SYSTEME ET INFRASTRUCTURE

Virtualisation et Cloud open source

Principes, mise en œuvre et outils open source

Maxime **BESSON**
Expert technique

Smile
OPEN SOURCE SOLUTIONS

www.smile.fr • +33 (0)1 41 40 11 00 • contact@smile.fr
www.smile-oss.com • blog.smile.fr • twitter: @GroupeSmile

PREAMBULE

SMILE

Smile est une **société d'ingénieurs experts** dans la mise en œuvre de **solutions open source** et l'intégration de systèmes appuyés sur l'open source. Smile est membre de l'**APRIL**, l'association pour la promotion et la défense du logiciel libre, du **PLOSS** – le réseau des entreprises du Logiciel Libre en Ile-de-France et du **CNLL – le conseil national du logiciel libre**.

Smile compte près de 600 collaborateurs dans le monde, dont plus de 500 en France (décembre 2012), ce qui en fait *le premier intégrateur français et européen de solutions open source.*

Depuis 2000, environ, **Smile mène une action active de veille technologique** qui lui permet de découvrir les produits les plus prometteurs de l'open source, de les qualifier et de les évaluer, de manière à proposer à ses clients les produits les plus aboutis, les plus robustes et les plus pérennes. Cette démarche a donné lieu à **toute une gamme de *livres blancs*** couvrant différents domaines d'application. La gestion de contenus (2004), les portails (2005), la business intelligence (2006), la virtualisation (2007), la gestion électronique de documents (2008), les PGIs/ERPs (2008), les VPN open source (2009), les Firewall et Contrôle de flux (2009), les Middleware orientés messages (2009), l'ecommerce et les Réseaux Sociaux d'Entreprise (2010) et plus récemment, le Guide de l'open source et NoSQL (2011). Chacun de **ces ouvrages présente une sélection des meilleures solutions open source** dans le domaine considéré, leurs qualités respectives, ainsi que des retours d'expérience opérationnels.

Au fur et à mesure que des solutions open source solides gagnent de nouveaux domaines, Smile sera présent pour proposer à ses clients d'en bénéficier sans risque. Smile apparaît dans le paysage informatique français comme **le prestataire intégrateur de choix** pour **accompagner** les plus grandes entreprises dans l'adoption des meilleures solutions open source.

Ces dernières années, Smile a également étendu la gamme des services proposés. Depuis 2005, un département consulting accompagne nos clients, tant dans les phases d'avant-projet, en recherche de solutions, qu'en accompagnement de projet. Depuis 2000, Smile dispose d'un studio graphique, devenu en 2007 Smile Digital – agence interactive, proposant outre la création graphique, une expertise e-marketing, éditoriale, et interfaces riches. Smile dispose aussi d'une agence spécialisée dans la TMA (support et l'exploitation des applications) et d'un centre de formation complet, Smile Training. **Enfin, Smile est implanté à Paris, Lille, Lyon, Grenoble, Nantes, Bordeaux, Marseille et Montpellier. Et présent également en Espagne, en Suisse, au Benelux, en Ukraine et au Maroc**.

QUELQUES RÉFÉRENCES DE SMILE

SMILE est fier d'avoir contribué, au fil des années, aux plus grandes réalisations Web françaises et européennes. Vous trouverez ci-dessous quelques clients nous ayant adressé leur confiance.

Infrastructure et Hébergement

Agence Nationale pour les Chèques Vacances, Pierre Audoin Consultants, Rexel, Motor Presse, OSEO, Sport24, Eco-Emballage, Institut Mutualiste Montsouris, ETS, Ionis, Osmoz, SIDEL, Atel Hotels, Cadremploi, SETRAG, Institut Français du Pétrole, Mutualité Française, Bouygues Telecom, Carrefour, HEC, Jardiland, Orange, TNS Sofres, Manpower, Ministère de l'économie, Eram, Kantar Worldpanel, Fiducial……

Sites Internet

EMI Music, Salon de l'Agriculture, Mazars, Areva, Société Générale, Gîtes de France, Patrice Pichet, Groupama, Eco-Emballage, CFnews, CEA, Prisma Pub, Véolia, NRJ, JCDecaux, 01 Informatique, Spie, PSA, Boiron, Larousse, Dassault Systèmes, Action Contre la Faim, BNP Paribas, Air Pays de Loire, Forum des Images, IFP, BHV, ZeMedical, Gallimard, Cheval Mag, Afssaps, Beneteau, Carrefour, AG2R La Mondiale, Groupe Bayard, Association de la Prévention Routière, Secours Catholique, Canson, Veolia, Bouygues Telecom, CNIL…

Portails, Intranets et Systèmes d'Information

HEC, Bouygues Telecom, Prisma, Veolia, Arjowiggins, INA, Primagaz, Croix Rouge, Eurosport, Invivo, Faceo, Château de Versailles, Eurosport, Ipsos, VSC Technologies, Sanef, Explorimmo, Bureau Veritas, Région Centre, Dassault Systèmes, Fondation d'Auteuil, INRA, Gaz Electricité de Grenoble, Ville de Niort, Ministère de la Culture, PagesJaunes Annonces…

E-Commerce

Krys, La Halle, Gibert Joseph, De Dietrich, Adenclassifieds, Macif, Furet du Nord, Gîtes de France, Camif Collectivité, GPdis, Projectif, ETS, Bain & Spa, Yves Rocher, Bouygues Immobilier, Nestlé, Stanhome, AVF Périmédical, CCI, Pompiers de France, Commissariat à l'Energie Atomique, Snowleader, Darjeeling…

ERP et Décisionnel

Veolia Transport, Solucom, Casden Banques Populaires, La Poste, Christian Louboutin, PubAudit, Effia Trasnport, France 24, Inra, Publicis, Nomadvantage, Nouvelles Frontières, Anevia, Jus de Fruits de Mooréa, Espace Loggia, Bureau Veritas, Skyrock, Lafarge, Cadremploi, Groupe Vinci, IEDOM (Banque de France), Carrefour, Corsair, Le Bon Coin, Jardiland, Trésorerie Générale du Maroc, Ville de Genève, ESCP, Sofia, Faiveley Transport, INRA, Deloitte, Yves Rocher, ETS, DGAC, Generalitat de Catalunya, Gilbert Joseph, Perouse Médical…

Gestion documentaire

Primagaz, UCFF, Apave, Géoservices, Renault F1 Team, INRIA, CIDJ, SNCD, Ecureuil Gestion, CS informatique, Serimax, Veolia Propreté, NetasQ, Corep, Packetis, Alstom Power Services, Mazars…

Consultez nos références, en ligne, à l'adresse : http://www.smile.fr/clients.

CE LIVRE BLANC

Si 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.

Et ce n'est pas seulement vrai pour le monde des serveurs puisque **la virtualisation s'attaque de plus en plus à la problématique du poste de travail** où un marché immense s'offre à elle. De plus, **la virtualisation est à la base du phénomène Cloud** – présenté dans ce livre blanc – notamment depuis l'apparition de solutions d'Iaas open source telles qu'OpenStack (également décrit comme « le framework open source plateforme du cloud ») permettant **l'industrialisation de la virtualisation**.

Pratiquant les différentes 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 (dont le Cloud Computing).
- 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.
- Une présentation d'une problématique connexe à la virtualisation : le stockage, et les solutions à mettre en œuvre en environnement virtualisé telles que les interfaces d'administration.

Enfin en conclusion, nous présentons un **tableau comparatif** dressant la synthèse des fonctionnalités présentes dans les différents outils, et une note présentant **l'avenir très prometteur** de la virtualisation open source.

N'hésitez pas à nous transmettre vos avis et évaluations sur ce livre blanc.
Une seule adresse : contact@smile.fr

open source

SOMMAIRE

Les Principes de la Virtualisation

Partage d'un serveur

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

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.

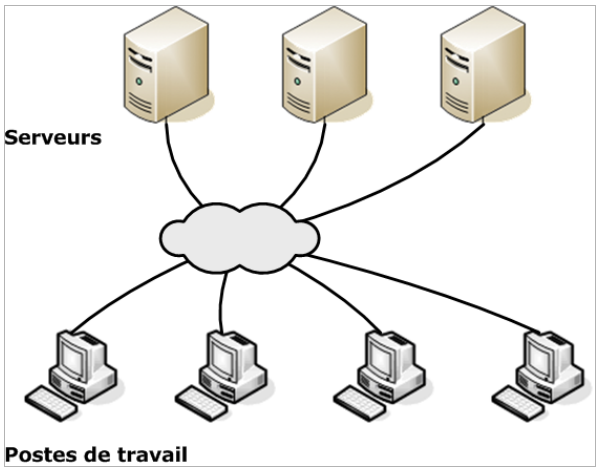
Pour être utile de manière opérationnelle, la virtualisation doit respecter deux principes fondamentaux :

Le *cloisonnement* : chaque système d'exploitation a un fonctionnement indépendant, et ne peut interférer avec les autres en aucune manière.

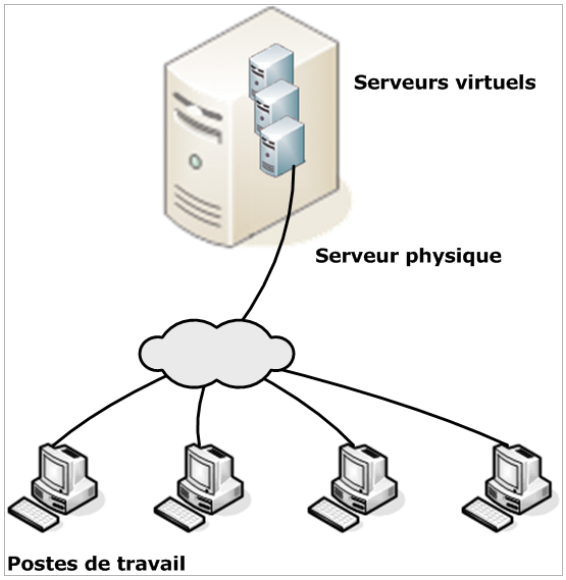
La *transparence* : le fait de fonctionner en mode virtualisé ne change rien au fonctionnement du système d'exploitation et a fortiori des applications.

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. On estime que dans un datacenter privé ordinaire, le taux d'utilisation moyen est de l'ordre de 10%, et qu'une utilisation généralisée de la virtualisation permet d'atteindre 35%. C'est encore loin de 100, mais c'est quand même trois fois moins de serveurs.

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.

Éviter 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. D'une certaine manière, le serveur devient une ressource ordinaire, une

« commodité », et au lieu de *répartir des applications* sur des serveurs, on *fournit du serveur* aux applications.

Avec certaines solutions de virtualisation, le déplacement s'effectue de manière totalement transparente pour le système invité. Le délai de ce déplacement nécessite le transfert de l'espace disque et de la mémoire, et nous verrons que certaines technologies de virtualisation et certaines configuration de stockage permettent des transferts à chaud sans arrêt des applications.

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 utilisateurs, 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 parmi 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 destinées 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, KVM, et OpenVZ, que nous décrirons plus en détail dans ce document.

Et pour finir les logiciels orientés poste de travail comme VMware Player ou VirtualBox 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 avec un rendement proche de 100%. La course à la performance n'est cependant pas terminée, et de nouvelles technologies comme la virtualisation des IO sont encore en cours de généralisation.

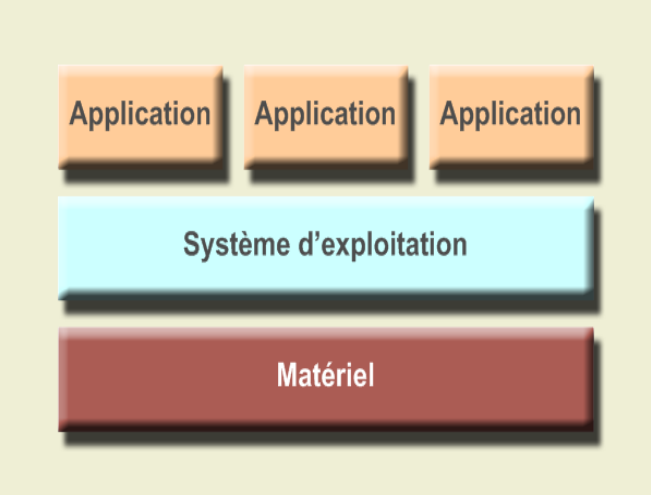
Mais dans l'ensemble, la technologie à la base de la virtualisation est mature, et l'essentiel des efforts se concentre désormais sur les outils permettant de tirer profit de la virtualisation : que ce soit les outils de stockage, de sauvegarde, ou d'administration. Les offres estampillées « Cloud Computing » sont un bon exemple des possibilités offertes par une virtualisation dont le cycle de vie est entièrement automatisé.

UN PEU DE VOCABULAIRE

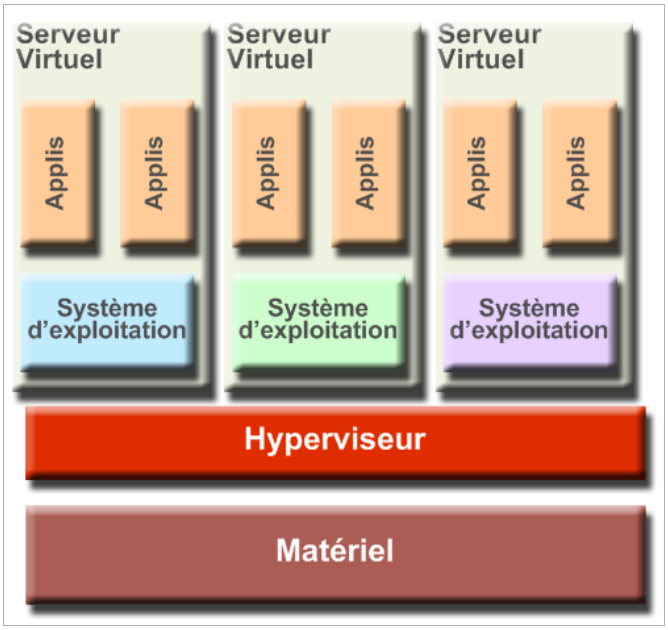
Hyperviseur

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.

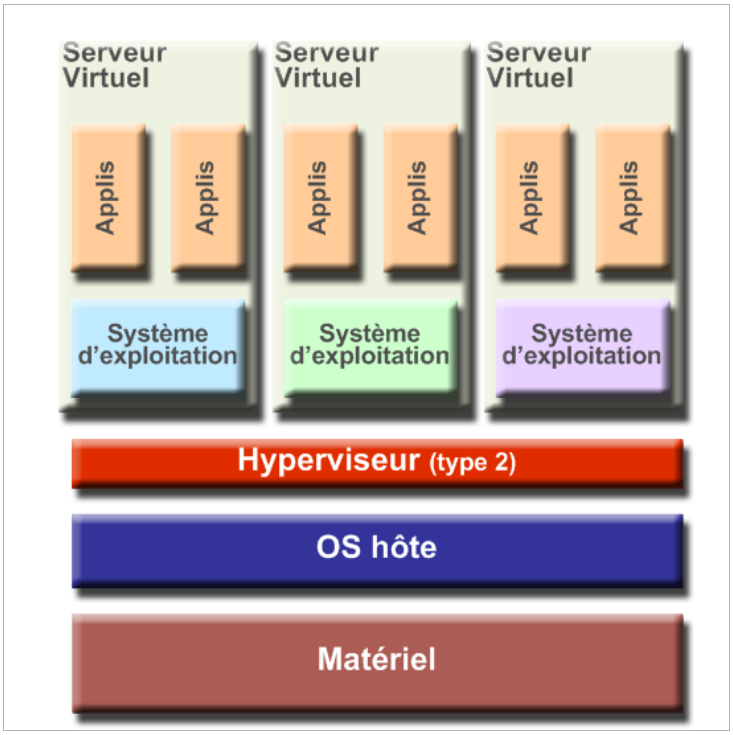
Ainsi, par rapport au schéma de base d'un serveur distinguant le matériel, le système d'exploitation, et ses applications :



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



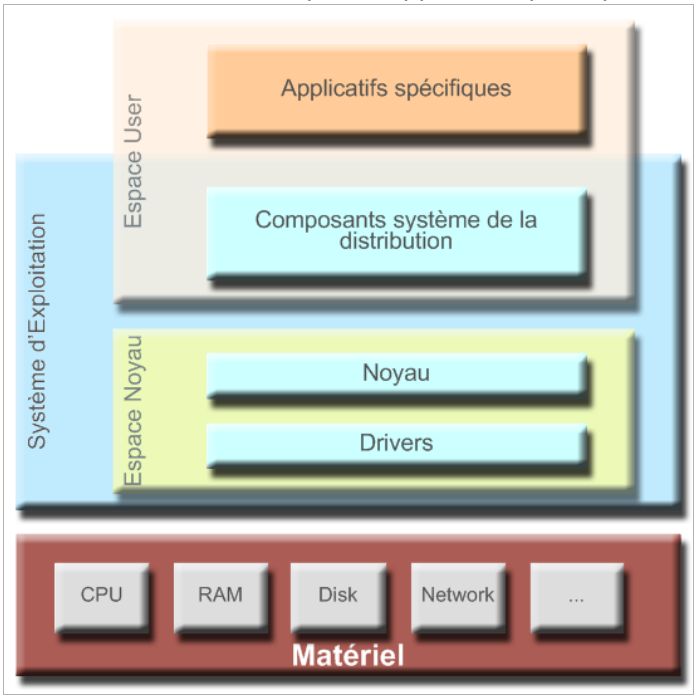
L'hyperviseur peut soit gérer lui-même toutes les ressources matérielles du serveur, soit s'appuyer pour cela sur un système d'exploitation existant. Dans ce dernier cas, on parle d'hyperviseur de type 2, comme figuré ci-après.



Espace noyau, espace utilisateur

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

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



OS hôte, OS invité

Dans le cas d'un hyperviseur de type 2, on appelle « OS hôte » ou *Host OS*, l'OS sous-jacent, sur lequel s'appuie l'hyperviseur.

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

Émulation

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 matériels, le différentiel de puissance des processeurs sur 10 ans comblait largement la perte résultant de l'émulation.

Le projet QEMU est la principale 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 presque 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 d'y assurer un haut 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 séparer le réseau d'administration des serveurs physiques, et le réseau des VM par lequel arrive le trafic utilisateur.

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.

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 toutes les ressources de la machine physique sur laquelle ils s'exécutent.

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

- Le 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.
- La 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 réservée à la création.
- Le stockage : les différents produits de virtualisation peuvent s'appuyer sur différents 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. De plus, une priorisation des accès est généralement possible pour favoriser certains environnements (par exemple les bases de données).

- Le 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.

LICENCES ET SUPPORT

La virtualisation permet d'exécuter des OS supplémentaires, et aussi d'en faire des multiples copies, backups, etc. Cela pose le problème des licences, qui ne sont le plus souvent pas prévues pour une utilisation en machines virtuelles. Il faut donc faire attention à ce problème. Microsoft Windows, par exemple, permet de licencier un certain nombre d'installations, et autorise d'avoir autant de machines virtuelles que l'on souhaite, du moment qu'on n'en exécute jamais plus simultanément que le nombre autorisé par la licence.

De même, il est fréquent que les éditeurs de logiciels ne supportent pas telle ou telle configuration matérielle, ou technologie de virtualisation. A l'inverse, certains logiciels sont certifiés compatibles avec Xen ou VMWare ESX. Dans les faits, plus la technologie de virtualisation est intrusive pour le système invité, moins il est probable qu'elle soit supportée par les éditeurs.

ÉTAT DE L'ART

Il existe différentes techniques de virtualisation, citons par niveau d'abstraction croissant :

- L'isolation
- La paravirtualisation
- La virtualisation complète, ou machine virtuelle
- Le partitionnement matériel

L'isolation consiste à mettre en place, *sur un même noyau de système d'exploitation*, une séparation forte entre différents contextes logiciels. Il s'agit de la technique de virtualisation la plus « légère » qui existe.

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 ou de modifications du noyau. Il s'agit d'un compromis entre un niveau d'abstraction élevé et un niveau de performance satisfaisant.

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.

Le partitionnement matériel, enfin, est la technique historique utilisée sur les gros systèmes. Elle consiste à séparer les ressources matérielles au niveau de la carte mère de la machine. Cette technique est surtout répandue dans les serveurs hauts de gamme, par exemple les *Logical Domains* de chez *Sun*. Elle est assez rare dans le monde x86. Les *blades* en sont un exemple, mais ils n'offrent pas des fonctionnalités aussi avancées que ce que l'on retrouve sur d'autres architectures matérielles comme *SPARC*.

Nous présenterons dans la suite les deux techniques majeures du monde x86 : l'isolation et la virtualisation complète.

L'édition précédente de ce livre blanc traitait de paravirtualisation, cette séparation n'est plus d'actualité car paravirtualisation et virtualisation complète sont désormais utilisées conjointement au sein d'un même produit.

ISOLATION

Présentation

L'isolation (aussi appelée 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 peuvent 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.

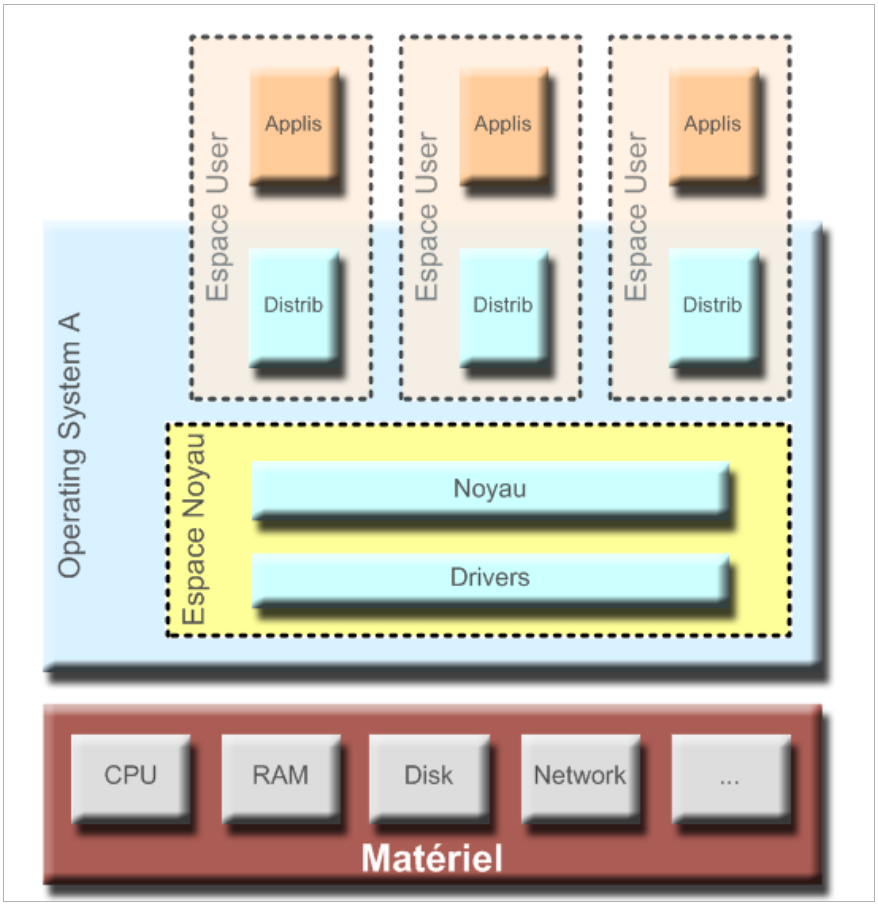
L'isolation est utilisée sous Unix depuis longtemps pour protéger les systèmes. Via des mécanismes comme *chroot* ou *jail* il est possible d'exécuter des applications dans un environnement qui n'est pas celui du système hôte, mais un « mini système » ne contenant que ce dont l'application a besoin,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

Les deux principales solutions pour l'isolation Linux sont OpenVZ et LXC. Linux-Vserver, une solution qui a eu son heure de gloire, est parfois rencontrée, mais son adoption diminue rapidement.

OpenVZ est aujourd'hui la solution la plus mature et disposant du plus grand nombre de fonctionnalités. C'est pourquoi c'est la solution que nous avons choisi de présenter en détails. LXC est pour sa part une solution relativement jeune et encore instable, mais très prometteuse, car plus profondément intégrée au noyau Linux.

VIRTUALISATION COMPLÈTE

Présentation

La virtualisation complète, comme son nom l'indique, consiste à simuler un ordinateur complet, de façon à exécuter le système d'exploitation de façon naturelle, sans que celui-ci ne se rende compte qu'il est virtualisé.

On parle aussi de 'machines virtuelles', en désignant ces systèmes simulés.

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 ». Ce matériel est soit émulé pour ressembler à un matériel réel (généralement répandu, comme les contrôleurs disque Intel PIIX ou les cartes réseau Broadcom), soit paravirtualisé, c'est à dire qu'il nécessite un pilote particulier dans le système invité pour fonctionner.

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

Les premières solutions de virtualisation complète étaient entièrement 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.

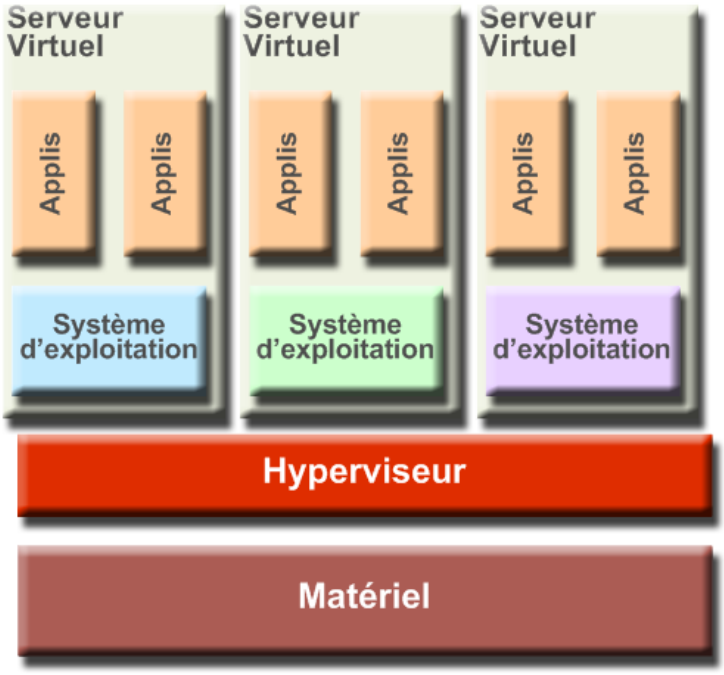
Les produits modernes tirent partie des nouveaux jeux d'instructions spécialisés des dernières générations de processeurs pour assurer des performances de calcul quasi identiques aux performances natives.

Avec l'essor de ce procédé, des techniques issues de la paravirtualisation se sont greffées aux hyperviseurs : en effet la présence de périphériques émulés ralentissait les entrées-sorties de la machine virtuelle, en particulier les accès disques et le trafic réseau. Un pilote de périphérique spécial, capable de dialoguer nativement avec l'hyperviseur sans passer par une interface simulée, permet d'obtenir des performances quasi natives pour les entrées-sorties.

La paravirtualisation des entrées-sorties n'est pas systématiquement supportée, contrairement à l'émulation de périphérique, car elle exige une coopération de la part du système invité. Elle est donc réservée aux systèmes les plus répandus comme Windows et Linux.

Dans les produits commerciaux, elle prend la forme d'un agent s'installant dans la machine virtuelle, et se nomme, selon le produit, « VMWare Tools », « XenServer Tools », « VirtualBox Guest Additions », etc.

En plus de l'accélération des entrées-sorties, ces agents permettent une meilleure interaction entre l'hyperviseur et le système, et permet notamment de commander l'extinction d'une machine virtuelle depuis 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, il souffrait d'une grosse baisse de performance des machines virtuelles. QEMU étant un précurseur, tous les autres produits de virtualisation open source ont emprunté sa couche d'émulation des périphériques.

Après une période de relative obsolescence, QEMU revient aujourd'hui sur le devant de la scène, suite à la fusion du projet avec KVM.

KVM est un module de virtualisation complète fourni par le noyau Linux, qui assure des performances de calcul excellentes, et la possibilité de migrer une VM à chaud entre machines sans nécessiter d'agent, contrairement à toutes les autres solutions. Quant à *virtio* il s'agit d'une couche de paravirtualisation des entrées-sorties, disponible uniquement dans les VM Linux. La combinaison de KVM et *virtio* permet à QEMU d'exécuter des machines virtuelles Linux extrêmement performantes et agiles. KVM est à la base de solutions telles que *RedHat Enterprise Virtualisation* où encore *OpenStack Compute* qui est présenté dans la suite de ce livre blanc.

Xen

Xen est initialement un logiciel de paravirtualisation, qui nécessitait l'utilisation d'un noyau spécial dans les machines virtuelles. Avec le temps, Xen s'est doté d'un mode de fonctionnement similaire aux machines virtuelles, afin de pouvoir virtualiser des systèmes tels que Windows, dont le noyau propriétaire interdit la paravirtualisation.

Initialement réservé à Linux, BSD et Solaris, la virtualisation Xen se tourne de plus en plus vers Windows **depuis le rachat de l'éditeur de Xen par *Citrix***. Simultanément, Xen est de moins en moins utilisé par les éditeurs de distribution Linux au profit de QEMU. Cependant **Xen est toujours aujourd'hui la référence pour la virtualisation haute performance sous Linux**, puisque plus ancien et bien ancré dans les versions actuelles des principales distribution Linux orientées entreprise : RHEL et SLES. **Xen a été récemment intégré au noyau Linux** ce qui garantit sa pérennité.

Les Principales Solutions

OpenVZ

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

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.

Rappelons que la technologie d'isolation ne permet d'exécuter que des serveurs virtuels Linux sur un hôte OpenVZ, même si ces serveurs peuvent être de distributions différentes.

Certaines distributions Linux proposent des versions packagées d'OpenVZ. En particulier, la distribution *Debian GNU/Linux,* dans les versions *Lenny* et *Squeeze,* permet dès l'installation du serveur physique de mettre en place cette solution en quelques secondes via son système de packages. Il faut cependant noter que OpenVZ sera remplacé par LXC dans la future version *Squeeze*, à paraître en 2013.

Présentation

Le projet OpenVZ 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 multiples 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

OpenVZ a été initialement développé par la société *SWSoft*, dans le but de fournir à des hébergeurs un moyen de disposer d'un grand nombre d'environnements séparés sur un petit nombre de serveurs physiques. Après quelques années d'existance, l'éditeur créa le projet OpenVZ en 2005 pour continuer à développer ce produit, nommé *Virtuozzo* en suivant les principes de l'open source.

Aujourd'hui, OpenVZ est le « moteur » du produit commercial *Paralells Virtuozzo Containers* utilisé par de nombreuses entreprises dans le monde, et constitue en lui-même un produit utilisable et bien intégré aux distributions Linux.

OpenVZ travaille à intégrer le plus possible de ses fonctions dans le noyau Linux afin de faciliter la maintenance et assurer la pérennité du projet.

Principe

OpenVZ 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 instances de l'OS sur la même machine. Toutefois elles partagent quand même le noyau, ce qui veut dire que si celui-ci se retrouve en défaut, alors tous les contextes le sont.

OpenVZ 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 OpenVZ découle des outils déjà existants sous Linux comme la barrière *chroot* et 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 d'administration.

Limitations

Du fait que les serveurs virtuels utilisent le noyau de l'OS hôte, OpenVZ 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 « conteneurs » 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 inférieure 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, comme le contrôle de l'horloge ou des paramètres du noyau 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.

Capacités

Contrairement à Linux-Vservers, son principal concurrent, présenté dans l'édition précédente de ce livre blanc, OpenVZ permet la virtualisation de la couche réseau (filtrage, routage, etc.), ainsi que la migration à chaud entre deux hôtes identiques.

Il est aussi possible de limiter finement l'utilisation des ressources de l'hôte pour chaque serveur virtuel, en particulier la mémoire résidente (RSS) ou la mémoire virtuelle (VSZ). On peut aussi mettre en place des quotas d'utilisation de l'espace disque. Il est enfin possible de mettre en place des priorités d'accès au CPU et aux disques.

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 Citrix, qui a racheté l'éditeur initial *XenSource*.

Citrix distribue une version commerciale de Xen, nommée *Citrix XenServer*, particulièrement adaptée à la virtualisation des OS Microsoft Windows et Linux RHEL et SLES. Elle est dotée 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 ont contribué au développement de Xen, et de gros efforts ont été faits par Citrix pour assurer une compatibilité parfaite avec Windows, compatibilité aujourd'hui reconnue par Microsoft.

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 aux serveurs virtuels considérés comme « critiques ».

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, Solaris 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 qu' OpenVZ, 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

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, et Solaris.

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.

Des pilotes de périphérique paravirtualisés permettent alors de retrouver les performances du mode paravirtualisé de Xen, ils sont disponibles librement pour Linux et dans la version commerciale de Citrix pour Windows.

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 interruption 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.

Notons que Xen est à la base de l'offre *cloud* EC2 de Amazon, qui permet d'allouer des serveurs virtuels à la demande.

Limitations de Xen

Le principal reproche qui peut être fait à Xen est le manque d'ergonomie de la distribution libre, qui 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 sont peu documentées. Ce qui fait de Xen une solution puissante, mais parfois délicate à appréhender et qui requiert une certaine expertise. Certains se tourneront plus volontiers vers des versions commerciales, plus faciles d'accès.

L'édition précédente de ce livre blanc émettait quelques réserves quant à la pérennité de Xen, dûes à la politique de développement de Citrix qui semblait se concentrer sous Windows. De plus, l'intégration de Xen dans le noyau Linux semblait peu probable.

Aujourd'hui, ces craintes n'ont plus lieu d'être, Xen est désormais intégré au noyau Linux, suite à un travail de longue haleine. Et le développement du produit continue même si celui-ci subit de plus en plus la concurrence de KVM, développé par RedHat.

Domaines d'Application

Nous allons maintenant voir quelques exemples d'application de ces techniques de virtualisation, dans les domaines où elles sont couramment mises en place.

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 au plan technique, 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 OpenVZ, 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, et pourra utiliser des applications fonctionnant en mode noyau tels des systèmes de stockage ou réseau avancés (GFS, DRBD, IPsec, etc.).

Plateforme 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.

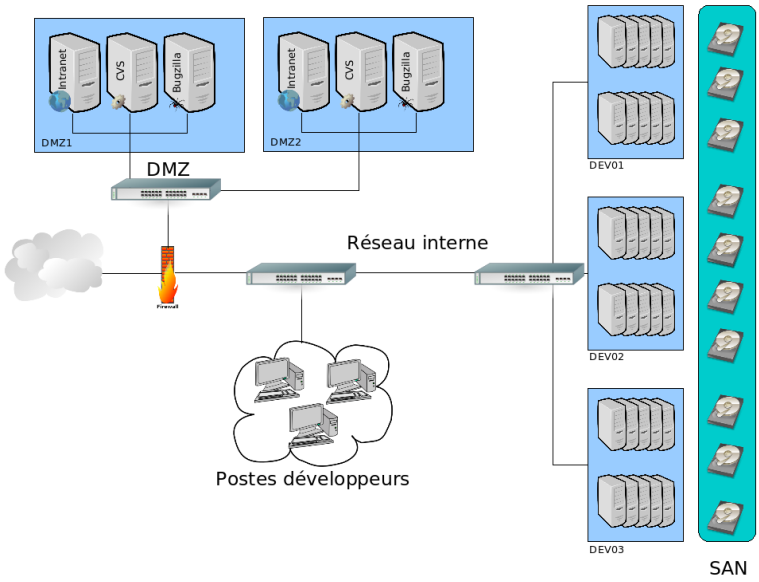
Les solutions de virtualisation permettent d'alléger quelque peu ces infrastructures de validation, leur coût matériel, mais aussi leur exploitation.

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 tenue en charge d'une application, en l'hébergeant sur plusieurs serveurs qui se partagent les visiteurs.

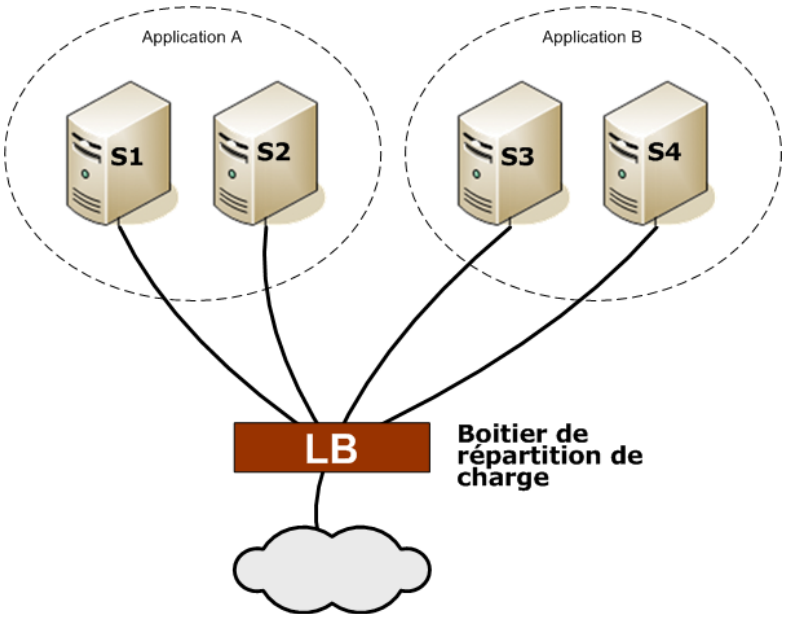
La répartition de charge est le plus souvent mise en œuvre au moyen d'un boitier spécialisé, qui dirige les requêtes des visiteurs sur les différents serveurs, en conservant ou non un même visiteur sur un même serveur. Les boitiers de répartition de charge savent en général détecter la panne d'un serveur, 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.

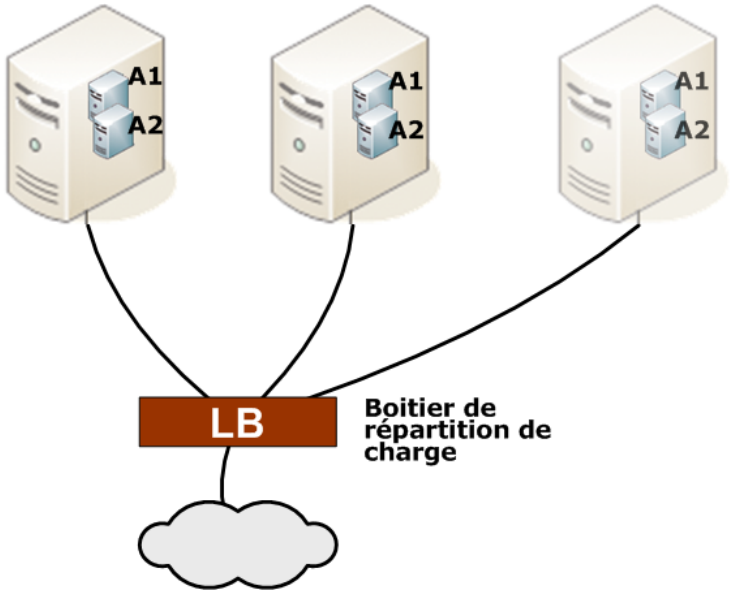
Mais si l'on est en présence de plusieurs applications, ayant chacune besoin de répartition de charge sur plusieurs serveurs, alors la virtualisation peut apporter une meilleure mutualisation de moyens.

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



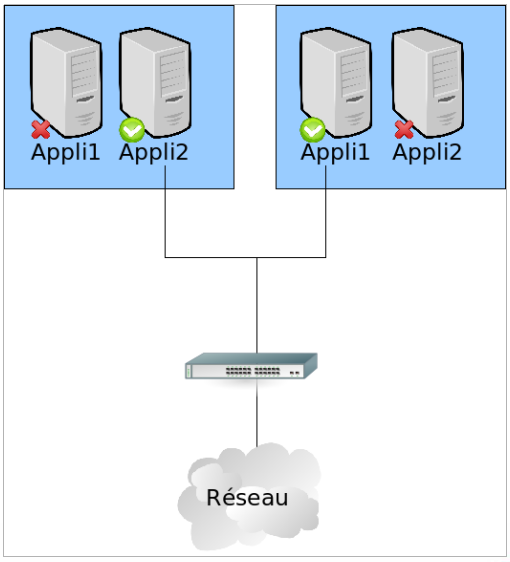
Architecture virtualisée, plateformes applicatives mutualisées

Reprise automatique

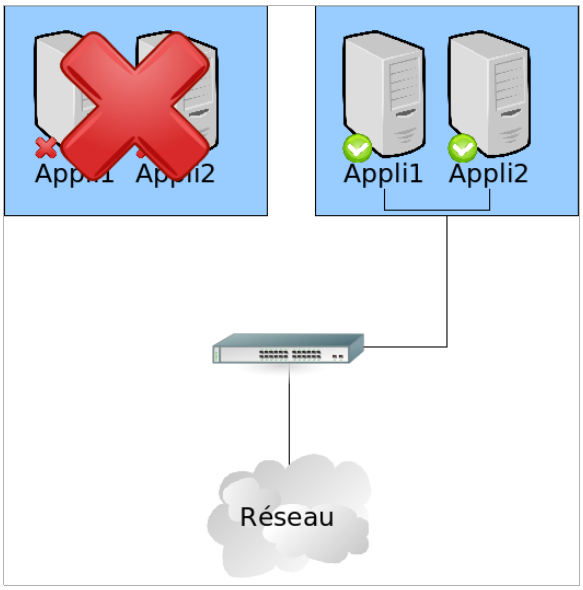
Un autre usage de la virtualisation dans une optique de haute disponibilité de service peut consister à avoir sur plusieurs serveurs physiques les mêmes environnements virtuels (synchronisés régulièrement).

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 boîtiers prêts à l'emploi : firewall, routeurs, solutions de sécurité tout-en-un, qui se branchent facilement sur le réseau et nécessitent 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 paquet 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 open source : 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 déployé 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 (eZ Publish, 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, de *endpoint* VPN, totalement isolé du matériel, et donc moins sensible en cas d'attaque.

Cloud Computing

L'une des idées fortes qui se cache derrière la notion, un peu vague, de Cloud Computing (informatique nébuleuse), est **l'abstraction de la plateforme d'une application**, à différents niveaux.

On parle d'**IaaS** (Infrastructure as a Service) lorsque l'on abstrait uniquement l'infrastructure physique d'une application. On conserve ainsi la notion de serveur, sur lequel une plateforme applicative (LAMP, ou encore J2EE) reste à installer.

On parle de **PaaS** (Platform as a Service) lorsque l'on abstrait cette fois-ci la plateforme, c'est ce que font des services comme Microsoft Azure ou Google App Engine.

Enfin, on parle de **SaaS** (Software as a Service) lorsqu'on abstrait tout, y compris l'application. Ce qui est ni plus ni moins que la fourniture « traditionnelle » de service payant via une application web. Les fers de lance de cette approche sont par exemple SalesForce ou encore Google Apps.

La virtualisation est bien sûr fondamentale dans la mise en place d'une IaaS. Cependant, il est nécessaire d'automatiser entièrement la mise à disposition de machines virtuelles. Le suivi de consommation des ressources à des fins de facturation fait partie du modèle commercial du Cloud Computing, il doit donc faire partie intégrante de la solution d'IaaS. Le produit doit également s'occuper de configurer automatiquement un espace de stockage persistant pour les VM, ainsi que leur fournir une connectivité réseau. Fournir cette connectivité nécessite à son tour de configurer des équipements tels que des switchs, et d'immobiliser des adresses IP. Enfin, l'accès, même indirect, des clients finaux à la plateforme physique du prestataire IaaS pose de nombreux problèmes de sécurité, de traçabilité, d'isolation entre les clients.

Une solution d'IaaS est donc un logiciel complexe, qui doit jongler entre de nombreux domaines d'activité. On comprend pourquoi les solutions de virtualisation ne prennent pas en charge directement l'IaaS. Des projets logiciels spécialisés sont nécessaires, et sont d'ailleurs souvent indépendants de la technologie de virtualisation sous-jacente. Dans la suite de ce livre blanc nous présenterons **OpenStack** qui est l'un de ces outils.

Le stockage

Dans tout projet de virtualisation se pose, à un moment ou un autre, la question du stockage. En effet, comme pour le calcul ou les entrées-sorties, la virtualisation implique généralement une couche d'abstraction supplémentaire au niveau du stockage des données. Il peut s'agir simplement de créer une arborescence de répertoires dans le cas d'un isolateur, ou de mettre en place un réseau de stockage hautes performances disposant de capacités avancées de réplication et de clichés dans le cas d'une plateforme de machines virtuelles.

Différents besoins

Quelque soit la technologie utilisée, une machine virtuelle se compose de deux élements :

- Des ressources : part de CPU alloués, mémoire vive autorisée, nombre de cartes réseau virtuelles...
- Des données : comme un serveur normal, on doit disposer d'un système d'exploitation, de bibliothèques, d'outils, d'applications et de leurs données.

Nous avons vu dans les parties précédentes comment étaient gérées les ressources, via diverses technologies de virtualisation et les fonctionnalités qui vont avec.

Le stockage, lui dépend généralement de la technologie de virtualisation utilisée, et surtout de sa « profondeur ».

Dans les technologies d'isolation, la virtualisation se fait au niveau de l'OS, et ne nécessite pas un dispositif de stockage particulier : chaque environnement virtualisé se présente sous la forme d'une arborescence gérable depuis le domaine de contrôle. Cette arborescence peut, de façon transparente, être située physiquement sur la même machine, sur un autre disque, sur un serveur distant, sur un réseau de stockage, etc. C'est la solution qui offre la plus grande souplesse.

Dans les technologies de machine virtuelle, l'hyperviseur ne fournit au système virtualisé qu'un espace de stockage. Il peut s'agir d'un volume, ou simplement d'un fichier, mais dans les deux cas cet espace est «hermétique» et ne peut être accédé depuis le domaine de contrôle. La encore, on peut placer l'intégralité de cet espace sur un disque local, un réseau de stockage, un autre serveur...

Dans ces deux cas, l'utilisation d'un disque local est la plus avantageuse en terme de performances et de facilité d'administration. Cependant, l'utilisation d'un stockage en réseau permet d'ouvrir la voie à de nouvelles fonctionnalités.

Stockage en réseau

Les pleines capacités des hyperviseurs modernes ne peuvent s'exprimer qu'au travers d'un stockage en réseau, en effet les hyperviseurs sont généralement gérés sous forme de «pools», formant une «force de travail» globale qui se partageront les machines virtuelles à

exécuter. Cette vision n'est possible que si le stockage est lui aussi unifié : sans cela chaque hyperviseur ne peut faire tourner que les serveurs virtuels présents sur son disque local, et n'est donc pas interchangeable.

Disposant d'un réseau de stockage, chaque hyperviseur a accès à toutes les machines virtuelles, et peut donc exécuter n'importe laquelle, et la transférer sans interruption à un autre hyperviseur en fonction de sa charge.

Seul OpenVZ permet la migration d'environnements à chaud quand les serveurs physiques ne partagent pas l'espace de stockage. Dans tous les autres produits, disposer d'un stockage réseau partagé est une condition nécessaire.

Nous allons présenter quelques technologies permettant de mettre en place un réseau de stockage.

NAS et NFS

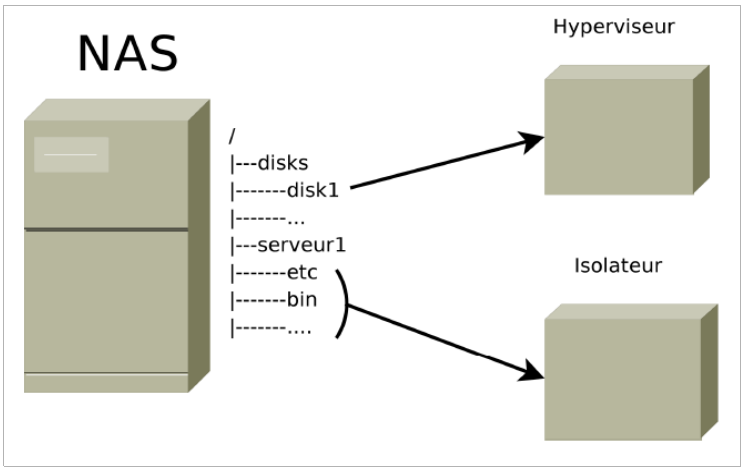
Un NAS, ou stockage réseau (*Network-Attached Storage)* est simplement un serveur fournissant leurs fichiers à d'autres serveurs par le réseau.

NFS est le standard universel pour l'accès aux fichiers sur un réseau, c'est le protocole le plus utilisé dans les NAS.

Dans le cadre d'un isolateur, il permet de stocker l'arborescence du serveur virtuel à distance. Dans le cadre d'une solution de virtualisation complète il permet de stocker à distance les fichiers contenant les disques durs de la machine virtuelle.

Ce dernier cas est déconseillé hors des environnements de test : NFS n'est pas adapté à la lecture aléatoire dans un seul fichier. En revanche pour un isolateur, stocker les données en NFS est intéressant, et le deviendra encore plus avec les systèmes de fichier de nouvelle génération tels que ZFS, HAMMER ou *btrfs*, qui permettent des snapshots instantanés, le versionnement des arborescences, et autres fonctionnalités pour l'instant réservées aux baies de stockage haut de gamme.

En plus des matériels dédiés, la plupart des systèmes d'exploitation proposent une implémentation serveur NFS, ce qui permet d'utiliser n'importe quel serveur comme serveur de stockage NFS. Ces derniers utilisent alors soit des disques locaux, soit leur propre réseau de stockage SAN.



SAN

Un SAN, ou réseau de stockage (*Storage Area Network)*, est un réseau sur lequel circulent les données entre un système et son stockage. Cette technique permet de déporter tout le stockage interne d'une machine vers un équipement dédié.

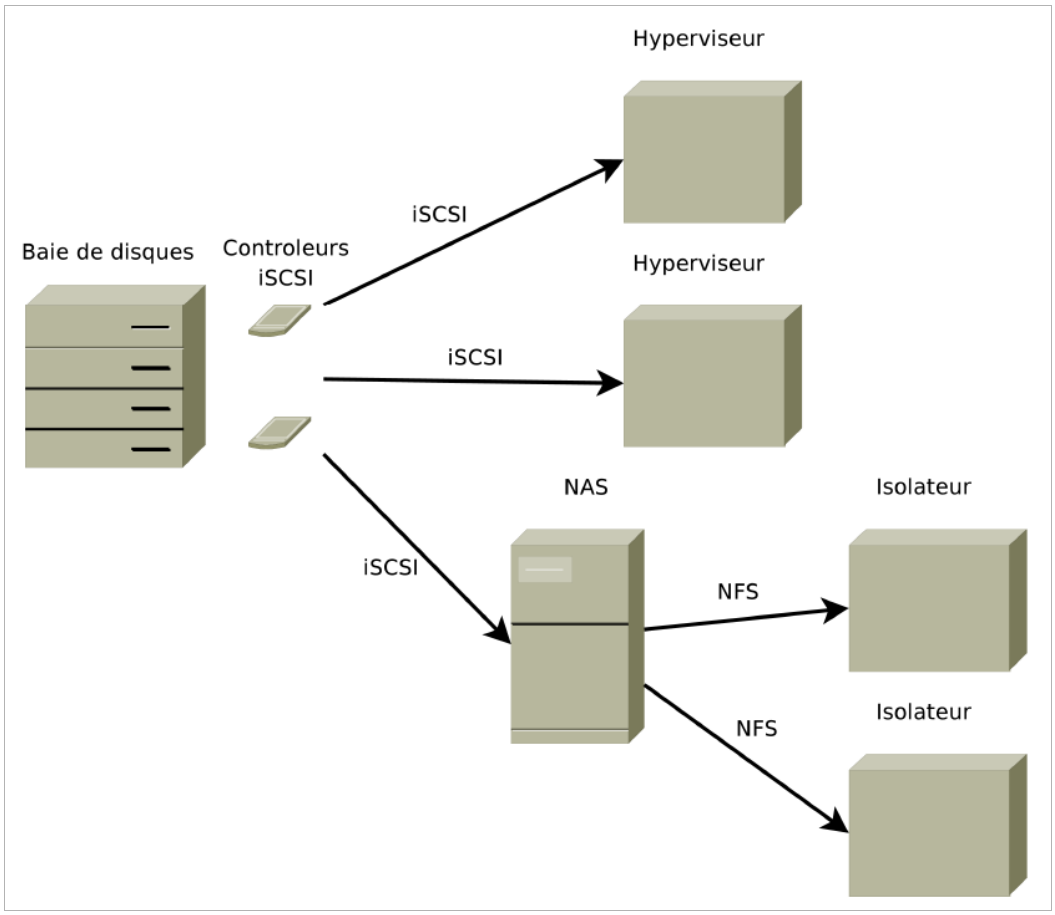
Les SAN sont des équipements dédiés, qui ne travaillent qu'aux plus basses couches du stockage, la notion de fichier leur est inconnue ; ils travaillent simplement sur des blocs de données et les fournit par le réseau à des serveurs qui eux sauront les utiliser. Cependant, les SAN les plus hauts de gamme sont dotés de capacités avancées, tel que la prise de cliché, ou encore la copie rapide de volumes.

Les deux principaux protocoles d'accès à un SAN sont iSCSI et Fibre Channel.

iSCSI

iSCSI est un protocole d'accès disque fonctionnant sur un réseau Ethernet, il permet d'implémenter un réseau de stockage en profitant de la connectique et des équipements de commutation standards. Comme le NFS, il peut être soit implémenté par une baie de stockage dédiée, ce qui assure les meilleures performances, soit par un serveur classique disposant du logiciel adéquat, par exemple IET (*iSCSI Entreprise Target)* sous Linux.

Voici un exemple de SAN : parmi les machines clientes du SAN, on retrouve un NAS : ces deux techniques peuvent être combinées car elles ne travaillent pas au même niveau.



Fibre Channel

La solution la plus haut-de-gamme pour implémenter un réseau de stockage est l'utilisation d'une baie dédiée et du protocole *Fibre Channel.* Basé sur des fibres optiques il assure une latence et un débit bien meilleurs que iSCSI, à un prix bien sûr plus élevé. Son principe d'utilisation est le même qu'un SAN iSCSI.

Critères de choix

Le choix d'une solution de stockage est basé sur la taille de l'infrastructure, le niveau de fiabilité attendu, et les fonctionnalités. Un stockage en réseau des VM apporte une plus grande flexibilité, et la possibilité de facilement rajouter des nœuds à l'infrastructure, mais au prix d'un investissement initial élevé (un réseau de stockage coute cher) et d'une architecture plus complexe.

Bien souvent, le stockage en local des VM, et une bonne politique de sauvegarde permet une reprise d'activité rapide en cas de problème sur un hôte, et on préférera implémenter la haute disponibilité au niveau applicatif plus qu'au niveau système, via par exemple un répartiteur de charge, ou une solution telle que *Linux Virtual Server, CARP,* et autres.

INTERFACES D'ADMINISTRATION

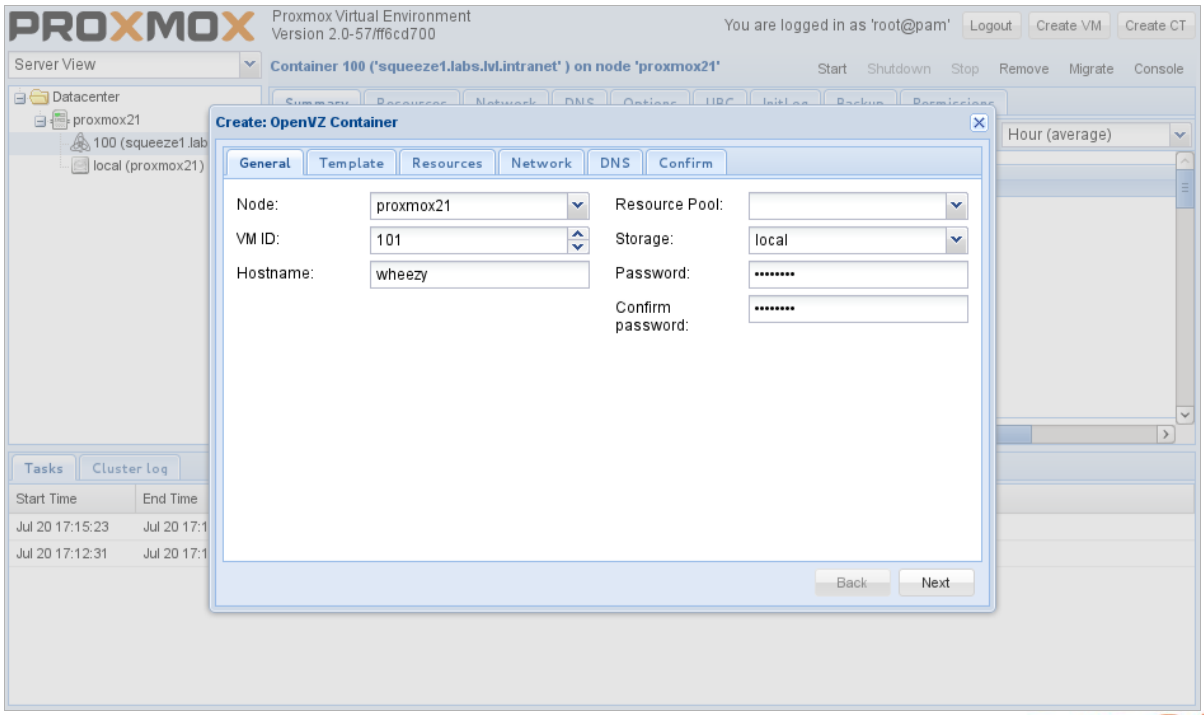
Si la plupart des solutions open source de virtualisation sont matures en ce qui concerne les fonctionnalités de base, il n'en est pas autant de leurs interfaces d'administrations, souvent minimalistes. Or, le succès commercial d'un produit ne dépend pas que de la qualité technique de son socle, mais également de sa facilité d'accès. Il est un principe fondamental de l'open source qu'un projet n'a pas pour but de tout faire, ainsi trouve-t-on des projets open source dont le but est uniquement de fournir des interfaces d'administration à des produits existants, laissant aux projets développant les briques de base le soin de se concentrer sur les problématiques techniques bas niveau.

PROXMOXVE

Proxmox VE est une plateforme de virtualisation open source dite « Bare Metal » produite, développée et maintenue principalement par **Proxmox Server Solutions GmbH** depuis 2008. Elle fournit un système minimal permettant de créer et de gérer des machines virtuelles en usant de technologies ouvertes d'isolation (OpenVZ), de virtualisation complète (KVM) et de clustering (Corosync), le tout via une interface web d'administration accessible.

Les deux mondes sur une plateforme simple d'accès

Proxmox VE fournit un accès simple à l'isolation et à la virtualisation complète sur une seule et même plateforme. Il est ainsi possible de créer et gérer des machines virtuelles avec KVM et des conteneurs avec OpenVZ, le tout via une interface d'administration moderne et simple d'accès.



Capture d'écran ProxMox – Source : www.proxmox.com

La mise en place d'un cluster et la haute disponibilité sont de plus à portée de main avec une configuration des machines virtuelles et conteneurs pour lesquels assurer une continuité de service, ainsi que la configuration du fencing avec support IPMI, APC et DRAC (Dell). Il est aussi possible d'effectuer des tâches de sauvegarde ponctuelles depuis celle-ci.

Il est possible d'installer la plateforme sur une Debian Squeeze, la société fournissant des dépôts pour la distribution incluant toutes les dépendances nécessaires.

Retours d'expérience

Proxmox VE est selon nous une plateforme adaptée pour les infrastructures de toutes tailles. Elle fournit toutes les fonctionnalités de base que l'on peut attendre d'un tel produit, telles la gestion du réseau, la migration et la sauvegarde (complète et snapshots) des machines virtuelles et conteneurs, la gestion et le monitoring des ressources utilisées, mais possède quelques lacunes.

Par exemple, il est impossible de gérer l'accès aux VLAN depuis l'interface d'administration. Il est de même impossible de n'utiliser que iSCSI si l'on veut avoir de la haute disponibilité sur les conteneurs et machines virtuelles, les conteneurs ne supportant que NFS pour la haute disponibilité actuellement.

La plateforme fournissant son propre noyau allégé, il est aussi impossible d'installer son propre noyau incluant des fonctionnalités spécifiques. A titre d'exemple, le noyau fourni ne supporte pas le RAID logiciel en conséquence de cet allègement.

Enfin, la création d'un cluster Proxmox VE et l'ajout de nœuds ne se fait pas via l'interface web mais en ligne de commande sur les serveurs, alors que la gestion du cluster peut se faire via l'interface web. Ce n'est en rien gênant, mais c'est un manque de l'interface web que l'on aimerait voir combler.

La plateforme est cependant suffisante pour des besoins de virtualisation basiques, vu qu'il reste possible d'effectuer certaines tâches non implémentées sur l'interface d'administration en ligne de commande, telle la création de bridges liés à des VLAN.

Limitations de Proxmox VE

Proxmox VE implémente un monitoring de l'utilisation des ressources par les machines virtuelles ainsi qu'une supervision basique des machines virtuelles via Ping dans le cadre de la haute disponibilité, suffisant pour s'assurer que le système est disponible, mais cela implique que le système ne verra pas une indisponibilité de service. Il est donc nécessaire de prévoir une supervision externe de type Centreon/Nagios pour être averti des indisponibilités de service.

OpenStack

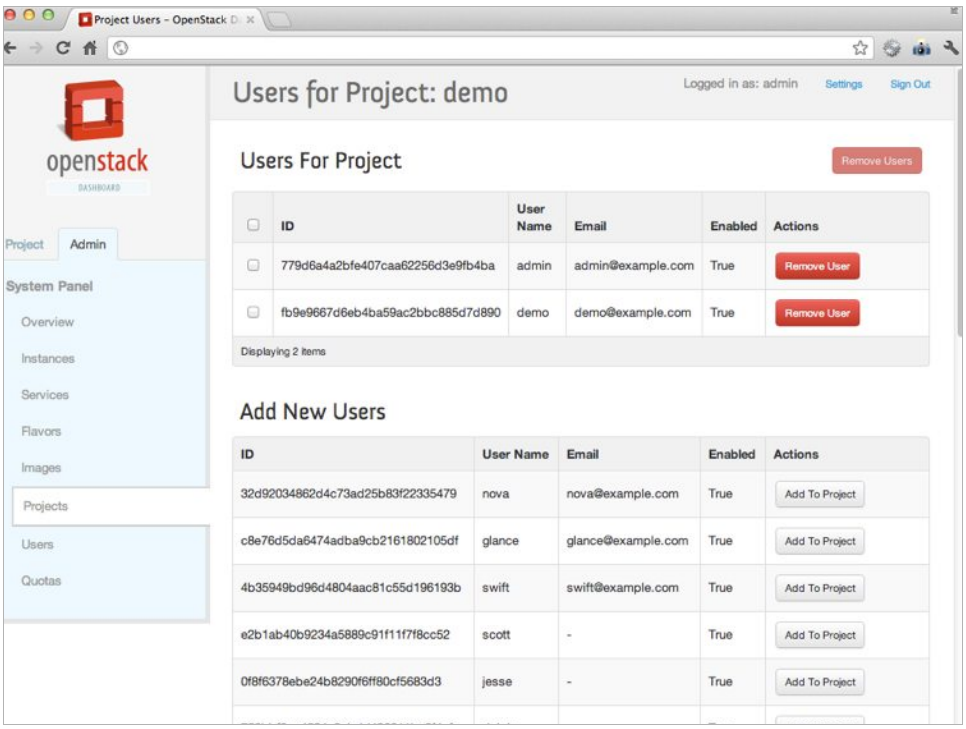
Présentation

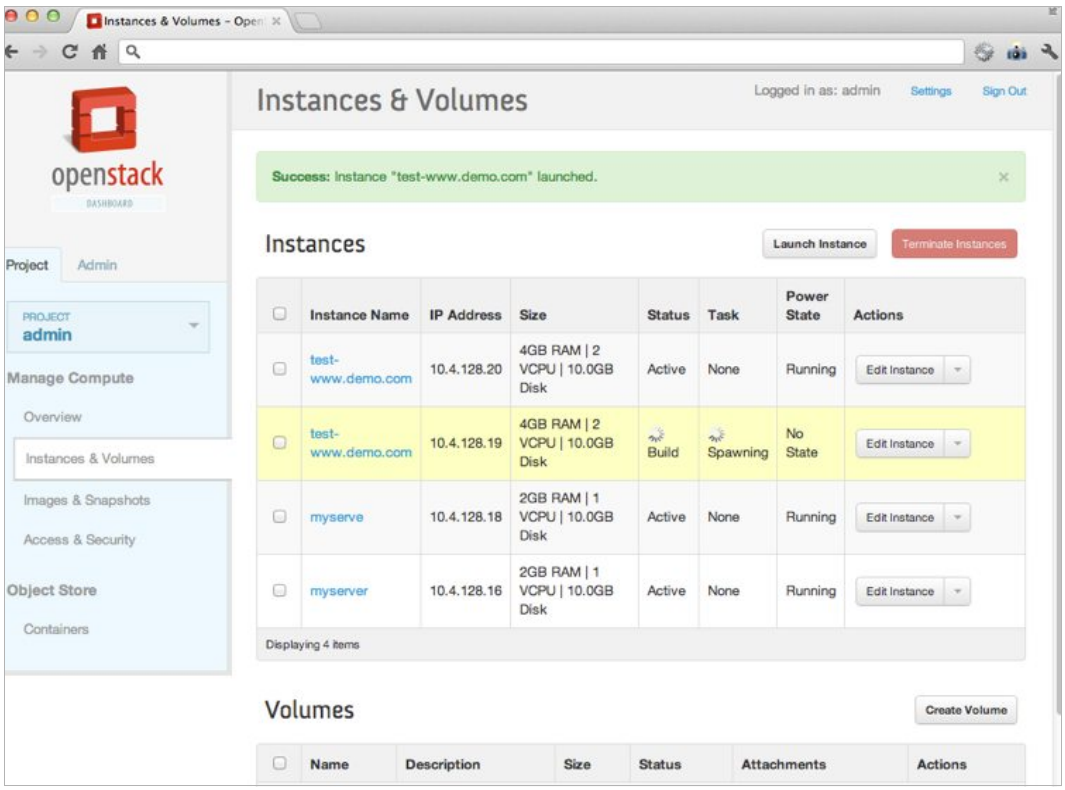
OpenStack est un logiciel beaucoup plus complet et ambitieux que ProxmoxVE. Son but est de fournir toutes les briques nécessaires à la mise en place d'un service IaaS : Infrastructure as a Service.

L'IaaS est l'une des facettes du Cloud Computing : il s'agit de fournir des ressources informatiques sous forme de machine virtuelles prêtes à l'emploi, et ce directement à la demande des utilisateurs, sans passer par la mise en place à la main des VM par un administrateur système. Le système IaaS gère la mise à disposition de la ressource en fonction de la capacité actuelle de la plateforme, et assure le suivi de consommation à des fins de facturation.

OpenStack, développé initialement par Rackspace et la NASA, propose un regroupement de logiciels open source sous licence Apache. Ce regroupement permet de mettre à disposition des ressources de calcul (des machines virtuelles) ainsi que des espaces de stockage, en reprenant les principes de l'IaaS.

Historiquement, Rackspace contribuait à la partie stockage d'OpenStack tandis que la partie calcul était développée par la NASA. A l'heure actuelle la fondation en charge du projet, nommé OpenStack Consortium, intègre plus de 150 entreprises, dont Canonical, Dell et Cisco.

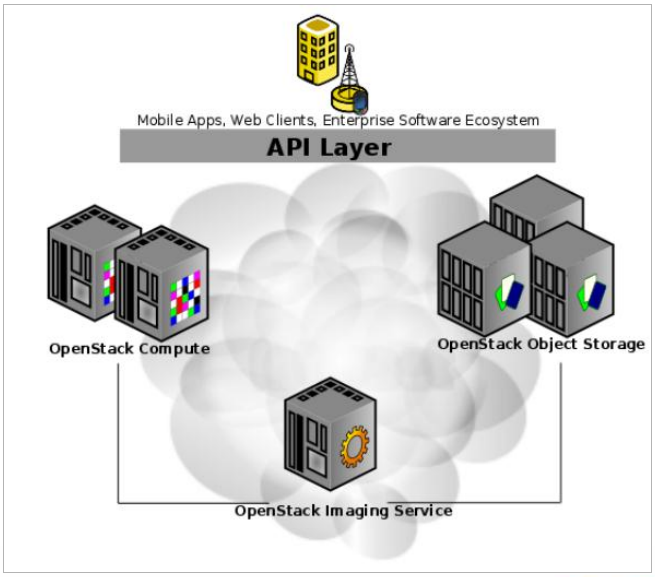




Captures d'écran OpenStack Dashboard – Source : www.openstack.org

Architecture du produit

Au niveau macroscopique, OpenStack se compose de 3 services principaux, chacun fournissant une API qui permet aux clients de demander des ressources. Fournir une API et non seulement une interface permet d'**automatiser la mise à disposition de ressources**, ce qui permet d'augmenter la capacité d'une plateforme particulière de façon totalement automatique, dans la limite de la puissance mise à disposition par l'infrastructure OpenStack.



Source : documentation officielle OpenStack

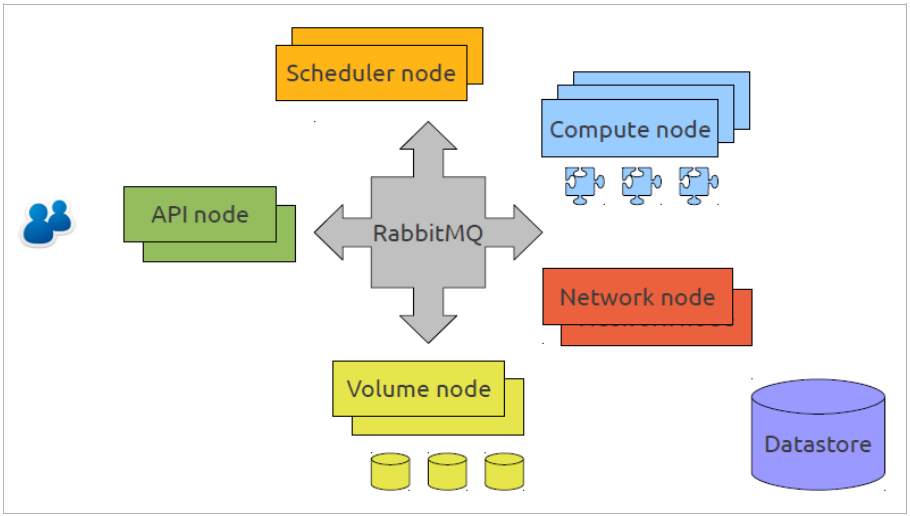
« OpenStack Compute » est le service qui instancie des machines virtuelles, c'est l'équivalent d'Amazon Elactic Compute Cloud (EC2)

« OpenStack Object Storage » met à disposition des clients un espace de stockage pour tous types de données : d'une simple image à une machine virtuelle entière. C'est l'équivalent d'Amazon Simple Storage Service (S3)

« OpenStack Imaging Service » est la brique qui fait le lien entre les deux produits et permet le stockage des modèles de VM sur Object Storage. Une fois ainsi stockés, ces modèles sont instanciables à la demande. Il est bien sûr possible pour les clients de créer de nouveaux modèles.

Par la suite, nous nous concentrerons sur OpenStack Compute, qui est le seul composant à utiliser les techniques de virtualisation présentées dans ce livre blanc.

Le schéma ci-dessous montre les principaux composants d'OpenStack Compute :



Source : schéma issu d'une présentation de Thierry Carrez lors de l'événement Openstack in action 2

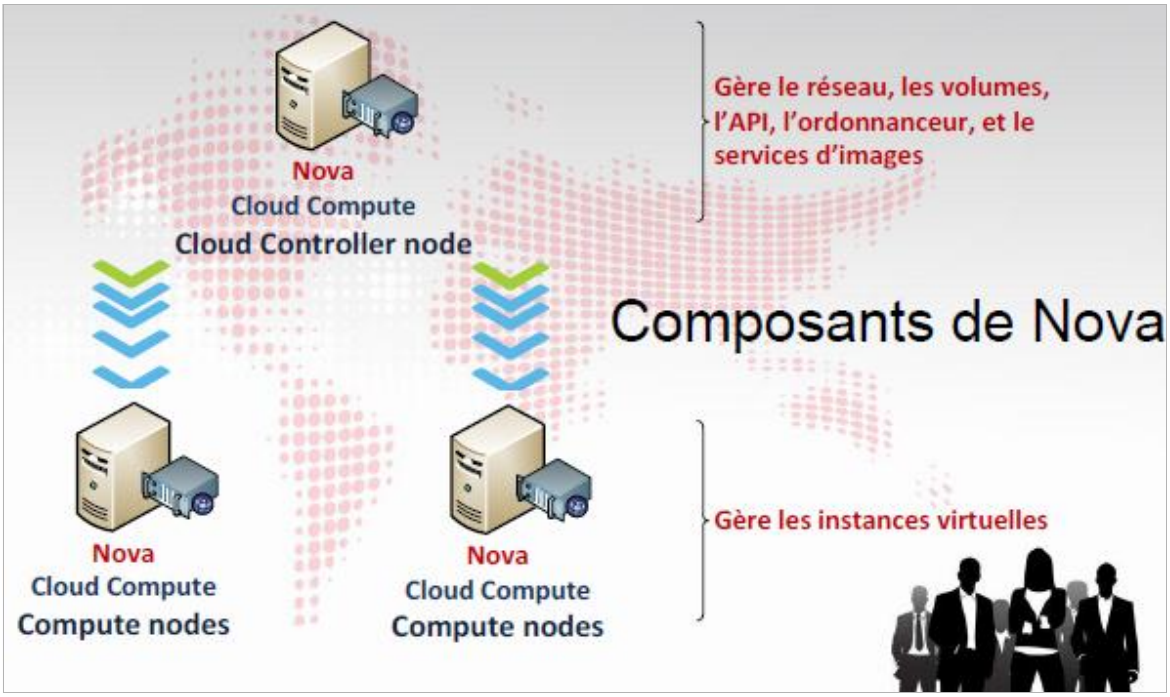
Les différents sous-composants de chaque service communiquent entre eux via un serveur de messages, en général RabbitMQ, qui garantie une communication fiable entre les services en toutes circonstances, y compris si l'un d'eux a rencontré une panne.

Parmi ces composants, on retrouve :

- Le serveur d'API, qui reçoit les demandes des utilisateurs et des autres services d'OpenStack, les valide, et les envoie à la file d'exécution.
- Le service de planification (scheduler) qui place les nouvelles VMs sur un des hôtes disponibles en fonction de critères pré-établis (priorités, etc.) mais aussi de l'état actuel de la plateforme (charge, indisponibilités, etc.)

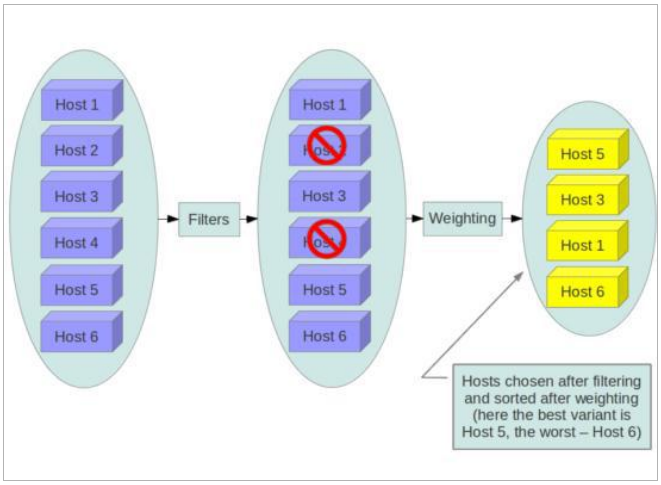
- Les services de réseau et de stockage, qui fournissent la connectivité réseau et l'espace de stockage aux VM.
- Les nœuds de calcul (compute node) eux même, qui sont les hyperviseurs.

Généralement, plusieurs de ces services sont regroupés sur la même machine, dégageant deux types de serveurs physiques : les nœuds de calcul et les nœuds de contrôle.



Source : http://my1.fr/blog/presentation-openstack/

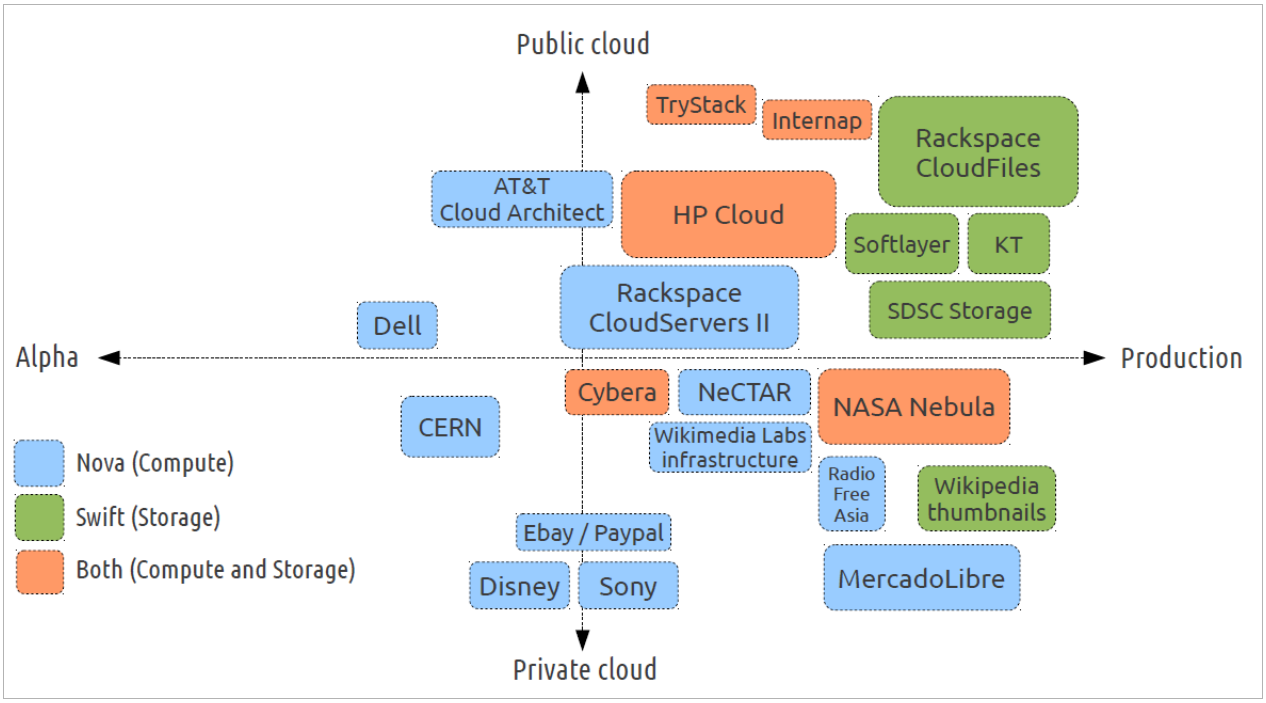
Un seul nœud de contrôle peut prendre en charge plusieurs nœuds de calcul, qui sont simplement des hyperviseurs qui exécutent des machines virtuelles basées sur les images fournies par OpenStack. Il n'est pas recommandé d'utiliser différents types d'hyperviseurs au sein d'une même installation d'OpenStack, en raison de l'incompatibilité des formats de disque, du matériel, et de l'impossibilité d'effectuer des migrations des VM d'un hôte vers un autre lorsque les deux hôtes utilisent un hyperviseur différent.



Source : documentation officielle OpenStack

Le choix de l'hôte lors de la création d'une nouvelle VM peut dépendre de plusieurs critères, par exemple, OpenStack peut choisir de créer une nouvelle VM sur la machine la moins chargée, et bien sûr ne créera pas de VM sur une machine temporairement indisponible. OpenStack peut tirer profit des fonctionnalités de migration à chaud des hyperviseurs lorsqu'elles sont disponibles afin de minimiser l'indisponibilité lors des phases de maintenance.

La pérennité d'OpenStack



Source : schéma issu d'une présentation de Thierry Carrez lors de l'événement Openstack in action 2

Sur ce graphique on peut voir quels étaient les principaux déploiements d'OpenStack. Il est à noter que la majeure partie des déploiements en productions sont constitués de la plateforme de stockage et non de calcul, qui reste surtout pour l'instant cantonnée à l'expérimentation ou à de la production à petite échelle.

La pérennité de OpenStack Compute est fortement liée à celle de l'hyperviseur choisi. En effet les différents hyperviseurs ne sont pas toujours bien supportés, avec des fonctionnalités non implémentées ou non testées. Par exemple, à ce jour seule une version relativement ancienne de VMWare est supportée. Plus précisément, la version *4.1 update 1*, sortie le 10 février 2011, alors que la version *5.0 update 1* est sortie depuis le 15 mars 2012. Initialement, HyperV de Microsoft était aussi supporté. Son intégration a finalement été abandonnée.

Le choix de l'hyperviseur est donc critique et dépend des besoins de la plateforme IaaS. Le lecteur trouvera à la fin de ce livre blanc un récapitulatif des principaux critères à considérer pour faire ce choix, auxquels il faudra ajouter le niveau de support par OpenStack : à l'heure actuelle, les hyperviseurs XenServer et KVM sont les mieux supportés.

OpenStack est packagé dans Ubuntu Server et dans Debian (Wheezy). D'autres distributions telles que RedHat Enterprise Linux sont moins bien supportées. Si la prise en charge de RedHat concentre de plus en plus d'efforts de développement, un des obstacles à l'intégration d'OpenStack dans les solutions de RedHat est actuellement son modèle de développement rapide dû à la jeunesse du produit. Vis-à-vis des 10 années de support proposées par RedHat aux entreprises, on comprend qu'une éventuelle intégration d'OpenStack à RedHat devra attendre que le produit soit suffisamment stable.

A ce jour, la solution fonctionnant 'out-of-the-box' est Ubuntu Server. L'intégration d'OpenStack et d'Ubuntu va même plus loin pour deux raisons. La première est l'implication de Mark Shuttleworth (fondateur et principal soutien financier d'Ubuntu). La seconde est le rythme de développement d'OpenStack. C'est-à-dire qu'une version majeure d'OpenStack sort tous les 6 mois en se calquant sur le rythme de sortie d'Ubuntu. Le choix d'OpenStack est motivé par la forte dynamique dont bénéficie Ubuntu : à l'heure actuelle, Ubuntu Server enregistre une forte progression en parts de marché sur les serveurs Web, et se situe environ au même niveau que RedHat, parfois même devant selon certaines études.

Un des freins à l'adoption d'OpenStack reste la faiblesse de son interface Web. A l'heure actuelle, une interface Web existe pour les opérations basiques (gestion des instances, gestion des IPs publiques…), mais pour un certain nombre d'actions d'administration ou d'utilisation, l'interface par ligne de commande (CLI) reste nécessaire. L'accent est clairement mis sur l'API, qui elle offre l'accès à toutes les fonctions.

Un autre frein réside dans la jeunesse d'OpenStack et son cycle de développement rapide, qui est de 6 mois, ce qui s'avère une période de temps trop courte pour de la production. Mais à l'instar d'Ubuntu, des versions 'LTS' (Long Time Support) d'OpenStack permet d'étendre tous les 2 ans le support d'une version.

CONCLUSION

SYNTHÈSE

	Isolateurs		Machines virtuelles	
	LXC	**OpenVZ**	**Xen**	**KVM**
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 contrainte au niveau du noyau, choisissez OpenVZ.

➔ Si vous gérez des environnements purement Linux, mais avec des besoins plus précis en termes de version noyaux, ou de fonctionnalités de haute-disponibilité, ou une grande hétérogénéité, choisissez Xen.

➔ Si vous gérez des environnements Linux, ainsi que quelques serveurs Windows où la performance n'est pas un impératif, choisissez également Xen.

➔ En environnement purement Windows avec des besoins de haute performance, et de simplicité d'utilisation, choisissez *XenServer* de Citrix.

➔ 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 l'expérimentation, essayez KVM.

L'AVENIR

Désormais bien installée dans le monde des serveurs, **la virtualisation s'attaque de plus en plus aujourd'hui au poste de travail**, où elle vise à régler tous les problèmes de déploiement et de maintenance. L'engouement est similaire à ce qu'il à été pour les serveurs, mais la problématique est plus complexe, car les contraintes d'utilisation sont plus fortes : il faut assurer une faible latence, des capacités graphiques à la hauteur du confort d'utilisation d'un poste dédié.

Côté serveur, 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 des ressources, et d'autre part cela permet de bénéficier des services qui ne sont pas liés à ce partage, par exemple la sauvegarde et reprise de l'environnement.

De plus, la solution d'IaaS OpenStack, portée par une énorme vague de la communauté open source permet d'aller encore plus loin dans la gestion et le déploiement de la virtualisation, et est en train de faire bouger les lignes face aux alternatives propriétaires.

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.

Pour bénéficier vous aussi des économies et de l'efficacité d'une infrastructure virtualisée, n'hésitez pas à faire appel aux ingénieurs système de Smile, ils seront heureux de mettre leur expertise à votre service.

Merci à Jonathan Raffre, Yanick Delarbre, Romain Gobinet, Patrice Bertrand et Badr Chentouf de Smile pour leur contribution à ce livre blanc.