un ensemble complet de ressources virtuelles, y compris un émulateur de processeur.
- ▶ **Limitations** : Ne permet de virtualiser que des systèmes d'exploitation prévus pour la même architecture matérielle que le processeur physique de l'ordinateur hôte. Par exemple, un ordinateur équipé d'un processeur Intel x86 sera incapable de virtualiser un système d'exploitation prévu pour fonctionner dans une architecture PowerPC.
- ▶ **Exemple** : VirtualBox, vmware Workstation...

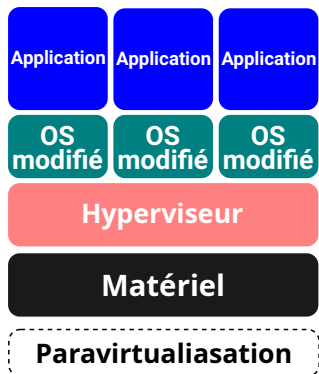
Virtualisation complète (ou Virtualisation matérielle)



Paravirtualisation

- ▶ **Description** : Contrairement à la virtualisation complète, la paravirtualisation nécessite une modification du système d'exploitation invité pour qu'il soit conscient de son statut de machine virtuelle. Les systèmes d'exploitation invités sont modifiés pour utiliser des appels système spécifiques qui interagissent directement avec l'hyperviseur.
- ▶ **Fonctionnement** : L'hyperviseur fournit une interface paravirtualisée, et les systèmes d'exploitation invités modifiés utilisent cette interface pour communiquer avec l'hyperviseur de manière plus efficace que la virtualisation complète.
- ▶ **Exemple** : Xen

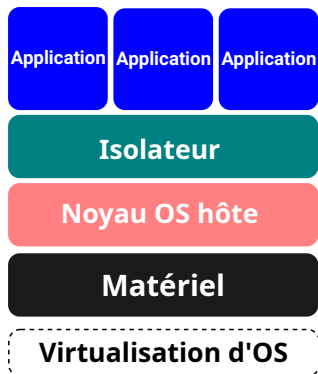
Paravirtualisation



Isolateurs (ou Conteneurs)

- ▶ **Description** : Les isolateurs, souvent utilisés dans le contexte de la virtualisation légère, isolent des environnements d'exécution distincts (conteneurs) sur un même système d'exploitation hôte. Chaque conteneur partage le même noyau du système d'exploitation, mais a son propre espace utilisateur isolé.
- ▶ **Fonctionnement** : Les conteneurs sont des instances légères et portables qui encapsulent une application et ses dépendances. Ils partagent les ressources du système hôte tout en étant isolés les uns des autres. La virtualisation légère permet une utilisation plus efficace des ressources par rapport à la virtualisation complète.
- ▶ **Exemple** : Docker, Lxc

Isolateurs (ou Conteneurs)



Gestion de VMs

- ▶ Vmware Vsphere, Citrix Xen permettent de gérer quelques serveurs physiques
- ▶ Vagrant: Gestion de VMs indépendamment des hyperviseurs
 - ▶ Notion d'images (boxes de Vagrant)
 - ▶ Configuration automatique de VM: support de Puppet, Chef, Ansible
 - ▶ Un fichier qui contient toute la configuration

Gestion de containers

- ▶ Orchestration de containers
 - ▶ Serveur unique: Docker, LXC
 - ▶ Serveur multiple: Docker swarm
- ▶ Orchestration avancés : Kubernetes, Swarm, Mesos

Domaines d'application de la virtualisation

- ▶ Virtualisation d'applications (du contexte d'exécution)
- ▶ Virtualisation de poste de travail
- ▶ Virtualisation de serveur
- ▶ Virtualisation du réseau (VLAN)
- ▶ Virtualisation du stockage

Virtualisation d'applications (du contexte d'exécution)

La virtualisation d'applications consiste à **isoler une application** et ses dépendances du **système d'exploitation** sous-jacent, permettant ainsi son exécution dans un environnement virtuel. Cela **facilite le déploiement**, la gestion et la **compatibilité** des applications sur différents systèmes.

Virtualisation de poste de travail (VDI)

La virtualisation de poste de travail implique la création d'environnements de bureau virtuels, où **les postes de travail** sont **hébergés** sur des **serveurs centraux** plutôt que sur des machines physiques locales. Les utilisateurs **accèdent** à leur **bureau virtuel à distance** , ce qui simplifie la gestion des logiciels et des mises à jour.

Virtualisation de serveur

La virtualisation de serveur **consiste** à **créer** des **machines virtuelles** (VM) sur un **serveur physique** . Chaque VM fonctionne comme un serveur indépendant avec son propre système d'exploitation et ses applications. Cette approche permet **d'optimiser** l'utilisation des **ressources matérielles** en **consolidant plusieurs serveurs virtuels** sur **une seule machine physique** .

Virtualisation du réseau (VLAN)

La virtualisation des réseaux fait référence à la création de **réseaux virtuels** au-dessus d'une infrastructure physique. Elle permet de découpler les fonctionnalités du réseau (comme les routeurs, les commutateurs, les pare-feu, etc.) du matériel sous-jacent en les exécutant **sous forme de logiciels** sur des serveurs virtualisés.

Virtualisation du stockage

La virtualisation du stockage **consiste** à **regrouper** et à **gérer les ressources de stockage** à partir de **plusieurs dispositifs de stockage physiques** . Cela permet de créer des **pools de stockage virtuels** , **facilitant** la **gestion** , **l'allocation dynamique** des ressources, et **améliorant** la **disponibilité des données** .