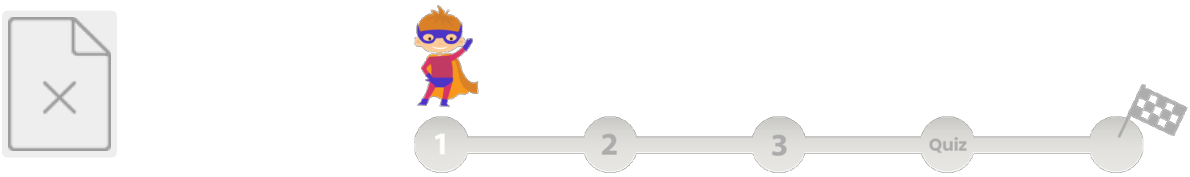
1 - Au cœur de la virtualisation : l'hyperviseur



1. Au coeur de la virtualisation : l'hyperviseur

1.1 Principe de virtualisation

En informatique, la **virtualisation** consiste à **exécuter** sur une **machine physique** que l'on appelle un **« host »** ou **hôte**, **un ou plusieurs systèmes d'exploitation** (machine virtuelle – VM pour Virtual Machine) que l'on nomme **« guest »** ou **invité**.

On retrouve souvent les **hyperviseurs de machines virtuelles** sous le nom de **serveur privé virtuel** (**VPS** pour Virtual Private Server) mais également d'**environnement virtuel** (**VE** pour Virtual Environment).

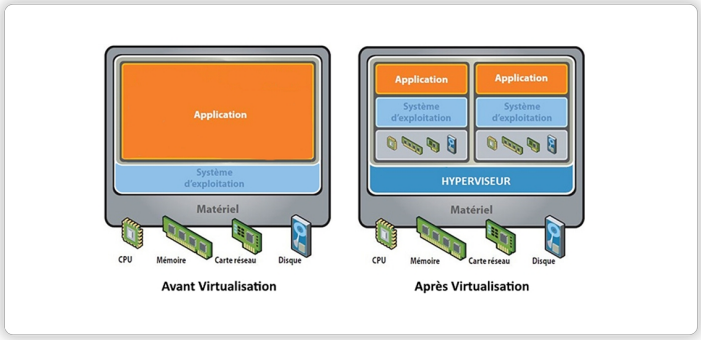


Figure 1 : Schéma d'une architecture avec ou sans virtualisation

Ces machines sont **totalement isolées** les unes des autres. Un **programme** qui s'**exécute** sur une **machine virtuelle** ne peut **pas lire** ou **modifier** le contenu d'**une autre machine virtuelle**.

De la même façon, une défaillance logicielle sur l'une des machines virtuelles n'aura aucun effet sur les autres.

Une machine virtuelle est souvent composée d'un **fichier descriptif** (.conf ou .xml pour qemu ou .vmx pour VMware) et d'un **fichier de disque dur** (.qcow2 pour qemu et .vmdk pour VMware).
Il existe de **nombreux outils** permettant de **convertir** les fichiers d'un format vers un autre.

1.2 L'histoire de la virtualisation

La virtualisation telle qu'on la connaît aujourd'hui a dû franchir plusieurs étapes significatives avant d'émerger dans les années 80.

a. Virtualisation des processeurs

La première étape de la virtualisation consistait à aménager les processeurs afin que ces derniers puissent **effectuer plusieurs tâches en parallèle**. Le mécanisme d'interruption permettant de changer la tache active d'un processeur posait des problèmes de synchronisation entre processus. C'est à Edsger Dijkstra que l'on doit les avancés significatives qui ont été effectuées dans ce domaine dans les années 70 notamment avec les sémaphores qui apporte des

mécanismes de blocage entre processus.

b. Virtualisation de la mémoire

La mémoire des processeurs était très coûteuse au début des années 70, il fallait donc multiplier les allers-retours entre les supports de stockage et la mémoire interne. Le concept de mémoire virtuelle est amené en 1961 permettant de faire concorder la capacité d'adressage d'un processeur à une grande quantité de mémoire.

Il était ainsi facile de **faire concorder** la **mémoire physique des processeurs** avec celle des **systèmes de stockage**.

c. Virtualisation des Systèmes d'Exploitation

Maintenant que le processeur et la mémoire sont utilisables de façon non séquentielle, il faut encore créer une interface permettant d'allouer les ressources physiques de la machine aux systèmes d'exploitation qui seront virtualisés. **C'est IBM qui a introduit le concept à la fin des années 70.**
La réalisation d'un système de machines virtuelles posait cependant toujours un problème. En effet, chaque machine virtuelle étant la copie de machine physique, il fallait encore unifier l'interface qui gérait l'accès aux ressources. Or, cette interface comportait des instructions particulières dites « privilégiées », qui utilisaient les fonctionnalités de base des machines physiques telles que le système d'interruptions, les commandes d'entrées-sorties ou encore la mémoire virtuelle.

La prochaine étape fut donc la **création d'un hyperviseur** afin de créer une nouvelle interface de façon à **homogénéiser la communication** entre l'accès aux ressources bas-niveau de la machine et tous les futurs systèmes d'exploitation qui fonctionneraient en mode invité.

1.3 Hyperviseur

Il existe **deux types** d'hyperviseurs :

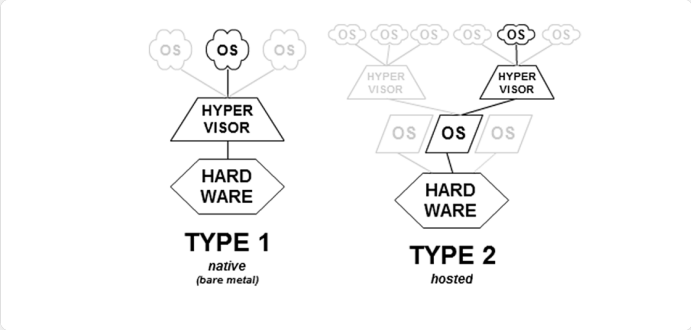


Figure 2 : Différence entre les deux types d'hyperviseurs

a. Hyperviseurs de Type 1 ou « bare metal »

Ce type d'**hyperviseur** communément appelé **« bare metal ou natif »** est en réalité un logiciel spécialisé qui s'**exécute directement** sur une **plateforme matérielle** : il n'a **pas besoin de système d'exploitation existant**.

D'autres systèmes d'exploitation peuvent alors être exécutés par-dessus l'hyperviseur qui va distribuer les ressources physiques de la machine aux instances virtuelles.
L'hyperviseur type 1 est un **noyau hôte allégé et optimisé**. Il existe sur certaines gammes de processeurs des instructions spéciales permettant d'optimiser le fonctionnement des hyperviseurs.
On retrouve AMD-V chez le fabricant de AMD et Intel VT chez Intel.

Voici une liste non exhaustive des hyperviseurs de type 1 les plus fréquents : Xen, Oracle VM, ESXI Server de VMware, Hyper-V de Microsoft.

Les machines virtuelles utilisant un noyau Linux KVM, qui transforment un noyau Linux complet en hyperviseur, sont également considérées comme hyperviseurs de type 1.

b. Hyperviseurs de Type 2 ou « hosted »

Cet autre type hyperviseur est en réalité un **logiciel** qui s'**exécute au-dessus d'un système d'exploitation déjà installé** comme Windows, macOS ou GNU/Linux. Un **système d'exploitation invité** viendra donc s'**exécuter à un niveau supérieur** et non pas au au-dessus du matériel comme un hyperviseur de type 1.

Les systèmes d'exploitation invités n'ayant pas conscience d'être virtualisés, ils n'ont pas besoin d'être adaptés.

Voici quelques exemples de tels hyperviseurs : VMware Workstation, VMware Fusion, l'hyperviseur open source QEMU, Virtual PC et Virtual Server, VirtualBox d'Oracle, de même que Parallels Workstation de SWsoft et Parallels Desktop.

1.4 Avantages de la virtualisation

La virtualisation s'est vite imposée comme standard depuis les années 2000. L'explosion de la bulle internet a vu naître la création de géants de l'internet comme Amazon ou Google.

Ces derniers ont dû **adapter leur infrastructure** pour **faire face à un trafic toujours grandissant**.

Ils n'y seraient jamais arrivés sans la **virtualisation**.

a. La consolidation des infrastructures

La **virtualisation** leur a permis de **consolider des parcs de serveurs** partout dans le monde.

Ils n'étaient pas plus obligé de déployer un serveur par application, mais au contraire, ils pouvaient **préprovisionner des centaines de serveurs** avec des **hyperviseurs** pour les mettre en attente jusqu'au jour où la demande aurait encore augmenté. Cela leur a également permis de **rationaliser les serveurs existants et à venir**.

Imaginez que vous travaillez pour **Google** dans les années 2005 : vous possédez alors un **parc de 250 000 machines** à travers le monde et votre **croissance** vous prévoit un parc de **2 millions de machines** pour 2008. Vous devez gérer un parc qui va doubler tous les ans sur les 3 prochaines années. Vous avez **10 à 15 marques différentes pour vos serveurs** et au moins **4 générations de processeurs** à **maintenir**.

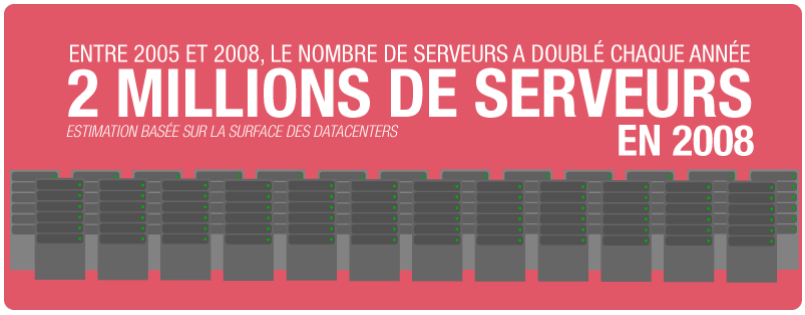


Figure 3 : Estimation de la croissance des serveurs chez Google

b. S'affranchir du hardware

La **virtualisation** vous permettra de **vous affranchir complètement de la partie hardware**.

Vos développeurs n'auront ainsi pas besoin de connaître l'architecture de vos serveurs. Si leur application nécessite 2

CPU et 4G de RAM, peu importe où vous la déploierez sur votre parc de VM, que ce soit un Xeon 3060 avec 2 cœurs à 2,40 GHz et 16 Go de RAM ou le dernier Xeon E7460 avec 6 cœurs à 2,66 GHz et 64 Go de RAM.

Cela vous permettra également d'uniformiser votre parc informatique, peu importe l'âge de vos serveurs : ils ne seront considérés que comme un pool de ressource disponible CPU / RAM / Disque.

c. Environnement cross plateforme

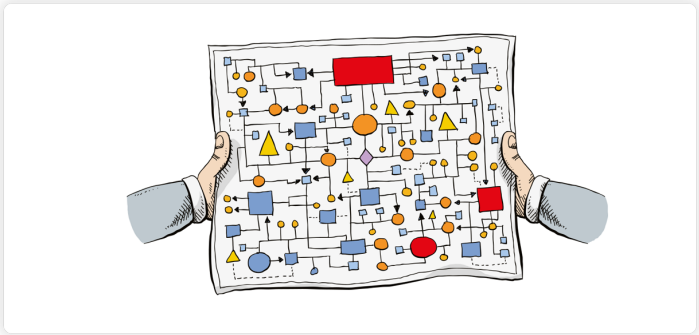
Vous aurez la possibilité d'installer **plusieurs systèmes d'exploitation différents** Windows / Linux sur une même machine.
Vous augmenterez votre résilience, car une machine virtuelle peut être déplacée d'un serveur physique vers un autre.

d. Mutualisation des ressources

En réduisant le nombre de serveurs grâce à la mutualisation des ressources qu'apporte la virtualisation, vous serez capable d'**administrer un plus grand nombre de serveurs** avec les ressources humaines que vous aviez. Vous pourrez également réduire votre facture d'électricité.

1.5 Augmentation de la complexité

Malgré tous les avantages qu'amène la virtualisation, il faut garder en tête qu'elle apporte un **lot de complexité supplémentaire**.



a. Augmentation significative des coûts.

Afin de faire fonctionner convenablement une architecture virtualisée, l'entreprise doit **investir** dans **un ou plusieurs serveurs physiques** disposant de **plusieurs processeurs** et de **beaucoup de mémoire** qui sont souvent plus coûteux que plusieurs petits serveurs.

b. De la Haute Disponibilité pour prévoir les pannes

Si un serveur physique tombe en panne, toutes les machines virtuelles qu'il contient tombent et deviennent indisponibles.
Afin de se prémunir de ce genre de désagrément, tous les hyperviseurs dignes de ce nom ont implémenté des **mécanismes de migration à chaud** de machine virtuelle.
Malheureusement, ce genre de fonctionnalité entraîne souvent des **coûts supplémentaires**.

c. Une vulnérabilité généralisée

Si l'hyperviseur contient une faille de sécurité, les machines virtuelles qu'il contient risquent également d'avoir ce problème. En augmentant les couches logicielles, la virtualisation a pour conséquence d'augmenter la **surface d'attaque** de l'entreprise.