Virtualisation & Partage de Charge

PRÉSENTÉ PAR
Arnaud A.A. AMELINA

VIRTUALISATION

Open sources Solutions

AFAHOUNKO Danny
RHCE – RedHat Certified Engineer

Sommaire

TERMINOLOGIES

Système Hôte

Le système hôte est la machine physique qui héberge les machines virtuelles.

Hyperviseur

L'hyperviseur fournit permet l’abstraction de la couche matérielle de la machine hôte. Il permet aussi l'exécution des différents systèmes invités de sur le système hôte. Il gère leur fonctionnement et fournit l'isolation entre les systèmes invités.

Système Invité

Le système invité est l'instance du système d'exploitation qui s'exécute dans l'environnement virtuel du système hôte. Il est encore appelé machine virtuelle (VM : Virtual Machine).

Historique

HISTORIQUE

La virtualisation est au centre de plusieurs débats informatiques de nos jours.

Mais beaucoup de personnes ne réalisent pas que la virtualisation n'est pas un nouveau concept.

HISTORIQUE

La virtualisation a été développée dans les années 1960 sur les mainframes IBM.

IBM System/370 a été la première plateforme commercialisée conçue pour la virtualisation. Avec l'introduction du système d'exploitation CP/CMS, plusieurs instances de systèmes d'exploitation peuvent être exécutés simultanément sur les mainframes IBM system/370.

La combinaison matérielle et logicielle pour supporter la virtualisation fut au centre des travaux de recherches est devenue une base dans la lignée des mainframes IBM. Par conséquent tous les mainframes IBM récents dans la lignée des systèmes Z continuent de fournir un support matériel pour la virtualisation.

Le logiciel originel CP/CMS a été remplacé par le z/VM (Virtual Machine) qui exploite plus efficacement le matériel de virtualisation.

De nombreuses approches de virtualisation moderne doivent beaucoup à l'implémentation originale des mainframes d'IBM.

HISTORIQUE

Dans la deuxième moitié des années 80 et dans les débuts des années 90, on a créé des embryons de virtualisation pour les ordinateurs personnels. En effet l'ordinateur Amiga pouvait lancer des pc x386, Macintosh 68xxx, voire des solutions X11, et bien sûr le tout en multitâche. Ces solutions pouvaient être soit purement logicielles, soit couplées à du matériel additionnel (ajout de processeur, carte réseau...). Cette machine, très en avance sur son temps, a popularisé cette technologie. Pour les PC il y avait le "SideCar" et "PC Task", pour le Macintosh Emplant et ShapeShifter.

Les grands Unix ont suivi avec les architectures NUMA des Superdome d'HP (PA-RISC et IA-64) et des E10000/E15000 de Sun (UltraSparc).

HISTORIQUE

Dans la seconde moitié des années 1990, les émulateurs sur x86 des vieilles machines des années 1980 ont connu un énorme succès, notamment les ordinateurs Atari, Amiga, Amstrad et les consoles NES, SNES, Neo-Geo AES.

La société VMware développa et popularisa à la fin des années 1990 et au début des années 2000 un système propriétaire de virtualisation logicielle des architectures de type x86 pour les architectures de type x86. Les logiciels libres Xen, QEMU, Bochs, Linux-VServer , Virtual Box et les logiciels propriétaires mais gratuits VirtualPC, VirtualServer et VMware Server ont achevé la popularisation de la virtualisation dans le monde x86. VMware a dernièrement rendu gratuit une version allegée de son hyperviseur phare ESX3i.

Les fabricants de processeurs x86 AMD et Intel ont mis en œuvre la virtualisation matérielle dans leurs gammes dans la seconde moitié de l'an 2000.

Pourquoi Virtualiser ?

POURQUOI VIRTUALISER ?

Qu'attendent les entreprises de la virtualisation ?

- La réduction du nombre de serveurs
- La réduction de l'espace occupé dans les datacenters
- La réduction de la consommation énergétique des datacenters
- Réduction des coûts d'administration
- Amélioration de la flexibilité et de la rapidité des services
- Amélioration de la qualité de services

Définitions & Concepts

DÉFINITIONS

La virtualisation est – dans sa définition la plus simple – une abstration du matériel à partir du logiciel.

L'encyclopédie francophone en ligne Wikipédia définit la virtualisation comme « l'ensemble des techniques 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 » [WFv].

Il s'agit donc d'utiliser une seule machine physique en remplacement de plusieurs et d'utiliser les possibilités offertes par la virtualisation pour démultiplier le nombre de machines virtuelles.

CONCEPTS

Il existe plusieurs catégories de virtualisation, utilisant chacune des technologies différentes. Les technologies les plus répandues sont :

- L'émulation (Emulation)
- La virtualisation complète (Full Virtualization)
- La paravirtualisation (Paravirtualization)
- Isolateur ou Cloisonnement (Operating System level virtualization)
- La Virtualisation matérielle

Pour être complet, nous allons aussi brièvement énumérer deux autres types de virtualisation même s'ils ne sont pas capables d'exécuter un sytème d'exploitation complet:

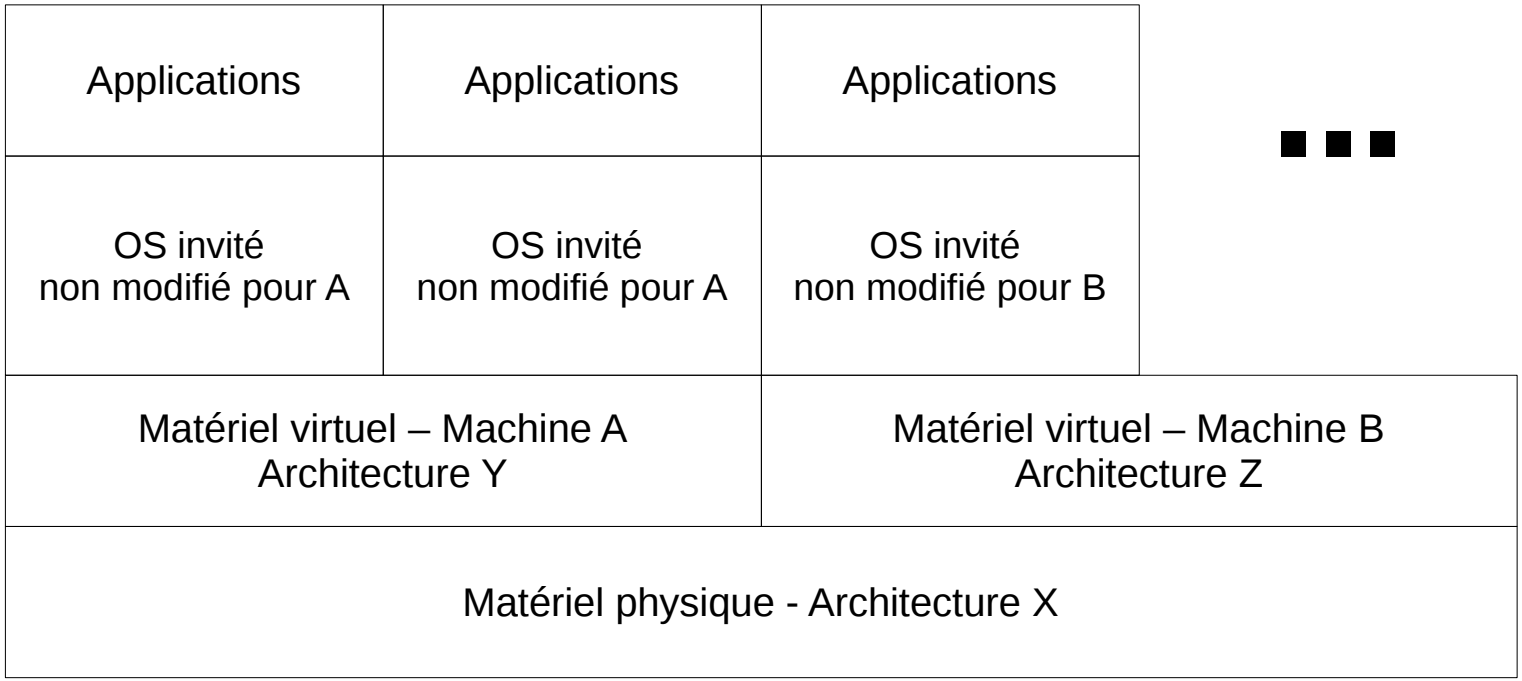
- Virtualisation des bibliothèques (Library virtualization)
- Virtualisation au niveau applicatif (Application Level virtualization)

EMULATION

Dans l'émulation, la machine virtuelle simule le matériel et l'ensemble des instructions nécessaires pour exécuter sans modification des clients d'une architecture matérielle complètement différente.

En règle générale, l'émulation est utilisée pour créer des systèmes d'exploitation nouveaux ou microcodes pour de nouveaux matériels, avant que le matériel ne soit disponible physiquement.

EMULATION



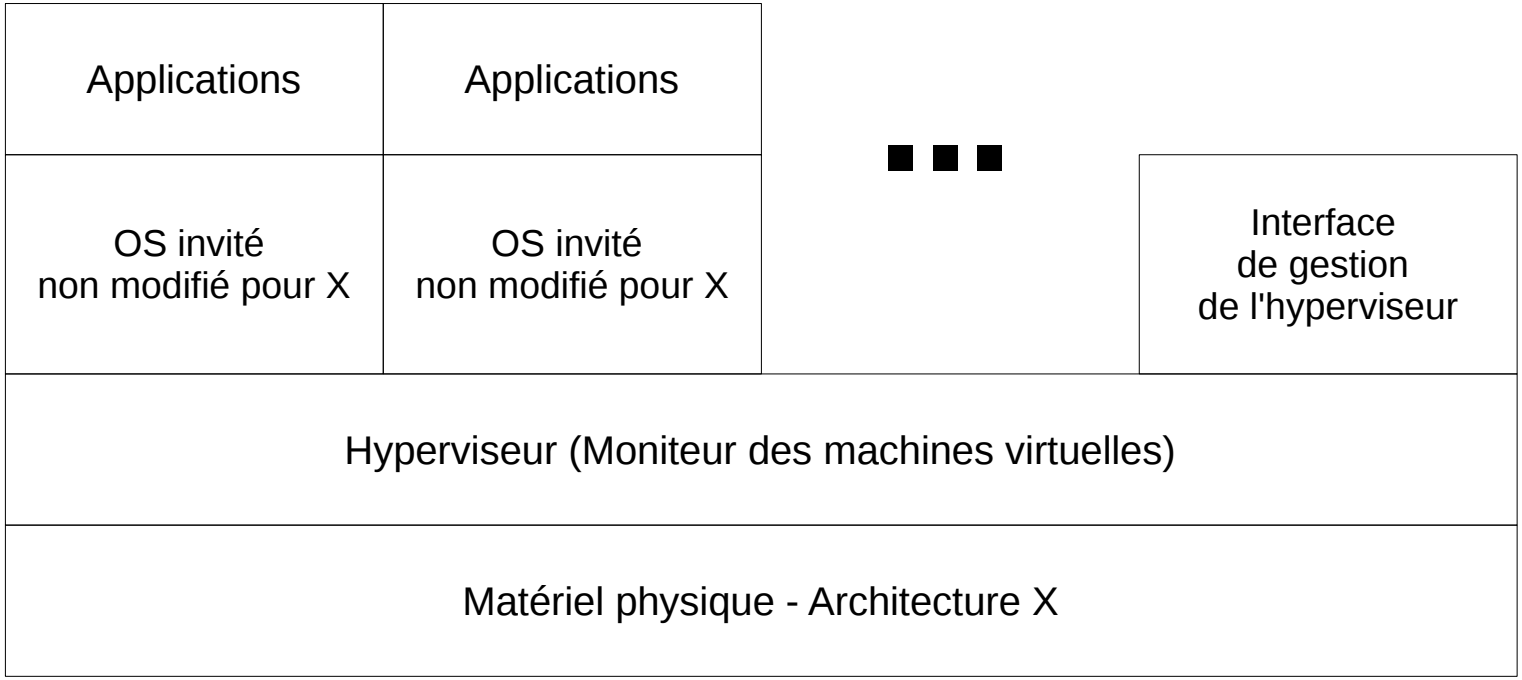
L'émulateur de machines virtuel fournit une architecture virtuelle qui peut ne pas être différente de l'architecture du système hôte.
Les OS invités s'exécute sans modification sur le matériel virtuel.

VIRTUALISATION COMPLÈTE

La virtualisation complète (full virtualization) est similaire à l'émulation. Comme dans l'émulation, les systèmes d'exploitation et applications non modifiés s'exécutent dans la machine virtuelle. La virtualisation complète diffère de l'emulation dans le fait que les systèmes d'exploitation et applications sont conçus pour fonctionner sur la même architecture que celle de la machine physique hôte.

Ce qui permet à un système de virtualisation complète d'exécuter plusieurs instructions directement sur le matériel physique sous-jacent. L'hyperviseur dans ce cas, donne l'accès au matériel physique et donne à chaque système d'exploitation invité l'illusion d'avoir sa propre copie. Il ne doit plus utiliser un logiciel pour simuler une architecture de base différente.

VIRTUALISATION COMPLÈTE

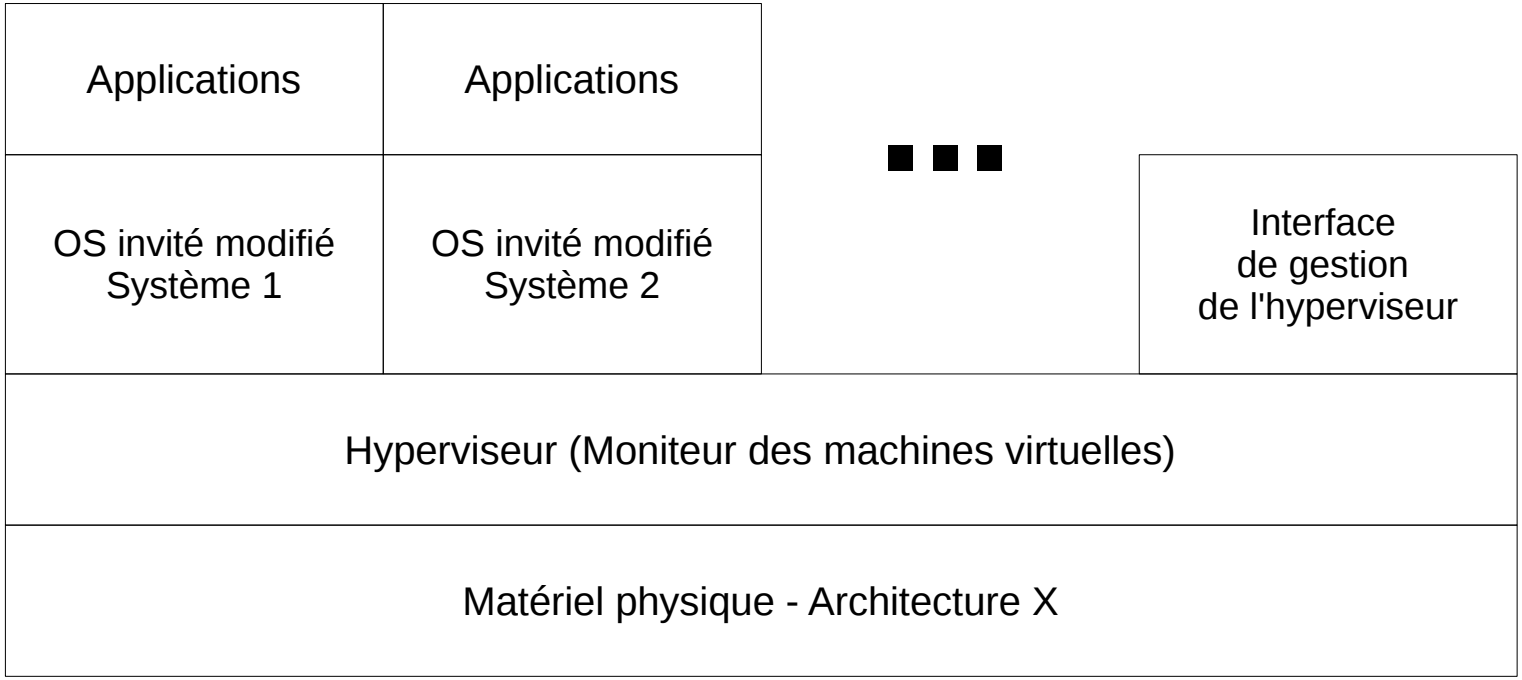


L'hyperviseur présente le matériel physique du système hôte à tous les systèmes invités.
Les systèmes d'exploitation invités ont un accès au matériel sous-jacent et s'exécutent sans modification et n'ont pas conscience qu'ils sont virtualisés.

PARAVIRTUALISATION

Dans la paravirutalisation, l'hyperviseur exporte une version modifiée de la machine physique hôte. La machine virtuelle exportée a la même architecture que le système hôte; ce qui n'est pas forcément le cas dans l'émulation.
Le système d'exploitation invité requière quelques légères modifications pour fonctionner sur l'architecture émulée par l'hyperviseur.
Les modifications effectuées visent à rendre le système émulé « au courant » du fait qu'il s'exécute dans une machine virtuelle. De ce fait, il pourra collaborer plus étroitement avec le système hôte, en utilisant une interface spécifique et accéder au matériel virtuel via des couches d'abstraction.

PARAVIRTUALISATION

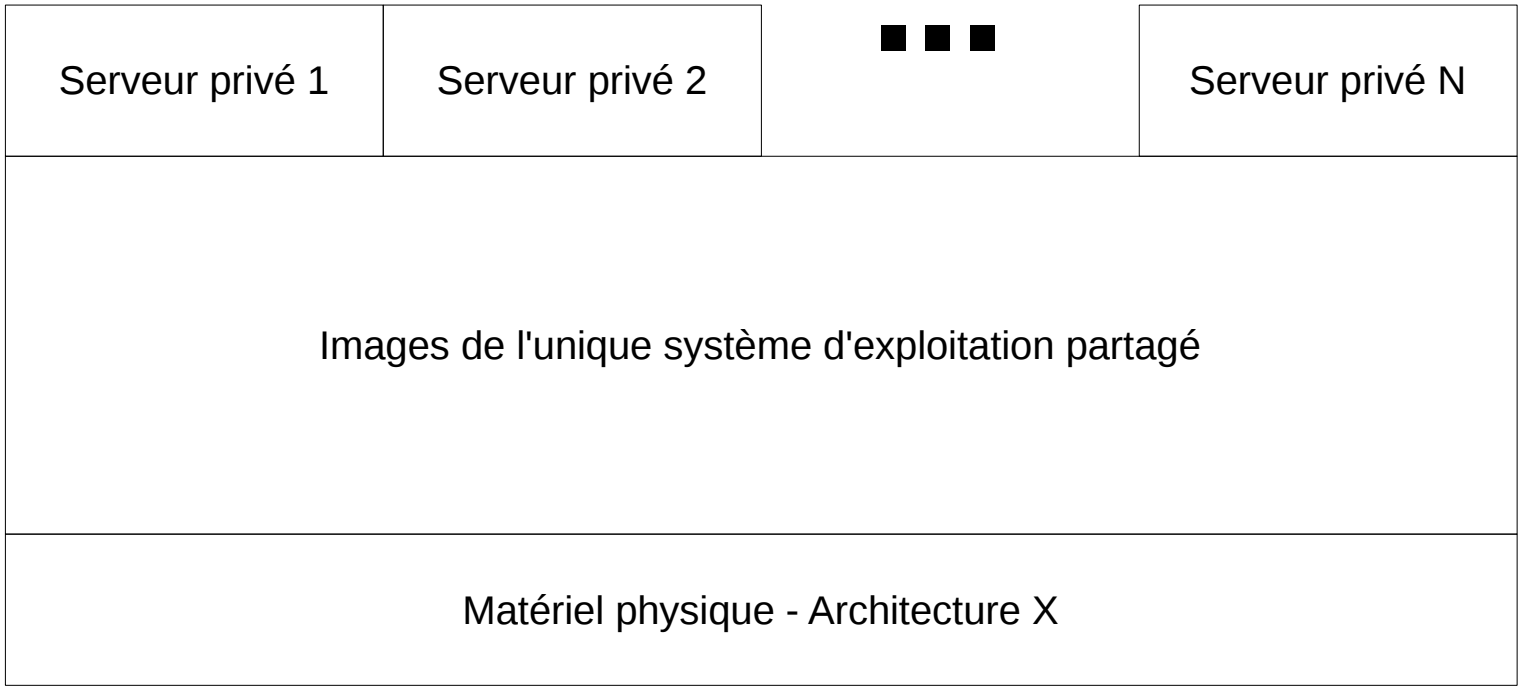


La paravirtualisation est similaire à la virtualisation complète à la seule différence que les systèmes invités sont modifiés et ont conscience qu'il s'exécutent dans un environnement virtuel.

CLOISONNEMENT

Une autre pratique répandue dans le domaine de la virtualisation est le cloisonnement.
Derrière ce nom se cachent plusieurs technologies visant à séparer fortement les processus s'exécutant sur un même système d'exploitation. Le cloisonnement vise à isoler chaque processus dans un conteneur dont il est théoriquement impossible de sortir. Un processus isolé de la sorte ne saura pas quels autres processus s'exécutent sur le même système, et n'aura qu'une vision limitée de son environnement. Le but principal de cette technologie est d'améliorer la sécurité du système d'exploitation et des applications.

CLOISONNEMENT



Avec le cloisonnement tous les serveurs virtuels privés sont exécutés dans le contexte d'un seul système d'exploitation partagé installé sur le matériel physique.

LA VIRTUALISATION MATERIELLE

Le support de la virtualisation peut être intégré au processeur ou assisté par celui-ci, le matériel se chargeant, par exemple, de virtualiser les accès mémoire ou de protéger le processeur physique des accès les plus bas niveau. Cela permet de simplifier la virtualisation logicielle et de réduire la dégradation de performances.

VIRTUALISATION DES BIBLIOTHEQUES

La virtualisation bibliothèque émule les systèmes d'exploitation ou de sous-systèmes via une bibliothèque de logiciels spéciaux.

Un exemple de ce type de virtualisation est la bibliothèque de Wine disponible sur les systèmes Linux. Wine fournit un sous-ensemble de l'API Win32 comme une bibliothèque pour permettre aux applications de bureau Windows d' être exécutées dans un environnement Linux.

VIRTUALISATION AU NIVEAU APPLICATIF

La virtualisation au niveau applicatif est l'approche de l'exécution d'applications dans un environnement d'exécution virtuel. Ce qui est différent de l'exécution normale d'une application sur un matériel.
L'environnement d'exécution virtuel fournit une API standard pour l'exécution multi-plateforme et gère l'utilisation des ressources locales par l'application. Il peut également fournir des ressources telles que le modèle de thread, les variables d'environnement, les bibliothèques d'interface utilisateur et les objets qui aide à la programmation d'applications.

L'exemple le plus répandu d'un tel environnement d'exécution virtuel est le Sun Java Virtual Machine.

RECAPITULATIF DES TYPES DE VIRTUALISATION

Virtualisation	Description	Avantages	Inconvénients
Emulation	L'hyperviseur émule une machine virtuelle complète	Simule un matériel qui n'est pas physiquement disponible	Faible performance Faible densité
Complète	L'hyperviseur émule une machine virtuelle identique à l'hôte.	Flexibilité – exécute les OS de plusieurs éditeurs	Les OS invités ne savent pas qu'ils sont virtualisés. Problèmes E/S.
Para	L'hyperviseur émule une machine virtuelle identique à l'hôte mais spécialisée	Léger, rapide OS invité coopère avec l'hyperviseur pour les E/S	Les OS invités doivent être légèrement modifié pour prendre en compte l'environnement virtuel.
Cloisonnement	Seul le système d'exploitation hôte est modifié pour permettre l'isolation	Rapide, légère couche de virtualisation. Infrastructure homogène – OS hôte et OS invité sont identiques	En pratique, une isolation complète est difficile à implémenter.
Matérielle	Support de virtualisation intégré au matériel	Augmente les performances le hyperviseur	Solutions propriétaires et ne suivent aucun standard

RÉCAPITULATIF DES TYPES DE VIRTUALISATION

Virtualisation	Description	Avantages	Inconvénients
Bibliothèque	Emule un système d'exploitation ou un sous-système via un logiciel spécial	Fourni les API manquants pour les développeurs d'applications.	Souvent moins performante que l'application native.
Application	L'application tourne dans un environnement d'exécution virtuel ce qui lui fournit une API standard quelque soit la plateforme.	Augmente la portabilité des applications et gère automatiquement les ressources	L'exécution est moins performante que le code natif.

Virtualisation - Avantages

AVANTAGES DE LA VIRTUALISATION

La virtualisation présente beaucoup d'avantages dans l'environnment de plus en plus dynamique des systèmes d'information:

- Plusieurs systèmes différents sur un même matériel physique
- Dimensionnement des machines virtuelles selon la demande
- Consolidation des serveurs et des services
- Un service = un serveur
- Economies substantielles sur le matériel et consommation énergétique
- Suppression des contraintes liées aux matériels ou aux versions de logiciels
- Haute disponibilité grâce à la Live Migration et aux clichés instantannés
- Fourni un environnement de tests aux développeurs et professionnels
- Tirer un meilleur avantage des processeurs récents multi-core
- Réduction du TCO (Total Cost Ownership)
- Réduction des espaces occupés par les serveurs
- Protection de l'environnement :)

Virtualisation - Inconvénients

INCONVENIENTS DE LA VIRTUALISATION

Comme toutes solutions informatiques, la virtualisation présente des contraintes :

- Le matériel doit être dimensionné au besoin (Processeurs, Disques, Mémoires ...)
- Certaines solutions de virtualisation requièrent des matériels compatibles
- Toute la sécurité de l'infrastructure virtuelle dépend de l'hyperviseur
- Coût de la formation sur la solution de virtualisation à implémenter
- L'administration d'un serveur physique est différente d'un serveur virtuel
- Faibles performances
- Toutes les machines virtuelles invitées dépendent de la machine physique hôte

Virtualisation - Sécurité

LA SECURITE EN VIRTUALISATION

- L'hyperviseur se charge de l'isolement des machines virtuelles
- Une meilleure sécurité peut être garantie lorsque les machines hôtes sont mises en cluster; évitant les points d'echec unique (single point of failure)
- La redondance est aisée diminuant ainsi les risques
- La spécialisation de chaque machine virtuelle dans un service précis permet d'avoir des politiques de sécurité propres à chaque serveur virtuel

Solutions Open Sources

LICENCES OPEN SOURCE

Tous les types de licence Open Sources

GPL (GNU General Public License).
LGPL (GNU Lesser General Public License).
CDDL (Common Development and Distribution License).

Les licences BSD et MIT sont des licences permissives écrites respectivement par les Universités de Californie, de Berkeley et l'Insititut Technologique de Massachusetts. Comme les licences open source, ces licences permettent l'utilisation des codes sources et fournissent moins de restrictions d'utilisation que les GPL.

Pour plus d'information http://www.opensource.org/licenses.

QUELQUES SOLUTIONS OPEN SOURCE

Produits	Virtualisation	Installation	Licence
Bochs	Emulation	Hébergé	LGPL
QEMU	Emulation	Hébergé	LGPL/GPL
User Mode Linux (UML)	Para	Hébergé	GPL
LGuest	Para	Bare Metal	GPL
OpenVZ	Cloisonnement	Bare Metal	GPL
Linux VServer	Cloisonnement	Bare Metal	GPL
Xen	Complète / Para	Bare Metal	GPL
KVM	Complète	Bare Metal	GPL
Solaris Containers	Cloisonnement	Hébergé	CDDL
BSD Jails	Cloisonnement	Hébergé	BSD
Wine	Bibliothèque	Couche applicative	GPL
Java Virtual Machine	Niveau applicatif	Couche applicative	GPL

BOCHS

BOCHS est un simulateur d'ordinateur x86 disponible pour s'exécuter sur plusieurs plateforme (x86, PowerPC, SPARC, Alpha, MIPS ...).
BOCHS peut être configuré pour émuler plusieurs générations de l'architecture x86 incluant le 386, 486, Pentium, Pentium Pro et même des implémentations modernes des architechures 64 bits.
Bochs émule aussi des instructions optionnelles comme le MMX, le SSE, le SSE2, et le 3DNow.
La particularité de BOCHS est qu'il n'emule pas seulement les processeurs. Il émule aussi un système complet avec les périphériques nécessaires aux opérations standards (clavier, souris, carte graphique, cartes réseaux ...).
BOCHS est capable de faire fonctionner plusieurs systèmes d'exploitation comme des invités: gamme Windows (XP, Vista ...), DOS, Linux ...

Il est important de noter que BOCHS a besoin d'un système d'exploitation hôte pour fonctionner et ne peut être installé sur une plateforme sans système d'exploitation (Bare Hardware).
BOCHS s'installe généralement sur les systèmes Linux, Windows ou Mac OS X.

Plus de détails : http://bochs.sourceforge.net/

QEMU

QEMU supporte deux modes de fonctionnement.

• **Le mode d'émulation complète du système (Full System Emulation):**
Ce mode est similaire au BOCHS parce qu'il émule un ordinateur complet avec les périphériques. Ce mode émule plusieurs architectures de processeurs comme le x86, le x86_64, ARM, SPARC, PowerPC et MIPS avec des vitesses raisonnables utilisant la translation dynamique.
Ce mode permet d'émuler un environnement capable de faire fonctionner des systèmes d'exploitation invités Microsoft Windows ou Linux sur des plateformes Linux, Solaris ou FreeBSD.

• **Le mode d'émulation utilisateur (User Mode Emulation):**
Ce mode est seulement disponible lorsqu'on exécute QEMU sur une plateforme Linux. Ce mode permet d'exécuter les binaires d'une autre architecture. Par exemple, les binaires compilés pour MIPS peuvent être éxécutés sur une architecture x86 de Linux.

Plus de détails : http://www.nongnu.org/qemu/

USER MODE LINUX (UML)

User Mode Linux ou UML est un noyau Linux compilé qui peut être exécuté dans l'espace utilisateur comme un simple programme. Il permet donc d'avoir plusieurs systèmes d'exploitation virtuels sur une seule machine physique hôte exécutant Linux.
UML est spécialement conçu pour exécuter des machines virtuelles Linux invitées sur le système hôte Linux.

Plus de détails : http://user-mode-linux.sourceforge.net/

LGUEST

LGUEST est une autre méthode de virtualisation qui se base sur le noyau Linux. LGUEST est maintenu par Rusty Russell et a été fusionné au noyau au cours de la période de développement de la version 2.6.23. Un détail très intéressant de LGUEST est qu'il est implémenté comme un module du noyau. Bien LGUEST pourrait ne pas être aussi fonctionnel que d'autres types de virtualisation, il est un outil extrêmement bon pour l'apprentissage et l'expérimentation avec des implémentations de virtualisation en raison de sa taille relativement petite de code. Une version expérimentale 64 bits de LGUEST est en cours d'élaboration par Red Hat. Bien LGUEST est nouveau, son inclusion rapide dans les sources du noyau en amont, il est intéressant de noter.

Plus de détails : http://lguest.ozlabs.org/

OPENVZ

OpenVZ est une technique de virtualisation de niveau système d'exploitation basée sur le noyau Linux. OpenVZ permet à un serveur physique d'exécuter de multiples instances de systèmes d'exploitation isolés, connus sous le nom de serveurs privés virtuels (VPS) ou environnements virtuels (VE).

Le système d'exploitation invité et hôte doivent être de type Linux (bien que les distributions de Linux peuvent être différentes dans des VEs différents). Cependant, la virtualisation au niveau OS d'OpenVZ offre une meilleure performance, une meilleure scalabilité (i.e. évolution), une meilleure densité, une meilleure gestion de ressource dynamique, et une meilleure facilité d'administration que ses alternatives.

OpenVZ est la base de Virtuozzo, un produit propriétaire fourni par SWsoft, Inc. OpenVZ est distribué sous la Licence publique générale GNU version 2.

OpenVZ comprend le noyau Linux et un jeu de commandes utilisateurs.

Plus de détails : http://openvz.org/

LINUX VSERVER

Linux-VServer est un isolateur des contextes de sécurité combiné à du routage segmenté, chroot, quotas étendus et autres outils standards.

Projet lancé à l'origine par Jacques Gélinas à l'origine du patch CTX, Linux-VServer consiste en un patch pour le noyau Linux qui permet d'exécuter plusieurs applications dans différents contextes de sécurité sur une même machine hôte. Linux-VServer est également muni d'un ensemble d'outils pour installer et gérer ces contextes.

Ce projet permet d'exécuter un ou plusieurs environnements d'exploitation (systèmes d'exploitation sans le noyau) ; autrement dit, il permet d'exécuter une ou plusieurs distributions sur une distribution.
Linux-VServer est une solution de virtualisation beaucoup plus poussée que le simple chroot.

À ne pas confondre avec le Linux Virtual Server Project.

Plus de détails : http://linux-vserver.org/

XEN

Xen est un logiciel libre de virtualisation, plus précisément un hyperviseur de machine virtuelle.
Il est développé par l'université de Cambridge au Royaume-Uni. Xen permet de faire fonctionner plusieurs systèmes d'exploitation virtuels (invités) sur une seule machine hôte. Xen permet d'exécuter plusieurs systèmes d'exploitation (et leurs applications) de manière isolée sur une même machine physique sur plate-forme x86, x86-64, IA-64 et PowerPC (bientôt sur SPARC). Les systèmes d'exploitation invités partagent ainsi les ressources de la machine hôte.
Xen est un « paravirtualiseur » ou un « hyperviseur » de machines virtuelles. Les systèmes d'exploitation invités ont « conscience » du Xen sous-jacent, ils ont besoin d'être « portés » (adaptés) pour fonctionner sur Xen. Linux, NetBSD, FreeBSD (portage en cours), Plan 9 et GNU Hurd peuvent d'ores-et-déjà fonctionner sur Xen.
Xen 3 peut également exécuter des systèmes non modifiés comme Windows sur des processeurs supportant les technologies VT d'Intel ou AMD-V (nom de projet: Pacifica) de AMD.
Les architectures x86, x64, IA-64, PowerPC et SPARC sont supportées. Le multiprocesseur (SMP) et partiellement l’Hyper-Threading sont supportés.

Plus de détails : http://www.cl.cam.ac.uk/Research/SRG/netos/xen/

KVM

KVM (Kernel-based Virtual Machine) est une machine virtuelle libre pour Linux. Elle fonctionne sur les architectures x86 disposant des technologies Intel VT ou AMD SVM (AMD-V).

Le module est intégré dans le noyau Linux depuis la version 2.6.20.
KVM est un fork de QEMU. Les développeurs des deux projets essayent de ne pas trop diverger et le code source des deux projets est fréquemment resynchronisé. La principale modification apportée est le support du module kvm. Lorsqu'on parle de KVM, on parle généralement de l'ensemble : la version modifiée de QEMU et le module kvm.
Les technologies mises en place par les deux principaux fondeurs que sont AMD et Intel étant différentes, le module kvm se décline en deux sous-modules : kvm-intel et kvm-amd ; le module kvm n'étant là en fait que pour fournir à l'émulateur une abstraction supplémentaire.

Plus de détails : http://kvm.qumranet.com/kvmwiki

SOLARIS CONTAINERS

Solaris Containers (y compris les zones Solaris) est une implémentation de la technologie de virtualisation du système d'exploitation de premier niveau mis à disposition en 2005 dans le cadre de Solaris 10.

Un conteneur Solaris est la combinaison de contrôles des ressources système et la séparation de la limite prévue par zones. Loi sur les zones complètement isolées comme les serveurs virtuels dans une seule instance du système d'exploitation. En consolidant plusieurs ensembles de services d'application sur un système unique et en plaçant chacun dans des conteneurs isolés serveur virtuel, les administrateurs système peuvent réduire les coûts et fournir toutes les mêmes protections que celles des machines distinctes sur une seule machine.

Plus de détails : http://opensolaris.org/os/community/zones/

BSD JAILS

Les BSD Jails (« jail » signifie prison en anglais) sont un système logiciel permettant d'emprisonner un processus et ses descendants.
En pratique, les jails sont souvent utilisés pour répondre à deux besoins :

• contraindre l'exécution d'une application sensible (une application tournant avec des privilèges importants, comme un serveur FTP par exemple) ; cela permet de garder le système hôte sain en cas de problèmes avec l'application « emprisonnée » suite à une activité malveillante ou tout simplement lors de tests/débogage ;
• faire une « image virtuelle du système » permettant l'exécution de multiples applications ; les objectifs sont les mêmes que précédemment, mais à plus large échelle.

Souvent, une installation assez complète du système est nécessaire au bon fonctionnement des applications placées dans le jail : bibliothèques, fichiers de configuration, etc.
Les jails BSD rappellent le chroot que l'on peut trouver sous GNU/Linux/Unix, tout en offrant plus de sécurité et de plus grandes possibilités de configuration.

http://www.freebsd.org/doc/fr_FR.ISO8859-1/books/handbook/jails.html

WINE

Wine est l'acronyme récursif anglophone de « Wine Is Not an Emulator », littéralement Wine n'est pas un émulateur. Parfois, on le considère aussi, à tort comme l'acronyme de « WINdows Emulator ». Ce logiciel est une implémentation libre de l'interface de programmation Microsoft Windows bâtie sur X et UNIX (BSD, Linux), c’est-à-dire qu'il permet d'utiliser sous Linux ou Mac OS X des programmes conçus pour fonctionner sous Windows. Le logiciel n'a donc pas besoin du système d'exploitation Windows pour fonctionner. En cela, Wine se différencie des émulateurs de machine comme QEMU et Bochs. Wine gère les modes 16 et 32 bits de l'interface Windows. Wine est maintenant sous licence LGPL, après avoir été sous licence WineHQ, puis X11.

Il fournit à la fois les outils de développement (Winelib) pour porter du code source Windows vers Unix, et un chargeur de programmes permettant à de nombreux binaires de fonctionner sans modifications.

Plus de détails : http://www.winehq.org/

JAVA VIRTUAL MACHINE

La Java virtual machine (abrégé JVM, en français machine virtuelle Java) est une machine virtuelle permettant d'interpréter et d'exécuter le bytecode Java.
Architecture générale : illustration du slogan Compile once, run everywhere

Ce programme est spécifique à chaque plate-forme ou couple (machine/système d'exploitation) et permet aux applications Java compilées en bytecode de produire les mêmes résultats quelle que soit la plate-forme, tant que celle-ci est pourvue de la machine virtuelle Java adéquate.

La machine virtuelle la plus utilisée est celle de Sun Microsystems. Elle est gratuite, propriétaire jusqu'à la version 6 (stable) et libre à partir de la version 7 (non encore officielle).

Le 11 novembre 2006, Sun Microsystems a publié les sources de sa machine virtuelle HotSpot et de son compilateur javac sous licence Open Source GPL.

Plus de détails : http://java.sun.com

Perspectives & Tendances

VIRTUALISATION DES DESKTOPS

La virtualisation des postes de travail est la nouvelle tendance du marché. Elle promet de réduire les coûts d'exploitation des parcs informatiques. L'offre est pléthorique qu'il s'agisse de solutions en mode connecté, hors réseau ou spécialisées.

Après la virtualisation des serveurs, les entreprises adoptent actuellement la virtualisation des postes de travail, pour les mêmes raisons : flexibilité, coûts moins élevés et facilité de supervision.

L'environnement de travail (système d'exploitation, applications et les données) est hébergé sur la machine hôte. Les utilisateurs accèdent à leur environnement via tout client compatible (ordinateur traditionnel, portable, smartphone, client léger ...).

Conclusion

CONCLUSION

Bonne pratique pour dans le processus de virtualisation des serveurs :

- Commencer petit, mais penser loin
- Requiert un investissement lourd au début, mais envisager le retour sur investissement
- Définisser votre stratégie de stockage premièrement
- Virtualiser les bonnes applications – virtualiser celles qui sont critiques et identifier celles qu'il ne faut pas virtualiser
- Combiner les machines virtuelles efficacement sur les machines hôtes.

Questions ? / Réponses