Smile
MOTORISTES INTERNET

Livre blanc

Virtualisation

Equipe Administration Système Smile

Version 1.0
Pour plus d'information : www.smile.fr
Tél : 01 41 40 11 00
Mailto : sdc@smile.fr

PREAMBULE

Smile

Fondée en 1991, Smile est une société d'ingénieurs experts dans la mise en œuvre de solutions Internet et intranet. Smile compte 220 collaborateurs (au 1er septembre 2007).

Le cœur de métier de Smile couvre trois grands domaines :

- La conception et la réalisation de sites Internet haut de gamme. Smile a construit quelques uns des plus grands sites du paysage web français, avec des références telles que Cadremploi ou Explorimmo.
- Les applicatifs Intranet, qui utilisent les technologies du web pour répondre à des besoins métier. Ces applications s'appuient sur des bases de données de grande dimension, et incluent plusieurs centaines de pages de transactions. Elles requièrent une approche très industrielle du développement.
- La mise en œuvre et l'intégration de solutions prêtes à l'emploi, dans les domaines de la gestion de contenus, des portails, du commerce électronique, du CRM et du décisionnel.

Dans chacun de ces domaines, l'équipe système de Smile intervient pour

- Aider aux choix d'architectures
- Mettre en place les plateformes d'hébergement, souvent dans des configurations hautes performances et haute disponibilité.
- Conduire des tests de charge et l'optimisation des performances
- Mettre en œuvre le monitoring et l'exploitation des sites confiés à Smile, et assurer la mise à niveau de ces plateformes.

Au travers de toutes ces missions, les administrateurs système de Smile ont acquis une maîtrise opérationnelle complète des solutions de virtualisation.

Ils vous offrent ici un échantillon de leur savoir faire.

Quelques références de Smile

Intranets et Extranets

Société Générale - Caisse d'Épargne - Bureau Veritas - Commissariat à l'Energie Atomique - Visual - CIRAD - Camif - Lynxial - RATP - Sonacotra - Faceo - CNRS - AmecSpie - INRA - CTIFL - Château de Versailles - Banque PSA Finance - Groupe Moniteur - Vega Finance - Ministère de l'Environnement - Arjowiggins - JCDecaux - Ministère du Tourisme - DIREN PACA - SAS - CIDJ - Institut National de l'Audiovisuel - Cogedim - Diagnostica Stago - Ecureuil Gestion - Prolea - IRP-Auto - Conseil Régional Ile de France - Verspieren - Conseil Général de la Côte d'Or - Ipsos - Bouygues Telecom - Prisma Presse - Zodiac - SANEF - ETS Europe - Conseil Régional d'Ile de France - AON Assurances & Courtage - IONIS - Structis (Bouygues Construction) - Degrémont Suez - GS1-France - DxO - Conseil Régional du Centre - Beauté Prestige International - HEC - Veolia

Internet, Portails et e-Commerce

Cadremploi.fr - chocolat.nestle.fr - creditlyonnais.fr - explorimmo.com - meilleurtaux.com - cogedim.fr - capem.fr - Editions-cigale.com - hotels-exclusive.com - souriau.com - pci.fr - odit-france.fr - dsv-cea.fr - egide.asso.fr - Osmoz.com - spie.fr - nec.fr - vizzavi.fr - sogeposte.fr - ecofi.fr - idtgv.com - metro.fr - stein-heurtey-services.fr - bipm.org - buitoni.fr - aviation-register.com - cci.fr - eaufrance.fr - schneider-electric.com - calypso.tm.fr - inra.fr - cnil.fr - longchamp.com - aesn.fr - bloom.com - Dassault Systemes 3ds.com - croix-rouge.fr - worldwatercouncil.org - Projectif - credit-cooperatif.fr - editionsbussiere.com - glamour.com - nmmedical.fr - medistore.fr - fratel.org - tiru.fr - faurecia.com - cidil.fr - prolea.fr - bsv-tourisme.fr - yves.rocher.fr - jcdecaux.com - cg21.fr - veristar.com - Voyages-sncf.com - prismapub.com - eurostar.com - nationalgeographic.fr - eau-seine-normandie.fr - ETS Europe - LPG Systèmes - cnous.fr - meddispar.com - Amnesty International - pompiers.fr - Femme Actuelle - Stanhome-Kiotis - Gîtes de France - Bouygues Immobilier - GPdis - DeDietrich - OSEO - AEP - Lagardère Active Média - Comexpo - Reed Midem - UCCIFE - Pagesjaunes Annonces - 1001 listes - UDF - Air Pays de Loire - Jaccede.com - ECE Zodiac - Polytech Savoie - Institut Français du Pétrole - Jeulin - Atoobi.com - Notaires de France - Conseil Régional d'Ile-de-France - AMUE

Applications métier et applications décisionnelles

Renault - Le Figaro - Sucden - Capri - Libération - Société Générale - Ministère de l'Emploi - CNOUS - Neopost - Industries - ARC - Laboratoires Merck - Egide - Bureau Veritas - ATEL-Hotels - Exclusive Hotels - CFRT - Ministère du Tourisme - Groupe Moniteur - Verspieren - Caisse d'Epargne - AFNOR - Souriau - MTV - Capem - Institut Mutualiste Montsouris - Dassault Systèmes - Gaz de France - CAPRI Immobilier - Croix-Rouge Française - Groupama - Crédit Agricole - Groupe Accueil - Eurordis - Mindscape - Xinek - Institut National de l'Audiovisuel - CDC Arkhineo

Ce livre blanc

Bien qu'elles soient arrivées à maturité assez récemment, les solutions de virtualisation ont très rapidement conquis le monde de l'administration système et des infrastructures d'hébergement, comme de développement.

C'est qu'elles apportent des bénéfices considérables, tant dans l'optimisation des coûts que dans la flexibilité de l'exploitation.

Pratiquant les différents solutions de virtualisation depuis leurs débuts, les administrateurs système de Smile les ont mises en œuvre dans une variété de contextes, et en maîtrisent toutes les possibilités, de même qu'ils en connaissent les difficultés.

Ce livre blanc s'efforce de réunir :

- Une présentation générale des concepts de la virtualisation de serveurs, et de ses champs d'application.
- Un recensement des solutions du marché, qui sont majoritairement open source, avec un focus particulier sur les plus matures.
- Un retour d'expérience sur le déploiement de ces outils dans différents contextes.

Enfin, un tableau comparatif fait la synthèse des fonctionnalités présentes dans les différents outils.

SOMMAIRE

Les Principes de la Virtualisation

Partage d'un serveur

Un serveur est une ordinateur utilisé à distance depuis différents postes de travail, ou autres ordinateurs. 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.

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.

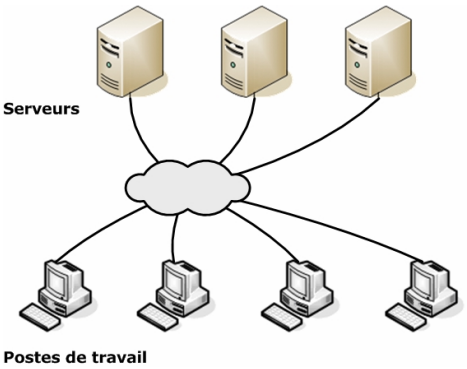
Le principe de la virtualisation est donc un principe de *partage* : les différents systèmes d'exploitation se partagent les ressources du serveur.

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

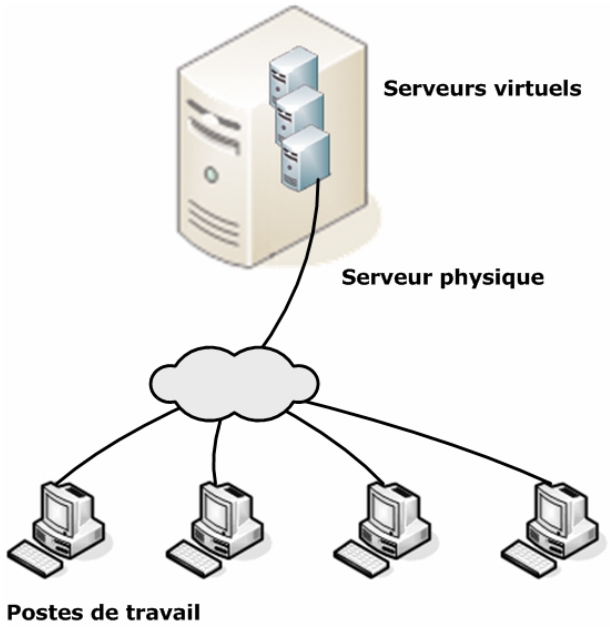
- Le *cloisonnement* : chaque système d'exploitation a un fonctionnement indépendant, et ne peut interférer avec les autres en aucune manière.
- La *transparence* : le fait de fonctionner en mode virtualisé ne change rien au fonctionnement du système d'exploitation et a fortiori des applications.

La transparence implique la compatibilité: toutes les applications peuvent tourner sur un système virtualisé, et leur fonctionnement n'est en rien modifié.

Pour ce qui est du cloisonnement, il existe bien sûr une interférence passive liée à la concurrence dans le partage des ressources. Mais nous verrons que ce partage peut être parfaitement contrôlé.



Architecture traditionnelle



Architecture virtualisée

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 ?

A cela, deux réponses.

La première relève de la rigueur du cloisonnement, au sein d'un même système, entre les différents contextes de travail. Le fonctionnement natif de la plupart des systèmes ne permet pas un cloisonnement suffisamment étanche. Nous verrons qu'une des voies de la virtualisation consiste à renforcer le cloisonnement.

La seconde relève du système d'exploitation lui-même, et des configurations système.

Il arrive couramment que 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.

Dans tous ces cas de figure, le partage de ressources offert par le système lui même ne convient plus : on veut partager les ressources *en dessous* du système d'exploitation, de manière à faire cohabiter plusieurs systèmes d'exploitation sur le même serveur physique.

Objectifs et bénéfices

Le premier objectif de la virtualisation est économique.

Partager les ressources physiques dont on dispose entre différents serveurs virtuels, permet de ne pas acheter plusieurs serveurs physiques, lorsqu'un seul a une capacité suffisante en termes de ressources.

Le constat sous-jacent est que les serveurs sont souvent sous-utilisés.

Parce que les serveurs commercialisés correspondent à un « quantum » minimal de puissance ; vous pouvez certes ajuster la configuration mémoire, mais si votre besoin est seulement un dixième de processeur, il vous faut un processeur entier, et donc un serveur entier, qui sera alors sous-utilisé.

Par ailleurs, les besoins d'une application donnée peuvent varier dans le temps de manière extraordinaire. Soit à court terme, avec les heures de pointes dans une même journée. Soit sur le long terme, avec par exemple un environnement de développement mis en sommeil pour plusieurs mois, puis à nouveau utilisé pour une opération de maintenance.

Bien sûr, on peut aussi partager un serveur dans le temps, en sauvegardant toute la configuration logicielle, y compris le système d'exploitation, et en installant un autre système pour une autre utilisation. C'est en fait ce que l'on faisait avant la virtualisation pour remettre en place l'environnement de développement d'un projet ancien afin d'y faire une opération de maintenance. La réinstallation d'un système complet est une opération lourde, qui peut prendre plusieurs heures, et présente un risque de petites variations de configuration. Aujourd'hui, on préfère conserver un environnement virtualisé prêt à l'emploi, qui ne consommera pratiquement aucune ressource, si ce n'est une part de l'espace disque.

Eviter de multiplier les serveurs physiques apporte des bénéfices en termes de *coût d'acquisition*, bien entendu, mais aussi en termes de *coût de possession*, tant au niveau de l'hébergement (rack, électricité, refroidissement, câblage, interfaces réseau), que de l'exploitation.

Mais la virtualisation apporte aussi des bénéfices qui ne sont pas directement liés au partage des ressources.

Ainsi, la virtualisation permet de déplacer un serveur virtuel d'un hôte à un autre de manière très aisée, y compris sur des environnements matériels très hétérogènes, puisque les couches matérielles dans les serveurs virtuels sont le plus souvent génériques.

Cette capacité à agencer aisément et rapidement la répartition des serveurs virtuels sur un parc de serveurs physiques est évidemment une révolution dans l'administration d'un parc de serveurs.

Avec certaines solutions de virtualisation, le déplacement s'effectue de manière totalement transparente pour le système invité, et donc sans arrêter les applications ni faire un shutdown. Le délai de ce déplacement n'est dicté que par le temps de transfert de l'espace disque, et nous verrons qu'il peut être pratiquement nul dans certaines configurations, ce qui permet des transferts totalement transparents opérés à chaud.

Historique

Le besoin de partager les ressources physiques pour une utilisation optimale est bien sûr d'autant plus fort que ces ressources sont coûteuses, et c'était donc un domaine de recherche important dès les débuts de l'informatique transactionnelle.

La capacité à gérer plusieurs utilisateurs simultanément, en séparant leurs contextes de travail, est apparue dès les années 70, et s'est généralisée dans les années 80 avec les grands moniteurs transactionnels, tels que CICS.

Chaque utilisateur dialogue avec le serveur de manière indépendante, comme s'il était seul, et utilise donc une petite part des ressources du serveur, selon son besoin. Néanmoins, cette séparation de contextes utilisateur, que l'on retrouve bien sûr aujourd'hui avec les serveurs Http et les outils serveurs d'application du web, n'est pas appelée virtualisation. En effet, si le contexte applicatif est propre à chaque utilisateur, le contexte logiciel est au contraire parfaitement homogène.

IBM figure dans les pionniers de ces technologies avec l'hyperviseur CM/CMS utilisé dès les années 60, qui fut le père de VM/CMS dans les années 70, devenu aujourd'hui z/VM, qui permet de faire tourner y compris AIX ou Linux au sein d'une machine virtuelle sur mainframe.

Dans la seconde moitié des années 1990, le monde de la micro-informatique découvre les émulateurs. La puissance des machines x86 leur permet d'émuler les générations précédentes de machines. Il devient alors possible d'émuler des machines Atari, Amiga, Amstrad ainsi que de nombreuses consoles.

A la fin des années 1990 la société VMware développe et popularise le produit du même nom, système propriétaire de virtualisation logicielle des architectures de type Intel x86, ouvrant la possibilité de mettre en place n'importe quel environnement x86 à des fins de tests ou de développement sans avoir besoin d'acheter une nouvelle machine. Contrairement aux émulateurs cités précédemment, il est enfin possible de faire tourner les applications professionnelles destinés aux processeurs x86 dans une machine virtuelle.

Il faut citer aussi aux rangs des précurseurs, Qemu, créé par Fabrice Bellard, qui a ouvert la voie et sur lequel se sont appuyées la plupart des solutions open source.

Viennent ensuite les logiciels libres comme Xen, Bochs, Linux-VServer, que nous décrirons plus en détail dans ce document.

Et pour finir les logiciels propriétaires VirtualPC et VirtualServer ont achevé la popularisation de la virtualisation dans le monde x86.

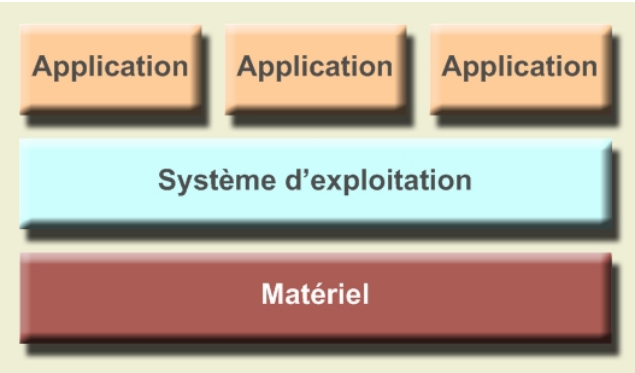
Pour répondre aux nouveaux défis de la virtualisation, notamment en terme de performances, les fabricants de processeurs x86, AMD et Intel, ont implémenté dans leurs gammes de processeurs des instructions spécifiques améliorant les possibilités de virtualisation. Ces processeurs ont commencé à être diffusés à partir de 2006. Ils permettent une virtualisation complète avec un rendement proche de 100%.

Un peu de vocabulaire

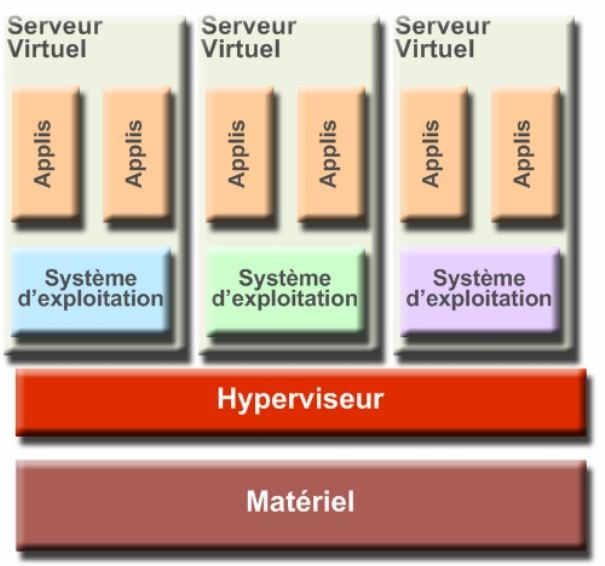
Hyperviseur

L'hyperviseur est la couche logicielle qui s'insère entre le matériel et les différents systèmes d'exploitation. C'est bien un composant clé, que l'on retrouve dans la plupart des technologies de virtualisation de bas niveau.

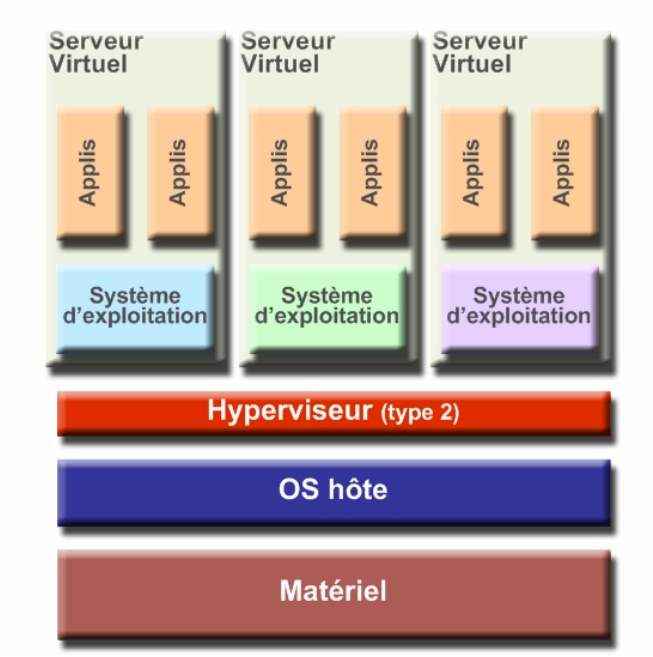
Ainsi, par rapport au schéma de base d'un serveur distinguant le matériel, le système d'exploitation, et ses applications :



L'hyperviseur vient s'insérer entre le matériel et plusieurs systèmes d'exploitation, de la manière suivante :



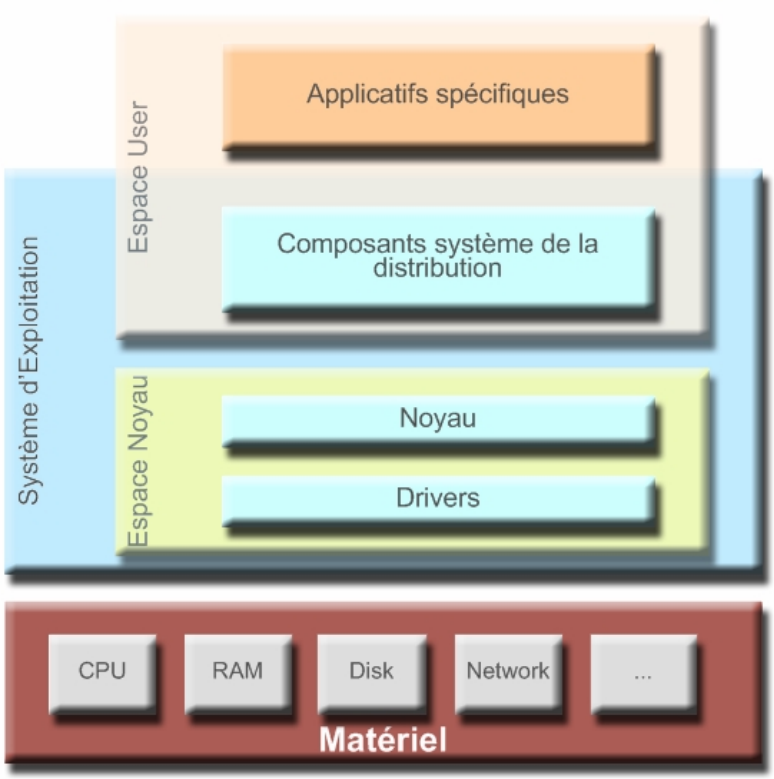
L'hyperviseur peut soit gérer lui-même toutes les ressources matérielles du serveur, soit s'appuyer pour cela sur un système d'exploitation existant. Dans ce dernier cas, on parle d'hyperviseur de type II, comme figuré ci-après :



Espace noyau, espace user

Rappelons que l'on distingue, dans un serveur deux espaces :

- L'espace noyau (*kernel space*), qui inclut le noyau du système d'exploitation et ses drivers.
- L'espace utilisateur (*user space*), qui inclut tout le reste, incluant tous les composants systèmes de la distribution ainsi que les applicatifs spécifiques.



OS hôte, OS invité

En virtualisation, on appelle « OS hôte » ou *Host OS*, l'OS sous-jacent, sur lequel s'appuie l'hyperviseur (de type II, donc).

On appelle « OS invité » ou *Guest OS*, les OS des machines virtuelles,

Emulation

L'émulation consiste à simuler l'exécution d'un programme en interprétant chacune des instructions destinées au micro-processeur. Il est possible d'émuler ainsi n'importe quel processeur et l'environnement complet d'un serveur.

On a vu apparaître ainsi, dans les années 90, des émulateurs reproduisant fidèlement les premiers micro-ordinateurs tels que Amiga, ou Atari.

L'émulation est la technique qui offre le plus haut niveau d'abstraction de la plateforme. Il faut rappeler en effet que toutes les autres techniques de virtualisation citées ont une exigence en commun : tous les exécutables doivent être compilés pour le processeur physiquement disponible sur le serveur.

L'émulation lève cette contrainte car les instructions ne sont jamais exécutées par le processeur, elles sont interprétées en simulant le processeur.

Cette interprétation est coûteuse en performances, de sorte que l'émulation est rarement utilisée en dehors d'applications ludiques ou de recherche. Dans le cas de l'émulation des vieux Atari, le différentiel de puissance des processeurs sur 10 ans comblait largement la perte résultant de l'émulation.

Le projet QEMU est une solution open source de virtualisation par émulation.

Performances et rendement

A l'évidence, puisqu'il y a partage des ressources physiques, chaque environnement virtuel dispose de ressources plus limitées que s'il avait un serveur physique dédié.

Mais la question essentielle est : la somme des ressources allouées aux différents environnements virtuels est-elle égale aux ressources physiques disponibles ? Autrement dit : Quel est le surcoût (« *overhead* ») de la virtualisation ? On pense en particulier au surcoût en termes de CPU, car les autres ressources sont en général moins précieuses.

Les bonnes solutions de virtualisation, appuyées sur des processeurs disposant d'instructions spécialisées, permettent un surcoût en

performances qui est aujourd'hui négligeable, c'est à dire que le rendement est pratiquement égal à 1.

➔ En d'autres mots : la mise en œuvre d'environnements virtualisés n'implique pas de perte de performances.

Il faut souligner aussi que dans certaines applications de la virtualisation, de nombreux environnement peuvent être *dormants*, en attendant un usage futur. Dans ce cas leur consommation de ressource CPU est à peu près nulle, et leur consommation de RAM est très faible.

Sécurité

Dans la pratique, la virtualisation n'apporte aucune dégradation en termes de sécurité.

Certes, la sécurité du serveur physique sous-jacent est critique, car un accès console sur ce serveur, ou sur l'hyperviseur de la solution de virtualisation pourrait compromettre l'ensemble des serveurs virtuels hébergés. Il est donc évidemment primordial de préserver ce niveau de sécurité, et donc de bien distinguer en termes d'habilitations, l'administration *du serveur physique et de la virtualisation* d'une part, et l'administration *des environnements virtualisés* d'autre part.

A l'inverse, le contrôle administrateur (*root*) sur l'un des environnements ne donne aucun droit, ni aucune possibilité, même pour un intervenant malveillant, ni sur l'environnement physique et l'hyperviseur, ni sur les autres environnements.

Enfin, la bonne pratique, scrupuleusement appliquée par les bons administrateurs, est de placer les serveurs physiques dans des réseaux différents, ce qui les rend inaccessibles depuis l'extérieur.

Administration

Si la virtualisation est transparente pour les utilisateurs, pour les applications, et même pour les systèmes d'exploitation invités, elle ne l'est pas bien sûr pour l'administrateur qui en a la charge.

La mise en œuvre et l'exploitation des solutions de virtualisation requièrent une vraie expertise. Pour un administrateur système de bon niveau, maîtriser une solution de virtualisation demandera plusieurs jours de formation, et quelques semaines de pratique.

Les solutions open source de virtualisation ont un packaging moins abouti, et ne fournissent pas d'outils graphiques aussi avancés que leurs concurrents propriétaires. Mais même si l'apprentissage des outils d'administration en ligne de commande nécessite un certain niveau de formation, ils permettent une maîtrise plus importante et plus grande souplesse d'utilisation.

Contrôle des ressources

Une des grandes problématiques dans un environnement virtualisé est le contrôle dans l'attribution et dans le partage des ressources du serveur physique.

On peut souhaiter répartir les ressources disponibles soit de façon équitable, soit en privilégiant certains environnements par rapport aux autres.

Les règles dépendent bien sûr du domaine d'application. Si 10 sites Internet se partagent un serveur physique et que l'un connaît un pic de trafic, on peut souhaiter lui laisser prendre 90% de la CPU tant que les autres n'en ont pas usage. A l'inverse, si un hébergeur a vendu 1/10ème de serveur à l'un de ses clients, il doit être en mesure de garantir que le client aura toujours son quota, quelle que soit la demande des autres clients.

Dans tout les cas les différents produits de virtualisation implémentent des mécanismes permettant d'assurer cette répartition, et d'éviter qu'un serveur ne pénalise les autres en consommant toute les ressources de la machine physique sur laquelle ils s'exécutent.

Les quatre ressources principales que l'on souhaite généralement contrôler sont :

- CPU : un ordonnanceur spécifique est généralement en charge de répartir la charge du ou des processeurs entre les différents serveurs virtuels. La plupart des technologies permettent d'attribuer des poids, privilégiant ainsi un serveur par rapport à l'autre ce qui permet d'assurer un minimum de puissance disponible, tout en tirant profit des ressources maximales de la machine physique.

- Mémoire : la mémoire est la ressource la mieux maîtrisée par l'ensemble des technologies de virtualisation. La mémoire que l'on souhaite attribuer à un serveur virtuel est souvent réservé à la création.

- Stockage : les différents produits de virtualisation peuvent s'appuyer sur différent types de stockage, adaptés à différentes échelles, tels qu'un simple répertoire, une image binaire d'un disque dur, ou un volume logique dans un SAN. L'espace disque disponible est connu à l'avance et peut être limité.

- Réseau : c'est la ressource la moins bien gérée par les technologies actuelles de virtualisation. Dans les produits présentés ici, aucune limitation de bande passante réseau n'est possible. En revanche, contrairement aux autres ressources, il est possible de contrôler le réseau en amont, au moyen d'un routeur implémentant des technologies de Qualité de Service.

ÉTAT DE L'ART

On distingue trois grandes catégories de solutions de virtualisation :

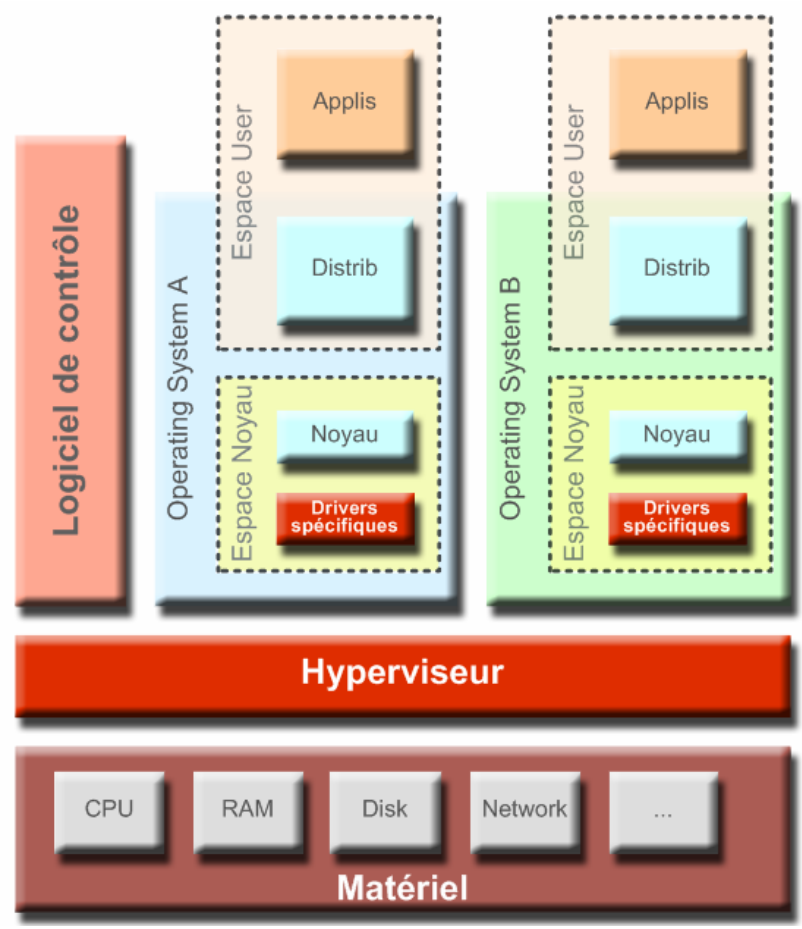
- L'isolation
- La paravirtualisation
- La virtualisation complète

L'isolation consiste à mettre en place, *sur un même noyau de système d'exploitation*, une séparation forte entre différents contextes logiciels. Ce n'est pas de la virtualisation à proprement parler, mais nous verrons que c'est une solution qui peut rendre les mêmes services.

La paravirtualisation et la virtualisation complète sont assez proches. Elles s'appuient sur une couche hyperviseur, qui gère totalement l'interface avec les ressources matérielles, et sur laquelle on peut installer différents systèmes d'exploitation.

La paravirtualisation présente aux systèmes d'exploitation une machine générique spéciale, qui requiert donc des interfaces spéciales intégrées aux systèmes invités, sous la forme de drivers.

Dans la virtualisation complète, l'hyperviseur intercepte de manière transparente tous les appels que le système d'exploitation peut faire aux ressources matérielles, et supporte donc des systèmes invités non-modifiés.



Isolation

Présentation

L'isolation (aussi appelé cloisonnement) est une technique qui intervient au sein d'un même système d'exploitation. Elle permet de séparer un système en plusieurs *contextes* ou *environnements*. Chacun d'entre eux est régi par l'OS hôte, mais les programmes de chaque contexte ne sont capables de communiquer qu'avec les processus et les ressources associées à leur propre contexte.

Il est ainsi possible de partitionner un serveur en plusieurs dizaines de contextes, presque sans ralentissement.

L'isolation est utilisée sous Unix depuis longtemps pour protéger les systèmes. Via des mécanismes comme *chroot* ou *jail* il est possible d'exécuter des applications dans un environnement qui n'est pas celui du système hôte, mais un « mini système » ne contenant que ce dont

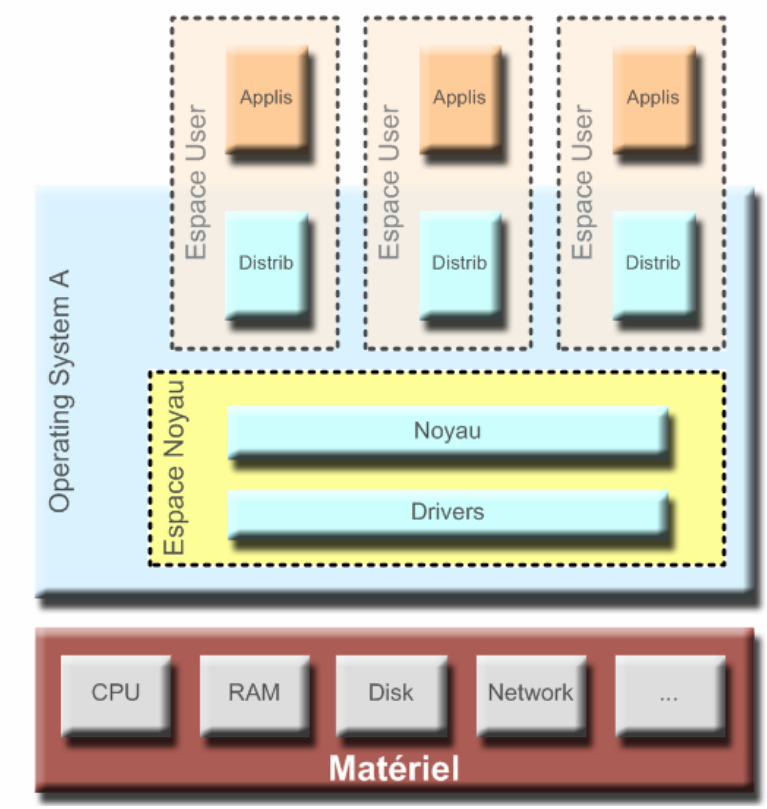
l'application a besoin, et n'ayant que des accès limités aux ressources. Il est possible également de lancer des programmes dans une autre distribution que celle du système principal.

Avec l'isolation, l'espace noyau n'est pas différencié, il est unique, partagé entre les différents contextes. Mais on définit de multiples espaces utilisateurs cloisonnés. C'est ainsi que l'on peut faire cohabiter différentes distributions de système d'exploitation, à condition qu'elles partagent le même noyau.

L'isolation des contextes est une solution légère, tout particulièrement dans les environnements Linux.

L'unicité du noyau reste bien sûr une petite limitation. D'une part en termes de robustesse, puisqu'un plantage du noyau – fort heureusement très rare dans le monde Linux – plante simultanément tous les environnements. D'autre part dans les utilisations possibles, puisque typiquement ce mode ne conviendra pas pour valider une nouvelle version de noyau.

Mais pour les besoins les plus courants de la virtualisation, la simplicité de mise en œuvre et le faible overhead sont d'excellents arguments.



Les solutions

La principale solution pour l'isolation est Linux-VServer, la plus mature et la plus avancée, que nous détaillerons plus loin.

OpenVZ est une alternative, qui se présente de la même façon et propose quasiment les mêmes fonctionnalités. Elle est à la base du produit commercial Virtuozzo, mais présente peu d'intérêt à nos yeux.

Paravirtualisation

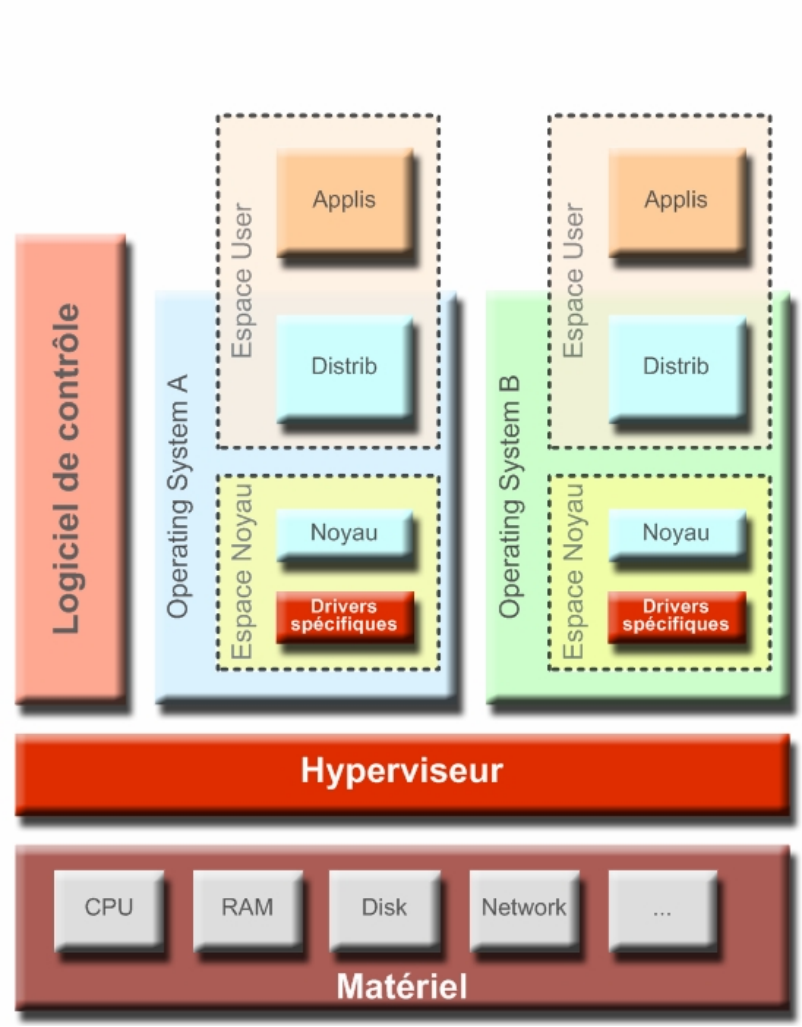
Présentation

La paravirtualisation est une technique de virtualisation de plus bas niveau que l'isolation. Elle partage avec cette dernière la nécessité d'utiliser un OS modifié. Plus précisément, en paravirtualisation ce n'est plus seulement l'OS hôte qui doit être modifié mais également les OS appelés à s'exécuter sur les environnements virtuels.

Le cœur de la paravirtualisation est un hyperviseur fonctionnant au plus près du matériel, et fournissant une interface qui permet à plusieurs systèmes hôtes d'accéder de manière concurrente aux ressources.

Chaque système virtuel doit donc être modifié de façon à utiliser cette interface pour accéder au matériel, en revanche, contrairement à l'isolation, plusieurs OS de familles différentes peuvent fonctionner sur un même serveur physique. Il est ainsi possible de faire fonctionner GNU/Linux, NetWare, Solaris (et d'autres) simultanément sur une même machine. Chaque OS aura alors accès à ses propres périphériques de stockage, sa propre mémoire, sa ou ses propres interfaces réseau, son ou ses propres processeurs, chaque ressource matérielle virtualisée étant partagée avec les autres environnements.

La nécessité de petites modifications au système d'exploitation invité exclut le support de systèmes « fermés », et en particulier de Microsoft Windows.



Xen

Xen est actuellement le seul représentant mature de la technologie de paravirtualisation dans le monde Open Source, les autres solutions étant propriétaires et plutôt réservées aux mainframes. Xen permet de paravirtualiser des systèmes GNU/Linux dont le noyau aura été spécialement patché.

Virtualisation complète

Présentation

Au sens strict, le terme 'machines virtuelles' désigne les systèmes virtuels exécutés via des technologies de virtualisation dites *complète* ou encore *native*.

Dans ce cas de figure, c'est le matériel d'un ordinateur complet qui est présenté au système d'exploitation par le produit, de sorte que la virtualisation est alors réellement transparente pour le système d'exploitation invité.

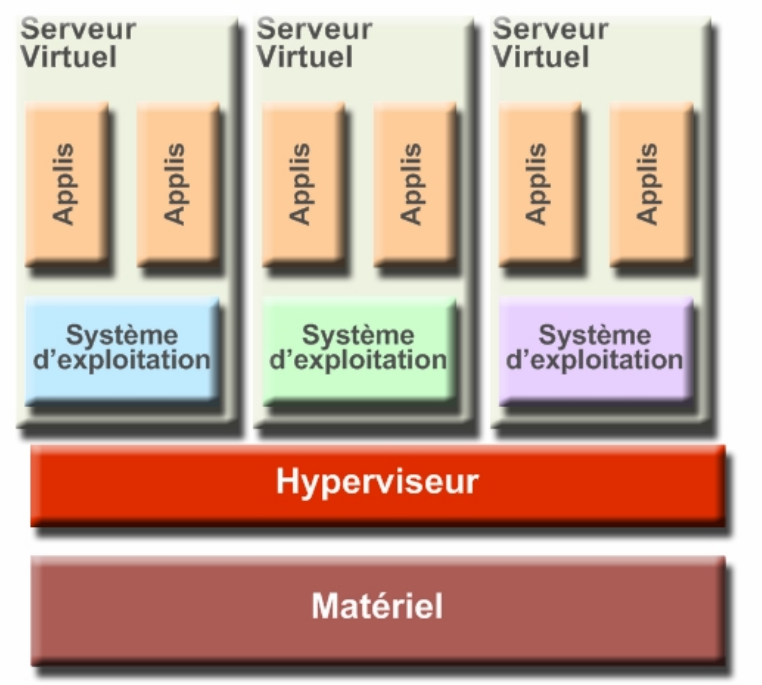
Cela permet donc de faire fonctionner plusieurs systèmes d'exploitation non modifiés sur un serveur physique. Le matériel du serveur physique est rendu abstrait et remplacé, du point de vue des serveurs virtuels, par un matériel 'générique' (en général propre au produit de virtualisation).

Les premières solutions de virtualisation complète étaient basées sur des émulateurs, donc des logiciels qui réinterprétaient chaque opération demandée par le système virtuel, pour les adapter au matériel physique, au prix d'une perte considérable de performances.

Petit à petit, la partie ré-interprétation est passée de l'espace utilisateur (des programmes) à l'espace noyau, regagnant une partie des performances d'origines, et les logiciels d'interface matérielle ont été remplacés par des hyperviseurs pour gagner en proximité avec le matériel physique.

Les produits modernes tirent partie des nouveaux jeux d'instructions spécialisés des dernières générations de processeurs pour assurer des performances quasi identiques aux performances natives, alors que l'interface matérielle est gérée au plus bas niveau par un hyperviseur.

Sur une machine virtuelle, il est possible d'installer n'importe quel OS non modifié, et donc aussi bien commercial qu'open source du moment qu'il dispose des pilotes pour le matériel générique que lui présente l'hyperviseur.



QEMU

QEMU a été un des premiers projets libres à proposer des performances quasi natives. A l'origine QEMU est un émulateur, et, bien qu'il ait intégré un système de compilation en temps réel vers le processeur cible, souffrait d'une grosse baisse de performance des machines virtuelles. Depuis peu, il dispose d'un module d'accélération pour le noyau Linux qui permet d'obtenir des performances natives, en virtualisant l'accès aux ressources matérielles. Aujourd'hui QEMU n'est plus une technologie privilégiée, mais une bonne partie du code de la couche de virtualisation matérielle de ce projet est encore utilisé par les solutions modernes.

Xen

Déjà présenté dans la partie précédente, Xen est un produit versatile qui propose, outre des possibilités de paravirtualisation, un mode virtualisation native.

Ce mode, appelé HVM-Xen, est très différent de la paravirtualisation et ne nécessite donc plus la modification des systèmes d'exploitations appelés à fonctionner dans les machines virtuelles.

HVM-Xen permet d'utiliser des systèmes d'exploitation fermés comme Microsoft Windows, mais aussi GNU/Linux avec un niveau

d'abstraction matérielle supérieur à celui proposé par le mode paravirtualisé.

Il est cependant nécessaire de disposer d'un processeur fournissant les jeux d'instructions d'aide à la virtualisation (Intel VT ou AMD Pacifica) pour pouvoir bénéficier de la virtualisation native sous Xen.

KVM

KVM est un nouveau venu sur la scène des machines virtuelles open-source. Il se présente sous la forme d'un module pour le noyau Linux permettant d'offrir une virtualisation native en utilisant les extensions de virtualisation des processeurs. Il fait partie depuis peu du noyau Linux officiel de Linus Torvalds.

Bien que très prometteur, KVM en est encore à ses débuts et ne dispose pas encore des outils d'administrations qui font l'intérêt de ses concurrents.

LES PRINCIPALES SOLUTIONS

Linux-VServer

Une des solutions les plus avancées et les plus matures dans le domaine de l'isolation est *Linux Vserver.*

Ce produit se présente sous la forme d'un patch pour le noyau Linux, et d'un ensemble d'outils d'administration. Le patch du noyau permet à un système GNU/Linux de gérer des contextes virtualisés. Les outils d'administration permettent de créer, d'instancier, et de contrôler les environnements virtuels.

La plupart des distributions Linux proposent des versions packagées de Linux Vservers. En particulier, la distribution *Debian GNU/Linux* permet dès l'installation du serveur physique de mettre en place cette solution puisqu'elle fournit un noyau pré-patché et un package permettant d'installer les outils d'administration en quelques secondes.

Présentation

Le projet Linux-VServer fournit aux systèmes GNU/Linux une méthode de virtualisation. Cette virtualisation se situe au niveau du noyau de l'OS. Cela rend possible l'exécution de multiple instances d'OS GNU/Linux sur la même machine. Ces instances fonctionnant de façon complètement sécurisées et partageant intelligemment les ressources du serveur hôte.

Historique

Le projet VServer a été lancé par Jacques Gélinas quelques temps avant 2001, ce n'est que à partir de 2003 que le projet prends l'ampleur avec une grande augmentation du nombre de développeurs et la nomination de Herbert Pötzl à la tête du projet. En 2003 cette équipe sort la version 1 de VServer. Toute une suite de services collaboratifs accompagne maintenant l'outil (mailing list, wiki ...).

Depuis le projet n'a cessé d'évoluer, de nouvelles distribution sont disponibles régulièrement, le support de la communauté s'est amélioré, et les outils ont été ré développé plus proprement.

Linux Vserver est donc aujourd'hui l'un des outils les plus matures, stable, et complet au plan fonctionnel.

Principe

Linux VServers est un isolateur de contexte. Il est capable d'isoler le contexte d'exécution de plusieurs OS sur la même machine. Nativement un noyau Linux ne permet aux processus que de tourner dans un seul contexte commun. Le patch noyau de ce projet open-source permet d'ajouter au noyau un ensemble d'outils pour isoler ces contextes. On peut alors faire tourner plusieurs OS de façon complètement isolé sur la même machine. Toutefois ils partagent quand même le même noyau, ce qui veut dire que dans l'éventualité où celui ci se retrouve en défaut ce sont tous les contextes qui le sont.

Linux Vservers est aussi capable d'isoler les contextes réseau, c'est à dire que les contextes ne voient pas le trafic des autres contextes de la machine hôte, tout comme la machine hôte en elle-même. Cette parfaite étanchéité en terme de sécurité et de confidentialité des données traversant les machines virtuelles et la machine hôte, est particulièrement appréciable, surtout dans le cas d'un hébergement.

En fait, le projet Linux-VServer découle des outils déjà existants Linux comme la barrière chroot, les limitations de ressources mais le tout intégré à un niveau plus bas de l'OS, plus développé et accompagné d'un ensemble d'outils pour administrer l'ensemble.

Limitations

Du fait que les serveurs virtuels utilisent le noyau de l'OS hôte Linux Vserver est incapable de faire tourner d'autres OS que GNU/Linux; en revanche cela lui confère un rendement proche des performances natives, en effet la couche "vserver" est très fine et permet de positionner les OS virtualisés au plus proche du noyau (on considère que la charge provoqué par l'isolation est inférieur à 1% des capacités de la machine. Les performances obtenues sont donc supérieures à 99% des performances natives).

Certaines fonctions nécessitant un accès direct au noyau par une des machines virtuelles, en particulier réseau (filtrage, routage, ...) sont désactivées par défaut, pour des raisons de sécurité.

Il est possible de configurer la machine virtuelle de façon à lui ajouter des "capacités", c'est à dire de donner des droits vis à vis du noyau de la machine hôte. Cela n'est toutefois pas recommandé du fait des risques de sécurité que cela peut engendrer. Si le besoin existe, il est

préférable de se tourner vers des solutions comme Xen qui donnent un noyau "invité" aux OS virtualisés.

Un autre point faible est l'impossibilité de migrer à chaud un Vserver, il est nécessaire de l'arrêter, le copier et le relancer. Avec Vserver, il faut un shutdown du système invité, comme sur un serveur physique.

De plus, la machine virtuelle ne dispose pas d'interface réseau de boucle locale (localhost:127.0.0.1) par défaut, ce qui peut nécessiter une configuration particulière pour certain programme, même si le patch Linux-Vserver redirige toute les demandes de bind à destination du 127.0.0.1 vers l'IP de la machine virtuelle.

Evolutions

D'un point de vue réseau, la couche actuelle présente encore quelques limitations: les vservers ne possèdent pas une vraie interface réseau mais simplement une fraction isolée de celles de l'hôte, cela peut être problématique dans certains cas. Par exemple il est impossible de créer des "virtualswitch" comme avec des solutions du genre VMWare ESX. En fait, l'accès au réseau s'effectue par "ip aliasing", l'adresse IP du vserver est ajoutée sur la carte réseau voulue et son trafic est isolé au niveau de l'api vserver pour sécuriser le tout.

Il est possible depuis les dernières versions du projet de faire de l'accounting quant à l'utilisation des ressources de l'hôte : CPU, mémoire, scheduling, etc.

Il est aussi possible de limiter l'utilisation des ressources de l'hôte pour chaque Vserver, en particulier mémoire résidente (RSS) ou mémoire virtuelle (VSZ).

Xen

Xen est une solution de virtualisation open source développée initialement par le département informatique de l'Université de Cambridge. Son développement est aujourd'hui activement sponsorisé par la société XenSource, fondée par l'un des créateurs de Xen.

XenSource distribue des versions Entreprise de Xen, dotées d'une interface d'administration avancée et d'un accès au support technique. Quant aux fonctionnalités, elles sont les mêmes que dans la version distribuée librement.

De grandes sociétés comme IBM contribuent au développement de Xen, qui est d'autre part soutenu par des fonds d'investissements.

En août 2007, XenSource a été racheté par Citrix, ce qui a fait naître quelques inquiétudes quant à son positionnement open source. Il est un peu tôt pour se prononcer sur ce point.

Fonctionnement de Xen

Chaque système s'exécutant sous l'hypervision de Xen s'appelle un *domaine*, et dispose d'une interface particulière d'accès aux ressources.

Il est possible d'attribuer à chaque domaine une limite de mémoire, une limite d'utilisation du CPU, ainsi qu'une priorité d'utilisation du temps de CPU disponible ce qui permet de donner une priorité plus importante par exemple au serveur virtuel considéré 'critique'.

L'un des domaines possède un rôle particulier au sein de Xen, il s'agit du *domaine zéro*. Ce domaine est le premier OS lancé par l'hyperviseur au démarrage du serveur physique. Il donne accès aux ressources par l'intermédiaire de ses pilotes de périphériques.. Depuis le domaine zéro, il est également possible d'avoir accès au bus de contrôle de Xen, permettant de lancer, d'arrêter et même de prendre le contrôle des domaines virtuels exécutés. Il est donc important d'accorder une attention particulière à la sécurité de ce domaine, par exemple en l'isolant du réseau.

Le système d'exploitation du domaine 0, et lui seul, doit disposer d'un noyau patché (modifié) d'une manière particulière. Pour l'heure, seuls GNU/Linux et NetBSD proposent les patchs permettant de fonctionner en domaine zéro.

De nombreuses distributions Linux fournissent l'hyperviseur Xen, un noyau patché pour fonctionner en domaine zéro, et des outils d'administration. C'est notamment le cas de *Red Hat Entreprise Linux*, ou encore *Debian GNU/Linux*. De la même façon que Linux VServers, Xen s'installe très facilement sur un serveur physique, mais nécessite quelques étapes supplémentaires, notamment en termes de partitionnement des disques.

La particularité de Xen en tant que solution de virtualisation est de fournir deux modes d'utilisation. Un mode paravirtualisation, et un mode virtualisation complète.

Paravirtualisation sous Xen

Pour l'heure seules les distributions GNU/Linux et certaines versions de BSD peuvent être exécutées en tant que domaine zéro. De même, seuls quelques systèmes sont utilisables en tant que domaine non-privilégiés de façon stable ; en particulier GNU/Linux, Plan9, NetBSD,

d'autres OS comme NetWare et Solaris n'étant supportés que de façon expérimentale.

GNU/Linux est naturellement la cible privilégiée de la paravirtualisation sous Xen. Les serveurs paravirtualisés avec Xen ne souffrent quasiment d'aucune perte de performance due à la présence de l'hyperviseur, et sa gestion du processeur est simple et garantit un partage équitable du temps de calcul. Il est possible de choisir le noyau des domaines virtuels indépendamment du domaine zéro, ce qui autorise une grande hétérogénéité dans le choix des distributions.

En mode paravirtualisé, Xen fournit aux domaines non-privilégiés des disques et des interfaces réseau virtuelles, lesquelles peuvent être configurées à chaud. Il est possible d'ajouter à chaud des disques ou interfaces réseau virtuels, au moyen des outils d'administration fournis dans le domaine zéro. Il est également possible de modifier à chaud la quantité de mémoire allouée aux domaines virtuels, les limitations de CPU et de redimensionner l'espace disque disponible.

Machine virtuelle sous Xen

Le mode HVM de Xen est apparu avec la version 3. Il utilise un noyau spécial en mode paravirtualisé pour simuler une machine virtuelle, ce qui permet de faire fonctionner des OS fermés comme Microsoft Windows pour lesquels il n'existe pas de patch Xen publics pour le mode paravirtualisé.

Ce mode n'est toutefois possible que si la machine hôte dispose d'un processeur doté des jeux d'instructions de virtualisation matérielle (Intel VT ou AMD Pacifica). Intel et AMD ont d'ailleurs contribué au code de Xen pour le support de leurs processeurs.

Au prix d'une couche de virtualisation supplémentaire, il est ainsi possible de retrouver tous les avantages de la machine virtuelle. Il est intéressant de noter que la couche d'interface entre l'OS virtualisé et l'hyperviseur provient en grande partie du projet open source QEMU, créé par le français Fabrice Bellard, l'un des pionniers en matière de machines virtuelles. En revanche, contrairement à QEMU, Xen ne peut héberger que des machines virtuelles compilées pour fonctionner sur la même architecture que celle du processeur de la machine hôte.

La machine virtuelle Xen possède la même souplesse que le mode paravirtualisé car elle dispose de la même interface de contrôle.

Avantages de Xen

Un des grands avantages de Xen est sa souplesse. Une grande liberté est permise, en particulier, dans le choix d'une solution de stockage pour les disques virtuels : fichiers plats, LVM, SAN, etc...

De même le réseau peut être personnalisé de façon à répondre à quasiment tous les besoins spécifiques. Il est notamment possible d'assigner les cartes réseau du serveur physique à une ou plusieurs interfaces virtuelles et ce pour chaque domaine (y compris le domaine zéro), ce qui permet d'isoler certains domaines d'un réseau, ou au contraire de donner à un domaine seulement le contrôle sur une interface. La couche de virtualisation réseau de Xen permet ainsi de mettre en place tous types d'application et de configurations réseau : NAT, VLAN, bridges, routage, etc.

De plus, Xen permet de "migrer" un domaine virtuel d'un serveur à l'autre quasiment sans interruptions en utilisant un mécanisme de sauvegarde de la RAM proche de l'hibernation *suspend-to-disk* ce qui confère une grande évolutivité à la solution.

Limitations de Xen

La seule critique qui peut être faite à Xen est le manque d'ergonomie de la distribution libre. Celle-ci ne dispose pas de l'interface graphique présente dans les versions payantes De plus, la documentation disponible librement n'est pas toujours actualisée, et de nombreuses possibilités intéressantes ne sont pas forcément documentées. Ce qui fait de Xen une solution puissante, mais parfois délicate à appréhender et requiert une certaine expertise.

D'autre part, les fonctionnalités HVM sont relativement récentes dans le projet, ce qui fait que les systèmes d'exploitation les moins répandus (FreeBSD ou bien des distributions Linux exotiques) présentent encore quelques dysfonctionnements.

En revanche, le support de Windows (XP, 2000, 2003) est parfaitement opérationnel.

DOMAINES D'APPLICATION

Hébergement VDS

Les offres d'hébergement étaient traditionnellement distinguées en deux catégories : hébergement dédié et hébergement mutualisé.

Dans un hébergement dédié, le fournisseur met à disposition de son client un ou plusieurs serveurs, configurés selon ses besoins. Selon les cas, le contrat peut prévoir une plus ou moins grande autonomie du client par rapport à la configuration et l'exploitation de son serveur, mais du moins techniquement rien ne s'oppose à ce que le contrôle soit total.

Avec un hébergement mutualisé, le fournisseur utilise un même serveur pour plusieurs de ses clients. Il utilise différentes solutions de cloisonnement pour maintenir une certaine étanchéité entre ces environnements.

Le partage de la ressource serveur permet bien sûr un coût très inférieur, particulièrement attractif pour les sites à faible trafic. Mais l'hébergement mutualisé simple a plusieurs handicaps :

- L'allocation des ressources du serveur n'est pratiquement pas contrôlée, de sorte que la qualité de service de chaque site peut être pénalisée par un pic de trafic, ou par la boucle d'un programme sur un autre site.
- La configuration logicielle est unique, et dictée par l'hébergeur. Elle fait le choix, en général, d'un même serveur Http, mais aussi très souvent d'un même outil de gestion de contenus et de base de données. La simple installation de telle ou telle librairie spécifique nécessaire à l'un des clients n'est en général pas possible. Et a fortiori, des configurations globales sur mesure sont interdites.
- En termes d'exploitation, chaque client est extrêmement confiné, de peur qu'il ne perturbe la configuration. Il dispose le plus souvent d'un simple accès en transfert de fichier sur son répertoire privé, et dans tous les cas n'a jamais l'accès *root* (administrateur) sur le serveur.

Entre ces deux modes d'hébergement, la virtualisation a permis un mode combinant les bénéfices de l'un et de l'autre : le partage de ressources d'une part, l'autonomie et le contrôle d'autre part.

C'est le mode que l'on appelle « VDS » pour *Virtual Dedicated Server*, un serveur dédié virtuel.

Il consiste tout simplement à mettre en œuvre des serveurs virtuels selon les différentes technologies décrites plus haut, et d'allouer un serveur virtuel à chaque client.

Le mode VDS permet donc :

- De partager un même serveur physique en N serveurs virtuels, alloués à différents clients. Le nombre de serveurs virtuels par serveur physique dépend bien sûr des besoins respectifs de chacun, mais n'a pas de limite théorique.
- De définir – du moins selon la technologie de virtualisation retenue – la part de ressources allouée à chaque client.
- De donner à chaque client un contrôle total sur son serveur virtuel : il peut y installer les composants de son choix, disposer d'un accès *root*, gérer ses utilisateurs et droits, rebooter le serveur, ré-installer l'OS.

Selon la technologie de virtualisation retenue, les limites de cette maîtrise pourront varier :

- Avec une technologie d'isolation de type Linux Vserver, il aura la liberté de choisir quelle distribution il souhaite installer, et quelles applications il utilisera, mais devra se satisfaire du noyau en place.
- Avec une technologie de virtualisation complète, il aura le choix du système d'exploitation installé sur sa machine, pourra rebooter à volonté sous différents OS, et mettre en place son propre filtrage réseau.

Plate forme de validation et de développement

La compatibilité des applications avec la grande variété des configurations informatiques disponibles est un enjeu majeur, tout particulièrement pour les progiciels.

Garantir cette compatibilité implique de tester les produits sur un large ensemble de plateformes, d'architectures, de systèmes d'exploitation

différents, associés le cas échéant à une variété de bases de données ou d'autres composants système.

Les grands éditeurs, tels que Dassault Système par exemple, utilisent pour cela des fermes de validation comportant plusieurs centaines de serveurs.

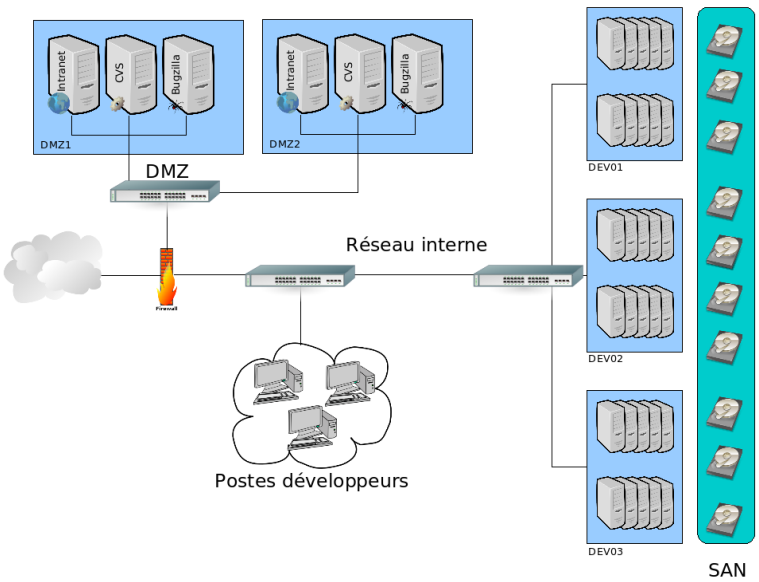
Les solutions de virtualisation permettent d'alléger quelque peu ces infrastructures de validation, leur coût matériel, mais aussi leur exploitation.

Les applications peuvent être compilées et testées automatiquement sur un grand nombre d'environnements virtuels (successivement ou même simultanément). Dans ce domaine, on privilégie naturellement les solutions de virtualisation complète, supportant une variété d'OS.

Les solutions de machines virtuelles avec émulateur, bien que moins efficaces en termes de performances, permettent même de simuler un processeur différent de celui de l'hôte.

Un autre usage de la virtualisation dans le cadre d'une plateforme de développement est l'instanciation et l'administration de parcs de serveurs de développement, d'intégration, de recette, etc...

Au sein de larges équipes de développement, comme c'est typiquement le cas chez un prestataire informatique, chaque projet peut posséder ses propres serveurs virtuels, sans aucun impact sur les autres projets, et mettre en place des environnements de test et de pré-production de façon souple et rapide.



Haute disponibilité

En matière de haute disponibilité ou de haute capacité d'accueil, les mécanismes centraux sont devenus classiques et bien maîtrisés : répartition de charge (*load balancing*) et reprise automatique sur incident (*failover*). Sur ces différentes techniques, la virtualisation apporte son lot d'avantages.

Répartition de charge

La répartition de charge est à la base un moyen d'augmenter la capacité maximale d'une application, en l'hébergeant sur plusieurs serveurs qui se partagent les visiteurs.

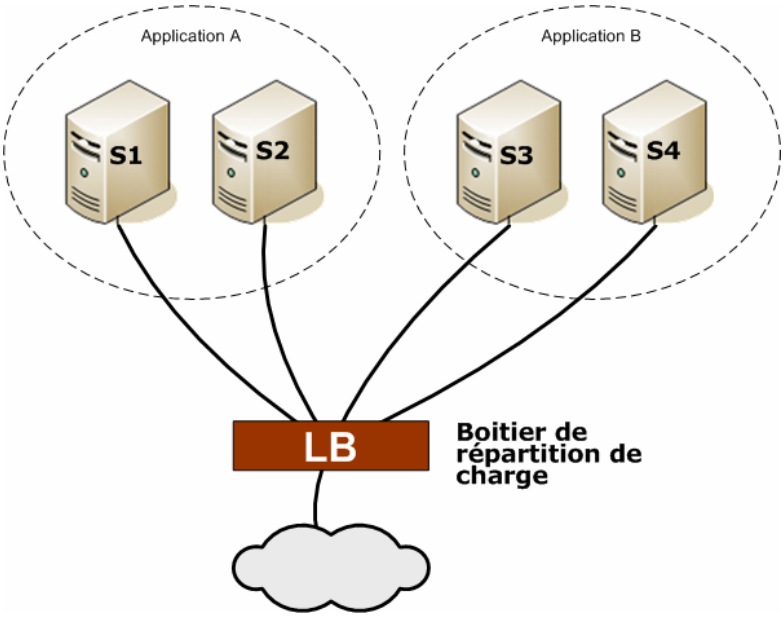
La répartition de charge est le plus souvent mise en œuvre au moyen d'un boitier spécialisé, qui dirige les requêtes des visiteurs sur les différents serveurs, en conservant ou non un même visiteur sur un même serveur. Les boitiers de répartition de charge savent en général détecter la panne d'un serveur, et ne plus lui affecter de trafic. Ainsi, *load balancing* et *failover* vont souvent de pair.

Pour des plateformes à très forte audience, et à vocation ciblée, le partage des serveurs physiques n'est pas d'une grande utilité. C'est le cas typiquement d'un grand site web recevant plusieurs centaines de milliers de visiteurs par jour, dont le trafic est réparti sur quelques serveurs. Pour autant, la virtualisation pourra avoir d'autres usages.

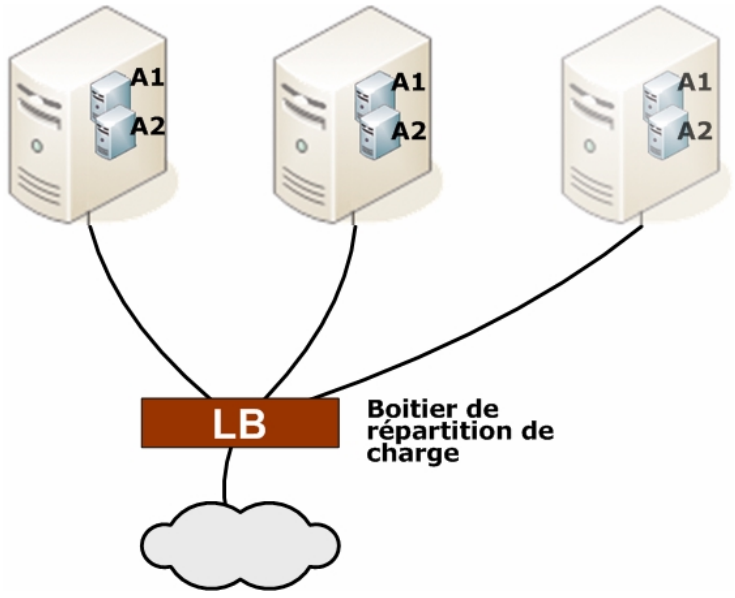
Mais si l'on est en présence de plusieurs applications, ayant chacune besoin de répartition de charge sur plusieurs serveurs, alors la virtualisation peut apporter une meilleure mutualisation de moyens.

Supposons que l'on exploite deux applications critiques A1 et A2. Chacune dispose de deux serveurs physiques entre lesquels le trafic est réparti. Supposons, ce qui arrive souvent, que ces serveurs ne soient pas utilisés à pleine capacité. Une bonne alternative d'architecture, consiste alors à réunir les deux applications sur deux, voire trois serveurs, chacun partagé en deux machines virtuelles, l'une pour A1, l'autre pour A2.

Ainsi au lieu de 4 serveurs, on n'en a plus que 3, voire 2. Et au lieu d'une répartition sur 2 serveurs, on a une répartition sur 3.



Architecture traditionnelle, plateformes applicatives séparées



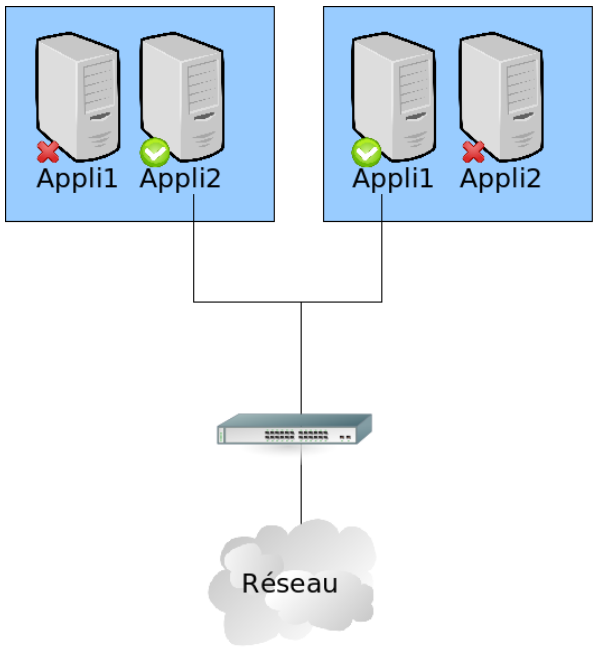
Architecture virtualisée, plateformes applicatives mutualisées

Reprise automatique

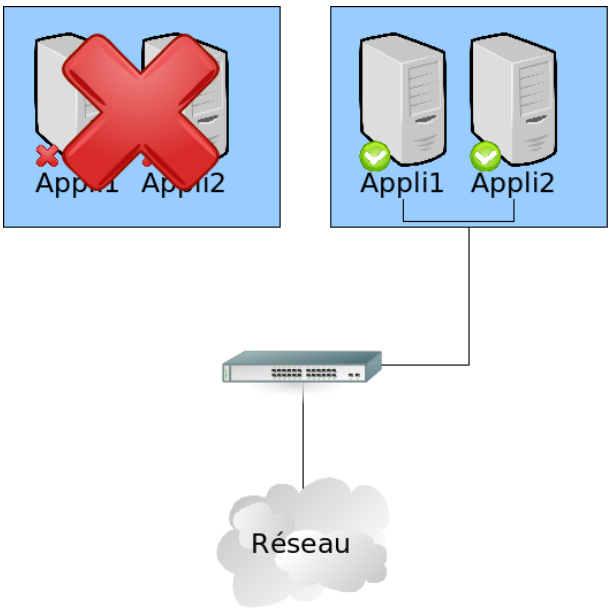
Un autre usage de la virtualisation dans une optique de haute disponibilité de service peut consister à avoir sur plusieurs serveurs physiques les mêmes environnements virtuels (synchronisés régulièrement).

Les différents serveurs physiques se partagent les différents serveurs virtuels, et si un des serveurs physiques tombe en panne, les machines dont il avait la responsabilité sont relancées sur les autres serveurs. Cela permet d'assurer un temps d'indisponibilité minimum, et une continuité de service malgré des performances amoindries. On peut ainsi travailler plus sereinement à la remise en route du serveur en panne.

Bien sûr il est possible de combiner répartition de charge et reprise automatique sur plusieurs hôtes physiques pour une robustesse encore accrue.



Après une panne d'un des serveurs :



Virtual appliance

Le concept d'*appliance*

Dans le domaine du réseau, le concept d' *appliance* est démocratisé depuis plusieurs années. Il s'agit de boitiers prêts à l'emploi : firewall, routeurs, solutions de sécurité tout-en-un, qui se branche facilement sur le réseau et nécessite très peu de configuration de la part des administrateurs.

Après les *appliances* physiques, les *software appliances* sont des configurations logicielles complètes packagées, incluant le système d'exploitation, la configuration système complète, l'application principale et tous les composants logiciels dont elle a besoin, le tout en un pacquet aisément installable. La software appliance permet à l'administrateur système de ne plus se préoccuper de la compatibilité de tels et tels composants logiciels : la configuration est unique, validée et packagée en amont. Les *software appliance* permettent d'alléger considérablement l'administration des configurations, de même que les tests de qualification d'un produit. Elles n'ont qu'un inconvénient : en l'absence de virtualisation, elle requièrent un serveur par application.

D'où le concept de *virtual appliance*, une *software appliance* qui s'installe dans une solution de virtualisation existante dans le but de remplir une certaine fonction.

Ces "*virtual appliances*" se présentent sous la forme d"images de machines virtuelles, déjà parfaitement configurées et packagées avec l'application voulue. Leur déploiement est aisé, bien loin de l'installation manuelle complète d'un système d'exploitation, d'une l'application et des utilitaires associés, en terme de temps donc de coût.

De plus ces "appliances" sont facilement sauvegardables et transportables car en général elles occupent un espace disque réduit (très peu de logiciels superflus, seulement l'OS de base est installé ainsi que l'application voulue).

De très nombreuses possibilités

On peut ainsi trouver ou construire des "appliances" pour tout type de besoins, il suffit ensuite de configurer quelques variables et l'architecture est opérationnelle et déployable à volonté.

On trouve sur le web des idées d'appliance pour tout les usages, et pour tout les produits phares de l'industrie opensource : LAMP, Asterisk, Nagios/Cacti, Joomla, etc..

Architecture LAMP

Pour faire du développement ou simplement des tests il est souvent très utile d'avoir des environnements LAMP génériques, par exemple pour les équipes de Smile, nous avons souvent à déployer ce genre d'environnement pour nos développeurs ou nos clients. Nous gagnons énormément de temps avec ce genre d'*appliance*, qui sont prêtes à l'emploi pour divers types de besoins (eZpublish, Typo3 ...).

Firewall, VPN

Les capacités réseau de certaines solutions de virtualisation permettent même de mettre en place des serveurs virtuels ayant la main sur les interfaces réseau, ce qui permet l'utilisation d'un composant virtuel pour servir de firewall, de système de détection d'intrusions, d'endpoint VPN, totalement isolé du matériel, et donc moins sensible en cas d'attaque.

CONCLUSION

Synthèse

	Isolateurs		Hyper viseurs	Machines virtuelles	
	Linux VServer	OpenVZ	XEN	XEN HVM	QEMU Bochs
Contrôle des ressources	***	**	**	**	*
Accounting	**	**	**	**	*
Performances	***	***	**	**	**
Scheduling	***	**	**	**	*
Configuration	***	***	**	**	*
Autres fonctionnalités	*	*	***	**	*
Réseau	*	**	***	***	**
Compatibilité OS			*	**	***

Quelle solution choisir ?

Quelques règles simples pour choisir une solution de virtualisation.

➔ Si vous gérez des environnements purement Linux, avec des besoins de hautes performances, sans besoin de déplacements à chaud, choisissez Linux VServer.

➔ Si vous gérez des environnements purement Linux, mais avec des besoins plus précis en termes de version noyaux ou de maîtrise du routage/filtrage dans la configuration réseau, choisissez XEN.

➔ En environnement purement Windows avec des besoins de haute performance, choisissez XEN HVM.

➔ En environnement mixte (Linux/Windows) : XEN, avec cohabitation des différents modes de fonctionnement

➔ En environnement mixte (Linux/Windows) mais avec peu de compétences spécifiques et besoin de la meilleure simplicité d'utilisation, retenez plutôt le produit commercial VMWare ESX.

➔ Et enfin pour des besoins d'expérimentation, essayez donc QEMU.

L'avenir

Après les évolutions apportées par VMWare et Xen, la prochaine étape pourrait bien venir de KVM, nouvellement intégré au noyau Linux et sujet d'une attention croissante de la part de la communauté libre. Il promet pour très bientôt de mettre la virtualisation à la portée des particuliers, et, moyennant le support des grands groupes dont bénéficient actuellement Xen et VMWare, de déferler sur le monde de l'entreprise.

Les technologies de virtualisation ont très rapidement tenu leurs promesses en termes de réduction de coût d'acquisition et de possession des parcs informatiques. En quelques années, elles se sont répandues, et même généralisées. De plus en plus d'administrateurs système préfèrent mettre en place un environnement virtualisé même s'il n'y a dans un premier temps qu'un seul serveur virtuel. D'une part cela permettra ultérieurement de mettre en œuvre un partage, et d'autre part cela permet de bénéficier des services qui ne sont pas liés au partage, par exemple la sauvegarde et reprise de l'environnement.

Les solutions open source apportent exactement le même niveau de service que les solutions commerciales en termes de robustesse, de performances et de pérennité. Il leur reste seulement à combler un petit retard en termes d'ergonomie des interfaces. Mais pour des administrateurs système chevronnés, elles sont le plus souvent privilégiées.