Virtualisation de Serveur

EFORT
http://www.efort.com

1. Introduction

La virtualisation a permis d'accompagner la croissance exponentielle de l'Internet en permettant de se dédouaner de plusieurs contraintes des serveurs physiques. Dès ses débuts dans les années 2000, la virtualisation a permis d'augmenter la densité d'occupation des serveurs en augmentant le taux d'exploitation des ressources disponibles. Cela a permis d'éviter d'avoir des serveurs surdimensionnés consommant beaucoup d'énergie tout en étant très peu occupés. Le but de ce tutoriel est de présenter la virtualisation de serveur avec les concepts de machine virtuelle et de conteneur, d'introduire le cloud et ses services et de décrire le réseau qui permet de relier les différents serveurs des datacenters virtualisés/cloud.

2. Serveur

Un serveur, comme la majorité des ordinateurs, est un ensemble de ressources matérielles spécialisées qui sont accédées par un système d'exploitation (OS, Operating System) via un ensemble de drivers spécialisés. Ces ressources peuvent être variées, mais consistent communément en les éléments suivants (Figure 1);

- Processeur CPU (Central Processing Unit): réalise la partie traitement
- RAM (Random Access Memory ou mémoire à accès direct): stocke et empile les instructions cours terme.
- Stockage: mémoire des données long terme.
- Cartes d'interface réseau (NIC). Permet à la machine de se connecter à d'autres machines ou d'autres équipements via un réseau.

L 'OS communique avec les pilotes (drivers) afin d'accéder aux ressources du serveur; une relation 1-à-1 existe entre eux afin qu'ils constituent un ensemble (matériel, drivers et OS). Une fois que l 'OS et les drivers sont chargés, le matériel est verrouillé.
Puis arrive l'application. L'application est chargée sur l 'OS, devenant ainsi enfermée dans l 'OS, les drivers et le matériel. Une fois chargée, tout est prêt ; le travail du serveur a été défini. L'exécution de cette application est la seule tâche du serveur.
La figure montre la relation entre les ressources matérielles, l 'OS et l'application qui s'exécute sur le serveur.
Compte tenu de toutes ces ressources excédentaires du serveur, pourquoi a t'il été conçu de cette façon? Il se trouve que quand l'informatique a démarré, cette approche était adaptée, parce qu'à l'époque les capacités de calcul étaient beaucoup plus faibles (donc les taux d'utilisation étaient beaucoup plus élevés par serveur) et il existait bien moins d'applications d'entreprise. Les problèmes rencontrés aujourd'hui n'existaient pas à l'époque, et sont plutôt apparus au fil du temps.

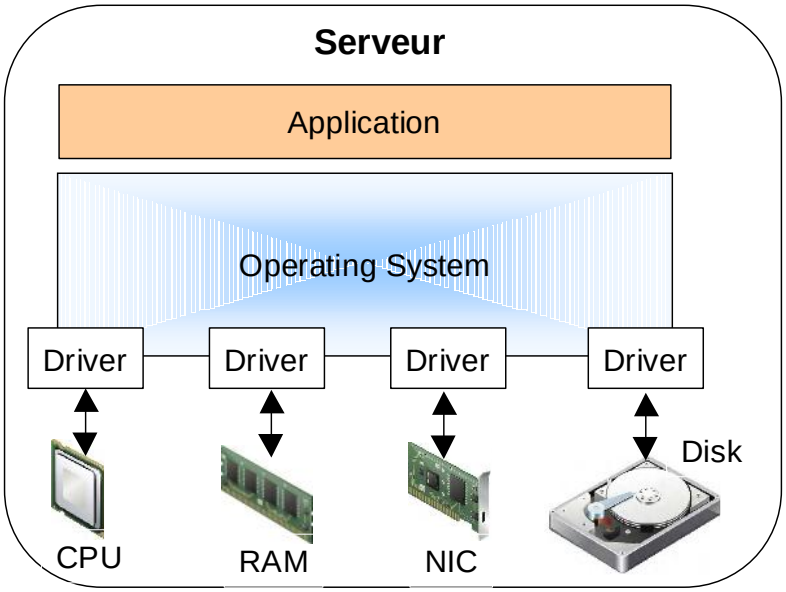


Figure 1 : Ressources matérielles, OS et application

3. Serveur virtualisé et machine virtuelle

Il faut noter dans cette description que l'application et l 'OS sont toujours associés, mais ils ont maintenant été découplés du matériel serveur.
Notons qu'auparavant il était très difficile de déplacer un système d'exploitation. La raison est que l 'OS était très lié au serveur parce qu'il pouvait uniquement accéder aux ressources du serveur via des pilotes qui permettaient l'accès au matériel. Parce que ces pilotes ont été écrits pour les systèmes d'exploitation spécifiques, les serveurs étaient en mesure d'exécuter une seule application.
Il est toujours nécessaire d'accéder aux ressources, ce qui est réalisé via les pilotes.
À haut niveau, une machine virtuelle (VM, Virtual Machine) crée une version logicielle d'un serveur qui s'exécute dans le serveur matériel. Cette VM est une copie exacte du serveur physique. De multiples VMs peuvent s'exécuter sur le même serveur physique, et peuvent toutes être adaptées aux besoins d'application. Un serveur physique peut maintenant mixer ensemble les applications Windows et Linux sur le même matériel.
En outre, parce que VMs sont des logiciels, non seulement il est possible de les exécuter sur n'importe quel serveur, mais il est aussi possible de suspendre ou geler une VM et la déplacer sur un autre serveur à n'importe quel moment, et ceci est censé être rapide et aisé. Il est aussi possible de réaliser une copie d'une VM ou encore un ensemble de copies et les faire tourner sur plusieurs serveurs.
La figure 2 illustre le concept d'une VM qui n'est pas liée au serveur ou au matériel sur ce serveur.

Comment cela fonctionne-t-il avec une machine virtuelle (VM, Virtual Machine) ?
L 'hyperviseur est l'élément essentiel pour l'exécution de VM. Un hyperviseur est un logiciel qui se situe entre le système d'exploitation et les ressources matérielles spécialisées du serveur. Une fois en place, c'est l'hyperviseur, plutôt que le système d'exploitation, qui gère les connexions entre les pilotes et les ressources du serveur.
En rajoutant une couche, comment cela peut conduire à une plus grande efficacité ?
La réponse est qu'un hyperviseur peut présenter une connexion virtuelle aux ressources du serveur pour l 'OS d'une VM. Ce qui est important, cependant, est qu'il peut le faire pour un grand nombre de VMs toutes s'exécutant sur le même serveur, même si les VMs ont toutes des OSs et des applications différentes
La figure montre un hyperviseur agissant comme interface pour les ressources matérielles d'un serveur pour plusieurs OSs.

D’un point de vue efficacité, le gain est énorme. En fait, il s’agit d’une révolution. Cette approche a changé l’industrie de l ’IT, du stockage, des serveurs, et des réseaux de manière profonde.
À haut niveau, une machine virtuelle crée une version logicielle d’un serveur qui s’exécute sur le serveur matériel. Cette VM est une copie exacte du serveur physique. De multiples VMs peuvent s’exécuter sur le même serveur physique, et peuvent toutes être adaptées aux besoins d’applications. Un serveur physique peut maintenant mixer ensemble les applications Windows et Linux sur le même matériel.
En outre, parce que les machines virtuelles (VMs) sont des logiciels, non seulement il est possible de les exécuter sur n’importe quel serveur, mais il est aussi possible de suspendre ou geler une VM et la déplacer sur un autre serveur à n’importe quel moment ; ceci est censé être rapide et facile. Il est aussi possible de réaliser une copie d’une VM ou encore un ensemble de copies et les faire tourner sur plusieurs serveurs.
La figure illustre le concept d’une VM qui n’est pas liée au serveur ou au matériel sur ce serveur.

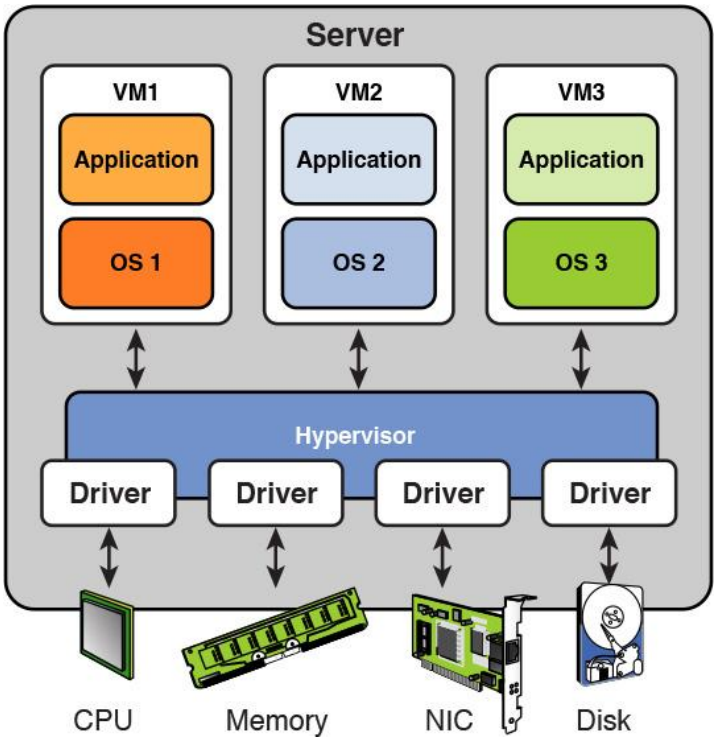


Figure 2 : Virtualisation de serveur et VM

4. Virtualisation : Définition

La virtualisation est « l’ensemble des technologies matérielles et/ou logicielles qui permettent de faire fonctionner sur une seule machine plusieurs systèmes d’exploitation et/ou plusieurs applications, séparément les uns des autres, comme s’ils fonctionnaient sur des machines physiques distinctes».
La virtualisation consiste à intercaler une couche d’abstraction entre un client et un fournisseur au sens large du terme.
La définition "formelle" de la virtualisation fait référence à l’abstraction physique des ressources informatiques.
En d’autres termes, les ressources physiques allouées à une machine virtuelle sont abstraites à partir de leurs équivalents physiques. Les disques virtuels, interfaces réseau virtuelles, réseaux locaux virtuels, commutateurs virtuels, processeurs virtuels et la mémoire

virtuelle correspondent tous à des ressources physiques sur des systèmes informatiques physiques.
L'ordinateur hôte "voit" ses machines virtuelles comme des applications auxquelles il dédie ou distribue ses ressources.
Il existe de nombreux types de virtualisation :

- d'application,
- de plate-forme,
- de réseau et
- de stockage.

Par défaut, le terme virtualisation fait référence à la virtualisation de plate-forme. Celle-ci correspond à l'utilisation de matériel serveur pour héberger plusieurs machines virtuelles invitées.
Chaque machine virtuelle est un environnement virtuel cohérent sur lequel est installé un système d'exploitation. Chaque machine virtuelle invitée fonctionne indépendamment des autres.
Un ordinateur hôte dispose de suffisamment de ressources matérielles pour offrir de la puissance de calcul et de l'espace disque à ses invités. Un système hôte typique se compose de plusieurs processeurs multicœurs, de plusieurs gigaoctets (Go) de RAM, de plusieurs téraoctets (To) d'espace disque et de stockage en réseau (NAS, Network Attached Storage) ou d'un réseau de stockage (SAN, Storage Area Network).

Pourquoi virtualiser ?

- Première raison : les matériels sont sous-utilisés et coûtent cher !
- Deuxième raison : les salles machines manquent de place et coûtent cher !
- Troisième raison : les serveurs ne participent pas à la protection de notre planète !
- Quatrième raison : les coûts d'administration des matériels sont importants !

Le premier avantage principal de la virtualisation est lié à l'optimisation de la consommation des ressources matérielles utilisées lors du fonctionnement simultané de plusieurs machines virtuelles sur un serveur physique. En effet, l'hyperviseur partage des temps processeur et de la mémoire vive et économise de l'espace de stockage. Des études estiment le taux d'utilisation moyen d'un serveur à 10 % et la virtualisation peut permettre d'atteindre les 80%.

Le deuxième avantage est la réduction du nombre de serveurs qui permet de réduire la consommation d'énergie électrique, La consommation électrique est étroitement liée au refroidissement et à la circulation d'air. La virtualisation réduit le nombre d'alimentations, de processeurs et de disques. Tous ces éléments génèrent et dissipent une quantité significative de chaleur. En diminuant le nombre d'éléments dégageant de la chaleur, on diminue également la puissance nécessaire pour refroidir efficacement la pièce.
Cela conduit au troisième avantage, puisque des études font état de réduction des coûts énergétiques qui peut atteindre 70% et de réduction d'émission de CO_2 qui peut atteindre 4 tonnes par serveur virtualisé.

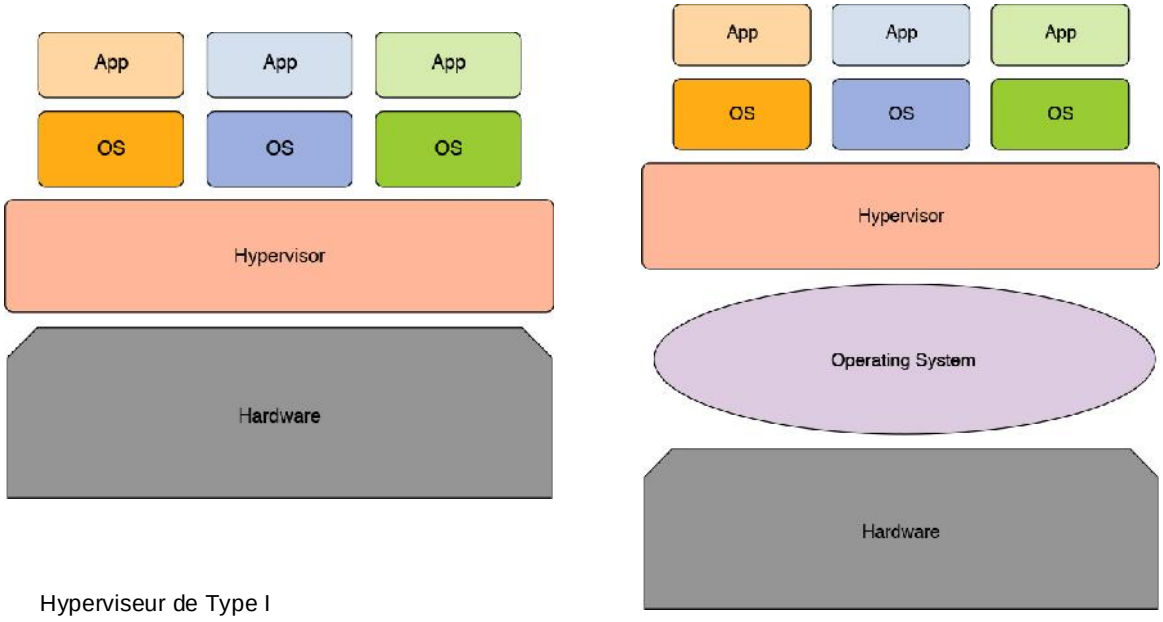
Le quatrième avantage est la concentration des services dans un seul serveur. Cela va libérer du temps pour un service informatique correctement formé en lui permettant de se concentrer sur des activités à plus forte valeur ajoutée. En effet, l'administration et la maintenance d'un serveur occupe un temps non négligeable du service informatique.
Les fonctionnalités proposées par les éditeurs de virtualisation apportent un gain en terme de maintenabilité et de souplesse d'exploitation. La virtualisation offre un lot d'outils conséquents parmi lesquels on retrouve les créations de nouvelles machines virtuelles, facilitées par la création de modèles, les migrations à chaud, le clonage, le déplacement à chaud d'un hyperviseur à un autre, la prise en compte de l'état de la machine virtuelle en direct et son éventuel redémarrage. De plus, il sera plus facile de respecter les bonnes

pratiques d'administration des serveurs qui préconisent un serveur pour un service. Enfin, la virtualisation augmente la disponibilité des serveurs en permettant une reprise d'activité plus rapide que dans une architecture classique composée de plusieurs serveurs physiques. Toutes les fonctionnalités de la virtualisation apportent la souplesse nécessaire pour assurer une reprise d'activité rapide et garantir la continuité de l'activité.

5. Hyperviseur

La présentation de la virtualisation se focalise très souvent sur le concept de machine virtuelle (VM).
Toutefois, lorsqu'on s'attache à mieux appréhender la virtualisation, l 'hyperviseur est une brique essentielle qui fait fonctionner l'ensemble.
Un hyperviseur est une plate-forme de virtualisation qui permet à plusieurs systèmes d'exploitation et donc plusieurs VMs de travailler sur une même machine physique en même temps. Il s'agit donc d'un moniteur de machine virtuelle (VM).
Il existe deux types d 'hyperviseur (Figure 3):

- Un hyperviseur de type 1 est un système qui s'installe directement sur la couche matérielle du serveur. Ces systèmes sont allégés de manière à se «concentrer » sur la gestion des OSs invités c'est-à-dire ceux utilisés par les VMs qu'ils contiennent. Ceci permet de libérer le plus de ressources possibles pour les VMs. Toutefois, il est possible d'exécuter uniquement un hyperviseur à la fois sur un serveur.
- Un hyperviseur de type 2 est un logiciel qui s'installe et s'exécute sur un OS déjà en place. De ce fait, plus de ressources sont utilisées étant donné qu'on fait tourner l'hyperviseur et l 'OS qui le supporte, il y a donc moins de ressources disponibles pour les machines virtuelles. L'intérêt qu'on peut trouver c'est le fait de pouvoir exécuter plusieurs hyperviseurs simultanément vu qu'ils ne sont pas liés à la couche matérielle.



Hyperviseur de Type I
aussi appelé hyperviseur bare-metal

Hyperviseur de Type II
aussi appelé hyperviseur hosté

Figure 3 : Hyperviseurs de type 1 et 2

Les principaux hyperviseurs de type 1 du marché sont ceux de VMware (ESX), de Microsoft (Hyper-V), de RedHat (KVM), d'Oracle (Oracle VM) et l'hyperviseur open-source Xen.

Il existe des hyperviseurs de type 2 qui sont des applications de virtualisation qui s'exécutent non pas directement sur du hardware mais sur un système d'exploitation (Windows, Linux, MacOS). Les principales solutions de type 2 sont VM VirtualBox (Oracle), WMware Workstation (sur Windows et Linux) et Fusion (sur Mac), et l'hyperviseur open-source QEMU.

6. Conteneurisation versus Virtualisation

Une machine virtuelle (VM - Virtual Machine) « imite » intégralement un serveur. Dans un serveur virtualisé type, chaque VM « invitée » contient un système d'exploitation complet, avec ses pilotes, fichiers binaires ou bibliothèques, ainsi que l'application elle-même. Chaque VM s'exécute alors sur un hyperviseur, qui lui-même fait fonctionner le matériel du serveur physique. A la base, le concept de conteneurisation permet aux instances virtuelles de partager un système d'exploitation hôte unique, avec ses fichiers binaires, bibliothèques ou pilotes.

Cette approche réduit le gaspillage des ressources car chaque conteneur ne renferme que l'application et les fichiers binaires ou bibliothèques associés. On utilise donc le même système d'exploitation (OS) hôte pour plusieurs conteneurs, au lieu d'installer un OS (et d'en acheter la licence) pour chaque VM invitée. Ce procédé est souvent appelé virtualisation au niveau du système d'exploitation.

Le rôle de l'hyperviseur est alors assuré par un moteur de conteneurisation, tel que Docker, qui s'installe par-dessus le système d'exploitation hôte.

Docker est une application qui fournit des capacités de conteneur en interagissant directement avec le système d'exploitation hôte, et offrant un moyen de créer des conteneurs qui peuvent être packagés, répliqués, portés, sauvegardés, etc. Docker est une plate-forme pour construire et exécuter des applications distribuées.

Comme le conteneur de chaque application est libéré de la charge d'un OS, il est nettement plus petit, plus facile à migrer ou à télécharger, plus rapide à sauvegarder ou à restaurer. Enfin, il exige moins de mémoire. La conteneurisation permet au serveur d'héberger potentiellement beaucoup plus de conteneurs que s'il s'agissait de machines virtuelles. La différence en termes d'occupation peut être considérable, car un serveur donné accueillera de 10 à 100 fois plus d'instances de conteneur que d'instances d'application sur VM.

En isolant les conteneurs les uns des autres, la conteneurisation assure la sécurité des applications et empêche la prolifération de logiciels malveillants entre les instances, même si, par définition, l'isolation est plus importante entre VM qu'entre conteneurs.

Les applications exécutées dans un conteneur peuvent être des applications autonomes ou système d'exploitation (OS) en tant qu'OS invité. Sur la base de l'application, les conteneurs peuvent être divisés en deux catégories:

• Conteneur d 'OS
• Conteneur d'application.

La figure 4 compare VM et conteneur.

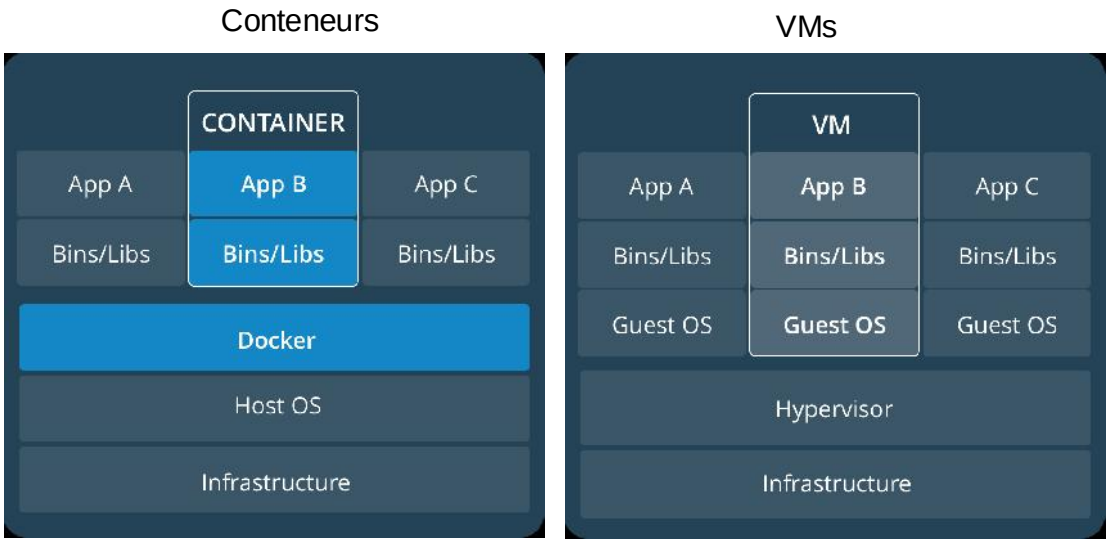


Figure 4 : Comparaison entre conteneur et machine virtuelle

7. Cloud computing et virtualisation

Le cloud Computing est une évolution des technologies de virtualisation. La virtualisation permet de donner plus d'agilité aux data centers, grâce aux trois propriétés suivantes :
• mutualisation des ressources : la virtualisation permet d'affecter les ressources d'une même machine à plusieurs applications ;
• abstraction sur la localisation : l'application est « quelque part » sur l'une des machines constitutives de la plateforme de virtualisation.
• élasticité : il est possible d'allouer des ressources supplémentaires à une application proche de la saturation, dans les limites physiques de la plateforme.

Le cloud Computing reprend ces propriétés, mais à une plus grande échelle :
• dans le cadre des plateformes de cloud Computing publiques (Google, Amazon, etc.), la mutualisation de ressources se fait à l'échelle de plusieurs milliers d'entreprises. On dispose donc de bénéfices liés au facteur d'échelle ;
• l'abstraction sur la localisation est à l'échelle de plusieurs continents dans le cadre des clouds publics : la garantie sur l'intégrité des données est donc supérieure à celle d'un data center utilisant deux sites distants de quelques kilomètres.
• avec des plateformes de plusieurs dizaines de milliers de serveurs, les clouds publics proposent une réserve de puissance et donc une élasticité exceptionnelle.

Le cloud offre deux grandes familles de services (Figure 5) :
• des services de fourniture d'application en location, appelés SaaS. Ces services sont généralement facturés au nombre d'utilisateurs actifs ;
• des services techniques de plateforme d'exécution en location, appelés PaaS et laaS. Ces services sont facturés selon les ressources techniques consommées.

SaaS signifie Software as a Service, c'est-à-dire un logiciel fourni sous la forme de service. Il s'agit donc de location d'application opérationnelle, clef en main, et non d'achat de logiciel informatique, à installer soi-même sur une machine. Les SaaS s'adressent donc aux utilisateurs finaux.

PaaS signifie Platform as a Service ou plateforme sous forme de service. Il s'agit de location de plateforme technique, permettant l'exécution de code développé en spécifique. Les PaaS s'adressent donc aux développeurs.

IaaS signifie Infrastructure as a Service ou infrastructure sous forme de service. Il s'agit de location de plateforme technique, permettant l'exécution d'architectures applicatives complètes, comprenant base de données, serveur d'application, etc. Les IaaS s'adressent donc aux équipes d'exploitation.

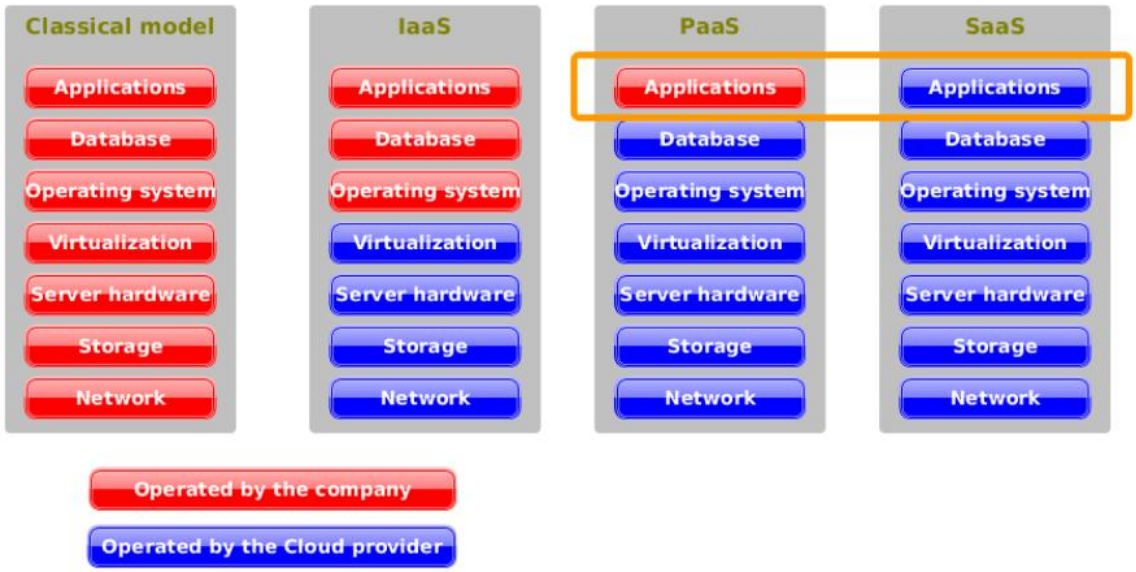


Figure 5 : Niveaux de service de cloud

8. Le réseau du datacenter

Pendant longtemps, les réseaux Ethernet de datacenter ont été bâtis en strates selon des architectures rappelant un arbre hiérarchique (Figure 6). Ce système hiérarchique est en train de trouver ses limites avec l'explosion des trafics de données entre serveurs (trafic dit Est- Ouest, par opposition au trafic Nord-Sud, qui va des serveurs vers les utilisateurs). Dans un réseau à architecture hiérarchique, on trouve au niveau le plus bas la couche d'accès qui permet aux utilisateurs d'accéder au réseau du datacenter.
La couche intermédiaire est une couche d'agrégation ou de distribution à laquelle la couche d'accès est connectée de façon redondante. Cette couche d'agrégation fournit une connectivité aux commutateurs d'accès adjacents et aux éléments du datacenter. Elle est elle-même connectée au sommet de l'arbre aussi baptisé cœur de réseau.
La couche de cœur de réseau fournit les services de routage entre les différentes sections du datacenter et elle est aussi en charge d'assurer la connectivité avec les éléments externes au datacenter comme la connexion avec d'autres datacenters, la connexion avec Internet ou la liaison vers des services externes au SI d'entreprise (services cloud, applications SaaS).
Ce modèle hiérarchique est sujet à certains goulets d'étranglements, si les liens entre les différentes couches sont sursouscrits. Dans ce modèle, l'usage de spanning tree est courant, ce qui se traduit aussi par la non-exploitation de tous les liens (certains n'étant pas actifs). Les multiples niveaux se traduisent aussi par un impact en termes de latence, certains serveurs devant remonter jusqu'au niveau de cœur pour communiquer avec d'autres serveurs.

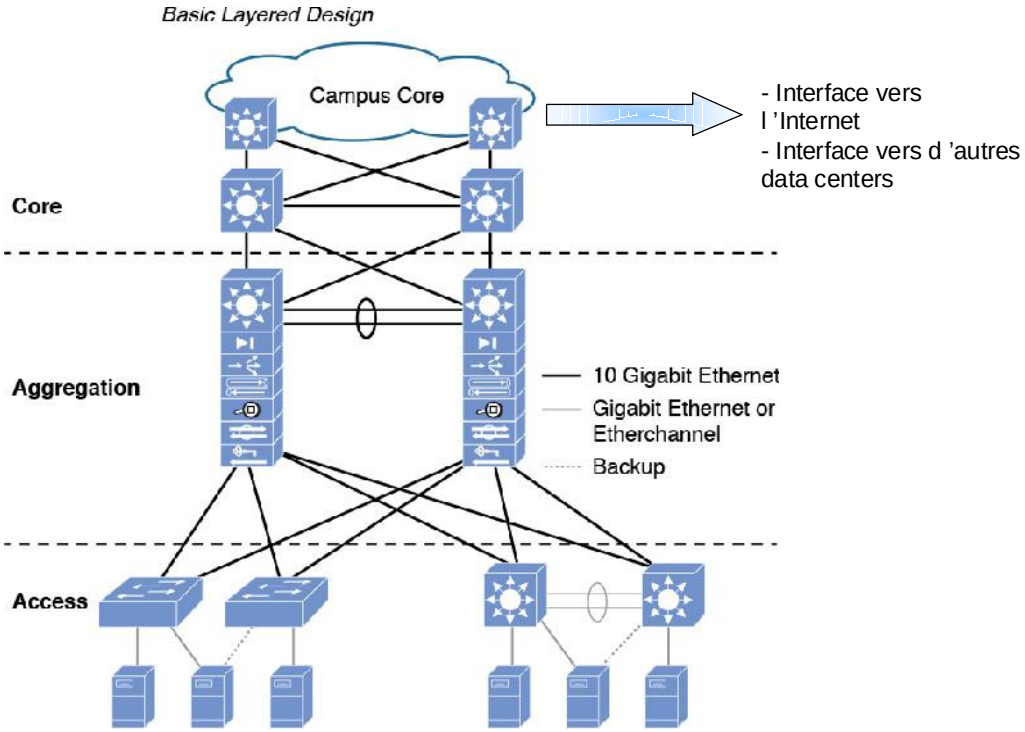


Figure 6 : Architecture réseau du datacenter traditionnel

9. Evolution de l'architecture réseau de data center

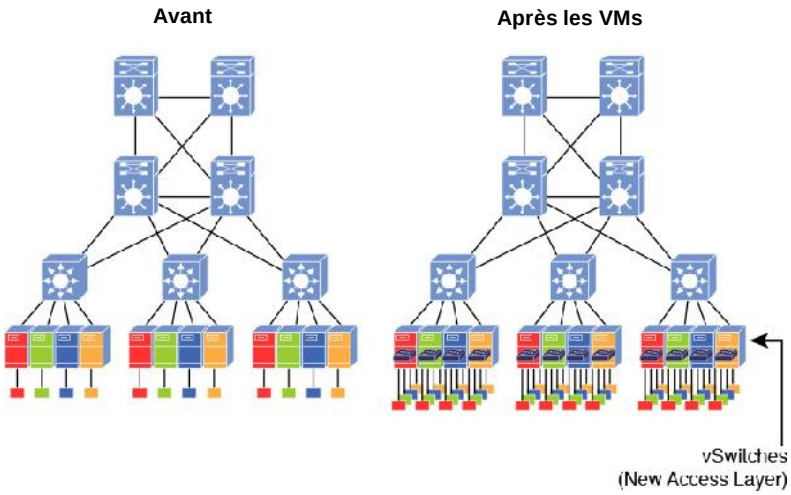


Figure 7 : Data center traditionnel (gauche) versus data center virtualisé (droite)

A gauche à la figure 7 est considéré le modèle classique où les commutateurs d'accès sont connectés aux serveurs.
A droite à la figue 7, il s'agit du data center virtualisé ou un commutateur virtuel (vSwith) par serveur permet aux VMs de ce serveur de pouvoir communiquer entre elles et avec le monde externe.
L'autre impact de la virtualisation est que les modèles de trafic ont migré d'un modèle principalement « client à serveur » (à l'intérieur du data center vers l'extérieur et vice-versa) pour être principalement « serveur à serveur » (au sein du data center) dès lors que les workflows de VM à VM augmentent et dès que la prévalence des pools de ressources

indépendantes augmente. En d'autres termes, au lieu de disposer d'un serveur qui fait tout, il y a des groupes de ressources spécialisées dans un cloud (i.e., le data center virtualisé).
Ce changement de trafic du « client à serveur » à « serveur à serveur » est souvent désigné comme un changement de trafic nord-sud à un trafic est-ouest. Le réseau de data center virtualisé est schématisé par des serveurs à la base s'exécutant de gauche à droite (ou est-ouest), puis par l'empilage de l'accès, l'agrégation, et des couches « core » au-dessus de telle sorte que la frontière du data center soit au-dessus.
Un exemple est une requête d'un client Web concernant le temps. Elle peut être traitée par un serveur Web (trafic nord-sud) qui obtient les données d'un serveur de stockage dans le même data center (trafic est-ouest) avant de répondre au client Web (trafic nord-sud).
Les data centers doivent encore gérer efficacement le trafic nord-sud, mais l'augmentation du trafic est-ouest a augmenté le besoin de bande passante uniforme et la latence entre les serveurs, en particulier étant donné que l'emplacement physique des VMs peut changer. La conception du data center doit tenir compte de cela pour assurer que la performance ne soit pas affectée.
Dans l'ancien modèle, le trafic entre les deux serveurs remonte un niveau (ou deux) dans la hiérarchie, puis redescend à un autre commutateur du même niveau. Appelée « hair pinning », cette méthode de commutation serveur-serveur n'est pas mauvaise tant que les serveurs sont fixes (afin que les performances soient prédictibles) et les communications entre serveurs se sont pas trop fréquentes. Cela n'est plus vrai dans le cas des data center virtualisés.
Dans les data center virtualisés, le premier commutateur d'accès est l 'hyperviseur, qui est lui-même un commutateur virtualisé (il supporte VLAN et VxLAN). Le premier commutateur physique est présent au-dessus du rack de serveurs physiques, jouant le rôle de commutateur d'accès pour de nombreuses VMs qui fonctionnent sur le rack. Ce commutateur est appelé top of rack (TOR) comme montré à la figure 8; il se connecte à son tour à un ou plusieurs commutateurs d'agrégation qui agrègent le trafic jusqu'au commutateur core. Le but est de garder le plus gros du câblage à l'intérieur du rack. Le seul câblage vers l'extérieur est celui du ToR vers les commutateurs d'agrégation. Ce nouveau modèle est appelé Leaf-Spine (littéralement, feuille-épine dorsale) et fonctionne bien parce qu'il est scalable horizontalement (est-ouest) sans dégrader les performances du trafic nord-sud.

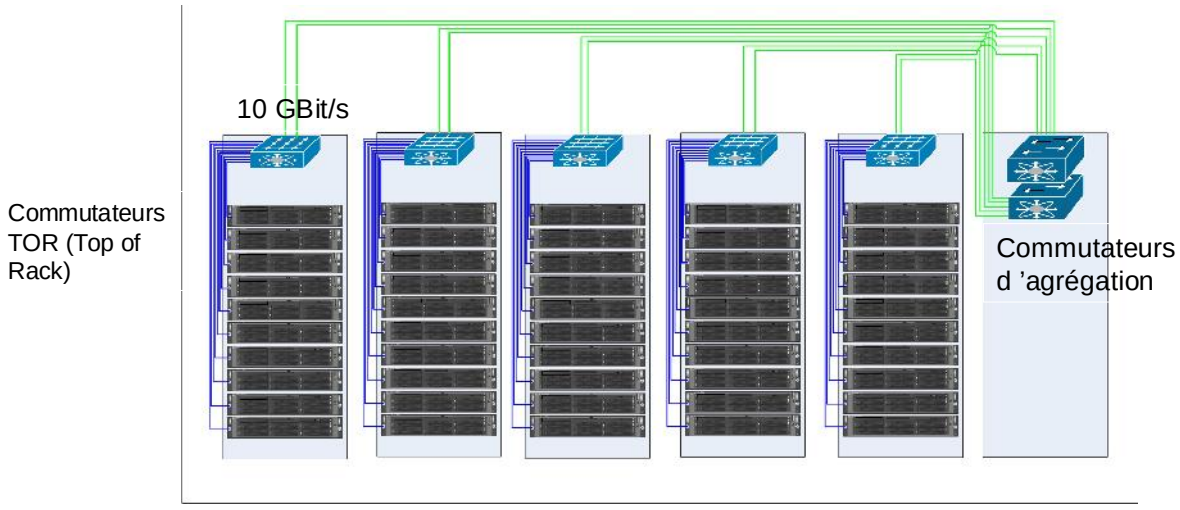


Figure 8 : Commutateur Top of Rack

Références

Jim Doherty, « SDN and NFV Simplified », Pearson Education, Inc, 2016.