

Edité par le Groupe LINAGORA



# LIVRE BLANC LA VIRTUALISATION





#### Les Livres Blancs LINAGORA:

Fidèles à notre culture d'entreprise technologique spécialisée Open Source, notre travail se base sur la mise en œuvre concrète des produits et leur comparaison sur la base de critères pratiques. L'analyse est confiée à nos spécialistes techniques, qui croisent leurs points de vue et leurs sources afin de conforter au maximum les résultats présentés. Évidemment, nous sommes preneurs de tous avis et contradictions, dans l'esprit de la culture Open Source. N'hésitez pas à nous solliciter à l'adresse suivante : wb@linagora.com.

#### Résumé:

Ce livre blanc présente un tour d'horizon des outils libres permettant la virtualisation.

Date de rédaction : Décembre 2007

#### Rédacteurs :

Benoît DONNETTE, Ingénieur Expert, Groupe LINAGORA, bdonnette@linagora.com.

David HANNEQUIN, Administrateur Système & Réseaux, Groupe LINAGORA, <a href="mailto:dhannequin@linagora.com">dhannequin@linagora.com</a>.

#### Licence et droits d'utilisation :

Ce document est publié sous licence Creative Commons @ 000 « CC BY-NC-SA » :

- CC : Creative Commons cf <a href="http://creativecommons.org/worldwide/fr/">http://creativecommons.org/worldwide/fr/</a>
- BY : Paternité. L'auteur conserve la paternité de son œuvre, et vous ne pouvez vous attribuer que vos propres contributions.
- NC : Pas d'Utilisation Commerciale (Noncommercial). Ce travail ne peut faire l'objet d'une utilisation commerciale<sup>2</sup>.
- SA : Partage des Conditions Initiales à l'Identique (Share Alike). Toute altération, transformation, tout produit dérivé de ce travail doit être distribué sous une licence identique ou similaire.

Crédit photo : Susanna Altarriba

<sup>1</sup> Vous trouverez le contrat de licence complet à <a href="http://creativecommons.org/licenses/by-nc-sa/2.0/fr/legalcode">http://creativecommons.org/licenses/by-nc-sa/2.0/fr/legalcode</a>



Page: 2 / 61

<sup>&</sup>lt;sup>2</sup> Pour une compréhension des usages autorisés, voir le tableau <u>www.adamfields.com/CC-NC-allowed-uses-flowchart.pdf</u>



# **Sommaire**

1. Introduction.	5
1.1 Intérêts	5
1.2 Historique	6
1.3 Comparaison des différentes techniques de virtualisation	7
1.3.1 Machine Virtuelle	8
1.3.2 Virtualisation d'OS, Isolateur	9
1.3.3 Hyperviseur complet	10
1.3.4 Paravirtualiseur	10
2. Le besoin	11
2.1 Contexte	11
2.1.1 Outils en place	12
2.1.2 Orientation.	12
2.1.3 Besoins identifiés	13
2.2 L'état de l'art	14
2.2.1 Fonctionnalités	14
2.2.2 Respect des standards	15
2.2.3 Scénarios d'utilisation	16
2.2.4 Ergonomie	17
2.2.5 Sécurité	17
2.2.6 Restrictions techniques	18
3. Les solutions.	19
3.1 Méthode de sélection	19
3.1.1 Présélection.	19
3.1.2 Comparaison des solutions sélectionnées	21
3.2 Solutions retenues	21
3.2.1 Qemu	22
3.2.2 KVM	
3.2.3 VirtualBox	29
3.2.4 Xen	32
3.2.5 OpenVZ	35
4. Synthèse de l'étude	39
4.1 Comparaison des solutions retenues	39
4.2 Fiche d'identité des logiciels	
4.2.1 Qemu	
4.2.2 Kym.	42





4.2.3 VirtualBox	45
4.2.4 Xen	48
4.2.5 OpenVZ	51
4.3 Perspectives des technologies	55
4.4 Comment faire ?	57
4.4.1 Réalisation d'un audit des besoins de virtualisation	57
4.4.2 Configuration du ou des outils les plus appropriés et réalisation d'une d'administration	
4.5 Méthodologie	58
4.5.1 Définition des critères communs	58
4.5.2 Définition des niveaux pour chaque critère	60





# 1. Introduction

Virtualiser : proposer, par l'intermédiaire d'une couche d'abstraction proche du matériel, une vue multiple d'un matériel unique, en sérialisant les appels vus concurrents de l'extérieur.

La virtualisation recouvre l'ensemble des techniques matérielles et/ou logicielles qui permettent de faire fonctionner sur une seule machine plusieurs systèmes d'exploitation, plusieurs instances différentes et cloisonnées d'un même système ou plusieurs applications, séparément les uns des autres, comme s'ils fonctionnaient sur des machines physiques distinctes.

Chaque outil de virtualisation implémente une ou plusieurs de ces notions :

- couche d'abstraction matérielle et/ou logicielle,
- système d'exploitation hôte (installé directement sur le matériel),
- systèmes d'exploitations (ou applications, ou encore ensemble d'applications)
   « virtualisé(s) » ou « invité(s) »,
- partitionnement, isolation et/ou partage des ressources physiques et/ou logicielles,
- images manipulables : démarrage, arrêt, gel, clonage, sauvegarde et restauration, sauvegarde de contexte, migration d'une machine physique à une autre réseau virtuel : réseau purement logiciel, interne à la machine hôte, entre hôte et invités.

## 1.1 Intérêts

#### Les intérêts sont :

- → utilisation optimale des ressources d'un parc de machines (répartition des machines virtuelles sur les machines physiques en fonction des charges respectives),
- → installation, déploiement et migration facile des machines virtuelles d'une machine physique à une autre, notamment dans le contexte d'une mise en production à partir d'un environnement de qualification ou de pré-production, livraison facilitée,
- → économie sur le matériel par mutualisation (consommation électrique, entretien physique, monitoring, support, compatibilité matérielle, etc.),
- → installation, tests, développements, réutilisation avec possibilité de recommencer arrêt du système hôte,
- → sécurisation et/ou isolation d'un réseau (arrêt des systèmes d'exploitation virtuels, mais pas des systèmes d'exploitation hôtes qui sont invisibles pour l'attaquant, tests d'architectures applicatives et réseau),







- → isolation des différents utilisateurs simultanés d'une même machine (utilisation de type site central),
- → allocation dynamique de la puissance de calcul en fonction des besoins de chaque application à un instant donné,
- → diminution des risques liés au dimensionnement des serveurs lors de la définition de l'architecture d'une application, l'ajout de puissance (nouveau serveur etc.) étant alors transparent.

Notons que l'INRIA travaille actuellement sur le projet Kerrighed³, certes orienté calcul, qui fait **exactement le contraire** : faire un serveur virtuel unique à partir de plusieurs machines. Si cette approche semble à première vue opposée à la virtualisation présentée ici, nul doute que la solution ultime proposerait **p** serveurs répartis sur **n** machines, assurant redondance, équilibrage de charge, haute disponibilité.

# 1.2 Historique

Une bonne part des travaux sur la virtualisation fut développée au centre de recherche IBM France de Grenoble (aujourd'hui disparu), qui développa le système expérimental CP/CMS, devenant ensuite le produit (alors nommé hyperviseur) VM/CMS, proposé au catalogue dès 1972.

Par la suite, les mainframes ont été capables de virtualiser leurs OS avec des technologies spécifiques et propriétaires, à la fois logicielles et matérielles.

Les grands Unix ont suivi avec les architectures NUMA des Superdome d'HP (PA-RISC et IA64) et des E10000/E15000 de Sun (UltraSparc).

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

La société VMware développa et popularisa au début des années 2000 un système propriétaire de virtualisation logicielle pour les architectures de type x86. Les logiciels libres Xen, Qemu, Bochs, Linux-VServer et les logiciels propriétaires mais gratuits VirtualPC et VirtualServer ont achevé la popularisation de la virtualisation dans le monde x86.

Cependant, les solutions envisageables sur architecture x86 sont longtemps restés bridés par les spécificités de l'architecture x86 et surtout par son utilisation. En effet, cette famille de processeurs (depuis le 80386) offre, pour l'écriture de systèmes, 4 niveaux de privilèges, mais à priori tous les systèmes d'exploitation ont pris le parti de placer le système au niveau le plus privilégié (ring 0), et les applications au niveau le plus faible (ring 3), sans penser à se rendre plus indépendant des



<sup>3</sup>http://www.kerrighed.org



niveaux d'exécution. Par conséquent, 2 niveaux de privilèges sont perdus, et écrire une couche de virtualisation pour embarquer des systèmes d'exploitation complets, nécessitant des privilèges supérieurs à ceux du ring 0, ne peut plus se faire que par émulation.

C'est pourquoi les fabricants de processeurs x86 AMD et Intel ont ajouté dans leurs gammes des instructions dédiées afin de proposer des solutions de virtualisation matérielle dans la seconde moitié des années 2000. Ces instructions permettent la coexistence de plusieurs "ring 0" simultanée, on parle parfois par extension à leur sujet de "ring -1".

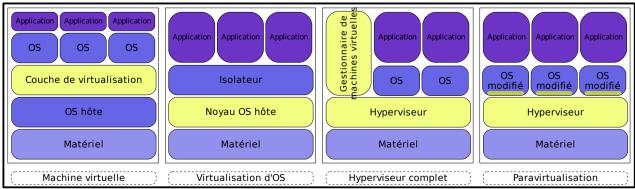
Aujourd'hui, les principaux éditeurs de solutions propriétaires de virtualisation sont VMWare, éditeur du logiciel éponyme, leader du marché, Microsoft avec Virtual PC et Virtual Server (ces deux produits sont spécifiques à l'environnement de Microsoft), Avanquest qui édite Parallels (dédié MacOS X sur Intel), le seul logiciel du marché qui permet une accélération 3D sur système invité, Citrix qui a acquis Xen, SWSoft, éditeur de Virtuozzo et Innotek GMBH avec VirtualBox.

# 1.3 Comparaison des différentes techniques de virtualisation

Très peu de systèmes démarrent directement, et sur plateforme PC tous les systèmes démarrent selon un processus (boot) en trois temps : le BIOS, le chargeur de démarrage (bootloader) et le système. Pour les systèmes embarqués et les anciens calculateurs, ces phases étaient aussi un choix à prendre en compte. Des moniteurs, OS minimaux emportant des fonctionnalités de débug, ont vu le jour. L'idée initiale était de se concentrer sur cette phase, puis ensuite d'en généraliser les concepts : un système d'exploitation reposant sur l'API d'un noyau réduit.

Les systèmes de virtualisation partent donc du principe de l'utilisation de couches logicielles intermédiaires. Afin d'avoir une idée théorique des performances des applications au sommet, il faut comparer verticalement l'empilage de couches. Il faut garder à l'esprit qu'il est possible d'élargir les schémas en rajoutant des environnements virtualisés consommant également des ressources de l'hôte, en mémoire puis en disque.

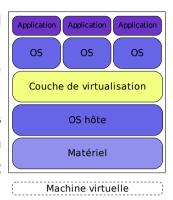
Fondamentalement, on trouvera deux idées principales: isolation par empilement ou par juxtaposition, dont les variations, dans des buts d'optimisation, donneront principalement quatre types de technologies:





#### 1.3.1 Machine Virtuelle

Une machine virtuelle est un logiciel qui tourne sur l'OS hôte, ce logiciel permettant de lancer un ou plusieurs OS invités, c'est l'archétype de la solution de virtualisation par empilement de systèmes. La machine virtualise le matériel (ce qui passe généralement par un émulation partielle) pour les systèmes d'exploitation invités : les systèmes d'exploitation invités croient dialoguer directement avec le matériel. En pratique on a recours à une émulation logicielle des périphériques, et parfois aussi de tout ou partie de la machine.



Cette solution isole bien les systèmes d'exploitation invités, mais elle a un coût, en premier lieu en **performance**, dont les principales victimes seront les entrées-sorties. Ce coût, assez important déjà s'il suffit de protéger les instructions privilégiées, peut être très élevé si le processeur doit être intégralement émulé. Cette solution n'est pas non plus économe en mémoire, puisque aucune économie d'échelle ne peut être réalisée en ce qui concerne les OS (kernels) chargés.

#### Exemple:

- → Qemu : émulateur de plateformes x86, PPC, Sparc,
- → VMWare : propriétaire, émulateur de plateforme x86,
- → Microsoft VirtualPC et VirtualServer : propriétaire, émulateur de plateforme x86,
- → VirtualBox : émulateur de plateforme x86.

Le talon d'Achille de cette solution est donc l'importante consommation en performances. Certains projets minimisent l'impact de ce coût en ressources en permettant des courts-circuits optionnels (KVM, KQemu, VirtualBox), disponibles uniquement hors du contexte d'émulation bien sûr.

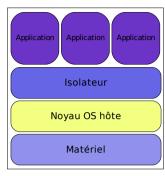
Les deux technologies suivantes constituent elles aussi le résultat de considérations d'optimisation de cette solution, le principe étant de diminuer au mieux l'épaisseur des couches de virtualisation tout en conservant le même degré d'isolation.





# 1.3.2 Virtualisation d'OS, Isolateur

Un isolateur est un logiciel permettant d'isoler l'exécution des applications dans des contextes ou zones d'exécution, c'est l'archétype de la solution de virtualisation par "juxtaposition". L'isolateur permet ainsi de faire tourner plusieurs fois la même application (à base d'un ou plusieurs logiciels) prévue pour ne tourner qu'à une seule instance par machine.



Notons que cette technologie consiste en quelque sorte à généraliser la notion de "contexte" Unix : ce dernier isole les processus (mémoire,

Virtualisation d'OS

accès aux ressources), on ajoute alors : une isolation des périphériques (c'est le rôle de l'isolateur), voire leur partage, les systèmes de fichiers donc les fichiers eux-mêmes et leurs accès.

Cette solution est très performante, du fait du peu d'overhead (chute de performance conséquente de l'ajout des couches de virtualisation), mais les environnements virtualisés ne sont pas complètement isolés, ils partagent en particulier le code du noyau. Cette solution est aussi remarquablement économique en mémoire par conséquence de la dernière remarque. Ces environnements sont donc bien adaptés au déploiement de nombreux serveurs virtuels de test ou développement basés sur un même système.

## Quelques isolateurs:

- → Linux-VServer: isolation des processus en user-space,
- → BSD Jail: isolation en user-space,
- → OpenVZ : libre, partitionnement au niveau noyau sous Linux et Windows 2003. C'est la version open-source du logiciel Virtuozzo.

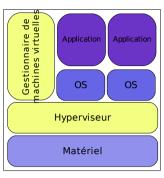
Les isolateurs tendent à isoler à un niveau de plus en plus proche du système, voire dans le système dans le cas de OpenVZ, qui peut être vu comme un Linux avec plusieurs tables de processus, chacune dans le contexte d'une distribution propre.





# 1.3.3 Hyperviseur complet

Partant du principe, exposé précédemment, qu'une approche pour une virtualisation efficace consiste à affiner les couches, une première approche consiste à proposer un novau léger (de type micro-novau par exemple), lequel est accompagné d'outils de supervision, et adapté pour faire tourner des systèmes d'exploitation natifs. Pour réussir cette approche, soit on émule le matériel (et on revient aux performances de la machine virtuelle pour les I/O), soit on dispose des instructions dédiées à la virtualisation. Dans ce dernier cas, les logiciels libres (Hyperviseur complet



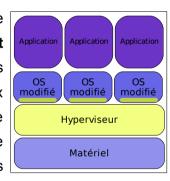
concernés se limiteront au monde x86 ou x86-64, munis des instructions ad-hoc.

Les principaux exemples de ce principe sont :

- → KVM : Intégré au novau GNU/Linux à partir de la version 2.6.20.
- → Xen sur une machine offrant le support des instructions AM-V (version 3.03 ou ultérieure) ou VTx (version 3.0 ou ultérieure).

#### 1.3.4 Paravirtualiseur

Un paravirtualiseur est un noyau hôte allégé et optimisé pour ne faire tourner que des noyaux de systèmes d'exploitation invités, adaptés et optimisés. Les applications en espace utilisateur des systèmes d'exploitation invités tournent ainsi sur une pile de deux noyaux optimisés, les systèmes d'exploitation invités ayant conscience d'être virtualisés. Cette approche offre l'avantage d'être utilisable en l'absence des instructions spécifiques, mais elle est impraticable pour des systèmes non libres pour lesquels l'éditeur ne fera pas l'effort Paravirtualisation d'adaptation.



#### Exemple:

Xen: noyau léger supportant des noyaux Linux, Plan9, NetBSD, etc,



Page: 10 / 61



# 2. Le besoin

#### 2.1 Contexte

La virtualisation est une approche système dans la continuité logique de l'évolution : après avoir "virtualisé le processeur" c'est-à-dire fait du temps partagé, pour faire du multitâche sur une unité de traitement unique, on virtualise le système entier pour disposer de plusieurs OS complets tournant simultanément sur une même machine. Cette approche, historiquement introduite sur les gros systèmes (IBM), se démocratise en raison de la puissance très importante des microprocesseurs aujourd'hui, laquelle est largement sous-exploitée, on estime l'utilisation de la puissance des serveurs à environ 10% aujourd'hui<sup>4</sup>.

Cette technologie a commencé sous la forme de projets d'amateurs (Boochs, émulateur de processeur x86 qui a peu à peu émulé toutes les phases de cette famille de processeurs), en parallèle d'une solution commerciale, **VMWare**. Par la suite, un projet Open Source a changé la donne. S'appuyant sur une partie du code de Boochs en améliorant et affinant ses concepts, Fabrice Bellard a démarré **Qemu**, apportant un compilateur just-in-time, puis une accélération en permettant des 'raccourcis' maîtrisés contrôlés par un fichier spécial (un périphérique Unix) dédié.

Dans le même temps, les laboratoires d'informatique de Cambridge lançaient un moniteur évolué permettant la cohabitation de plusieurs OS, **Xen**. Plusieurs projets Open Source de bonne facture existent aujourd'hui dans cette technologie. Signe concret de cette évolution, les deux fondeurs de microprocesseurs de PC ont intégré dernièrement des instructions destinés à faciliter et accélérer la virtualisation (VT pour Intel et AM-V chez AMD).

Signe de l'adoption large des technologies de virtualisation, RedHat a lancé le projet **Libvirt** (www.libvirt.org), dont le but est l'écriture d'une API commune à plusieurs solutions de virtualisation, base au développement d'outils graphiques ou d'administration génériques. La première mouture de libvirt visait Xen, la version à jour supporte aussi Qemu et OpenVZ, candidats étudiés ici.

Les gains de l'utilisation de la virtualisation sont matériels (une machine peut simuler plusieurs serveurs totalement cloisonnés, utile pour proposer par exemple une plate-forme de tests ou pour déployer de nouveaux services sur un parc préexistant sans se soucier de migrations), ou en sûreté de fonctionnement (un serveur sur une machine virtuelle peut prendre le relais en configuration cluster si le maître cède pour des raisons logicielles), ou les deux. Cette approche permet par exemple de configurer une ferme de serveurs de secours proposant plusieurs services

<sup>&</sup>lt;sup>4</sup>http://linuxvirtualization.com/pages/why-virtualize



Page: 11 / 61





différents sur une machine de secours unique sans contraindre tous les services aux mêmes versions logicielles, même environnement système, même configuration.

L'arrivée des technologies de virtualisation dans le monde des PC a donc ouvert de nouvelles voies pour <u>sécuriser</u>, <u>simplifier</u> et <u>rationaliser les coûts</u> dans les systèmes d'information.

# 2.1.1 Outils en place

La référence commerciale en la matière est **VMWare**, qui propose une gamme de solutions de virtualisation matures, chaque solution se trouvant accompagnée d'outils facilitant la gestion du parc virtuel. VMWare est disponible gratuitement pour les machines disposant d'au plus 2 unités de traitement, mais est payant au-delà. C'est un logiciel commercial distribué sous licence, aucune partie du code n'est libre. Il fait de surcroît l'objet d'un brevet aux Etats-Unis.

Le logiciel libre phare du domaine, **Qemu**, est aussi une base de référence, d'autant plus que plusieurs produits (dont **VirtualBox**) en utilisent des parties. Par contre, utilisé « sec », Qemu ne propose qu'une interface en ligne de commande qui, bien qu'efficace, ne manquera pas de paraître un peu abrupte. Il propose cependant beaucoup de fonctionnalités originales, ce qui le rend incontournable : il est le premier logiciel présenté ci-après.

Ces deux références sont des implantations selon la technologie des machines virtuelles, VMWare ayant aussi à son catalogue un produit qui s'apparente plus à un hyperviseur complet. Le gros avantage des systèmes libres d'aujourd'hui, c'est qu'ils couvrent presque toutes les techniques de virtualisation identifiées, avec donc Qemu et VirtualBox pour les machines virtuelles, Xen pour la paravirtualisation, OpenVZ pour la virtualisation au niveau OS, et Xen 3.0 ou kvm, utilisés sur des machines les supportant, pour l'hyperviseur complet.

#### 2.1.2 Orientation

Les technologies de virtualisation sur plate-forme x86 ont bénéficié d'une maturation rapide en raison du grand intérêt que la technologie présente et du sérieux des acteurs impliqués. Il en est de même pour les solutions Open Source. Ces dernières ont de plus l'avantage de couvrir la totalité des technologies existantes.

Bien que techniquement mature et de qualité en ce qui concerne les solutions libres, cette technologie est encore jeune. Si, en effet, la qualité technique de ces produits en font des éléments parfaitement capables d'assumer des utilisation en production, ce sont les outils d'administration qui n'ont pas encore tout-à-fait atteint le niveau de leurs équivalents propriétaires. Le jugement sera donc orienté sur un usage pour lequel cette déficience passagère ne présente pas de difficulté, c'est-à-dire en développement et en test de plate-forme, la mise en production



Page: 12 / 61



étant techniquement parfaitement viable, mais potentiellement complexe pour le court terme. Nous indiquerons évidemment les candidats techniquement adaptés à un passage en production en précisant leurs points fort dans ce contexte et en relevant les points qu'il peut leur rester à améliorer pour cela.

#### 2.1.3 Besoins identifiés

La virtualisation est une technologie dont l'intérêt naît de la constatation suivante : les serveurs sont globalement sous-utilisés (de l'ordre de 10% de charge moyenne comme cela a été cité en introduction). S'il est rationnel de voir la virtualisation sous l'angle de l'économie de matériel conséquente, il est bon de penser aux besoins réels souhaités, afin de choisir la solution adaptée. On retient couramment les utilisations suivantes :

- optimisation de l'utilisation des serveurs,
- établissement d'un plan de reprise d'activité en cas de catastrophe sur le site principal,
- banalisation de la plateforme afin de faciliter déploiement, support et pérennité.

Ces besoins stratégiques se voient dans une première approche pragmatique en terme de besoins de :

- ◆ faire cohabiter plusieurs OS différents, ces solutions de virtualisation peuvent techniquement permettre l'utilisation de plusieurs OS sur une même machine sans la redémarrer (cependant certains accords de licence, par exemple chez Microsoft pour Vista et XP, imposent l'achat d'une licence spécifique pour un usage en machine virtuelle).
- déployer plusieurs instances cloisonnées d'un même système, de configurations différentes,
- proposer rapidement un serveur de développement ou de test sans besoin de fourniture matérielle,
- consolider une architecture de production.

Outre la sensibilité naturelle à la performance des systèmes étudiés et à leur stabilité, on prêtera donc attention, en raison de besoins liés à des tâches d'administration système, à la disponibilité d'outils permettant une prise en main aisée de ces outils. Les critères de choix des systèmes de virtualisation vont donc se décliner ainsi :

 qualité des outils d'administration disponibles, car si la virtualisation est une réponse économique à une sous-utilisation matérielle globale, l'administration des systèmes est un poste aussi très important dans le coût des systèmes et datacenter, et il convient de prendre en compte le coût total de possession incluant le coût d'exploitation,

> LIN AGORA GROUPE

Page: 13 / 61



- diversité des plateformes invitées possibles, ce qui peut permettre de récupérer des applications lors de transitions de systèmes ou à l'occasion de migrations, ce qui a une influence sur le niveau souhaité de virtualisation.
- performance, de l'ordre du confort sur un poste de développement, nécessaire sur un serveur de test, la performance est une donnée critique en production. Selon les approches techniques et l'utilisation, le coût en CPU des couches de virtualisation peut varier de quelques pourcents à plus de 60% dans le cas d'une émulation totale,
- densité, surtout dans une logique de serveur de reprise d'activité, on veut pouvoir tirer le meilleur d'un nombre réduit de serveurs sur un site distant pour laisser passer l'incident,
- ◆ facilité de déploiement, l'intérêt numéro 1 de la virtualisation étant qu'une nouvelle installation ne demande qu'une simple copie de fichier,
- facilité de migration, pour pouvoir passer un service d'une machine à une machine plus puissante sans l'interrompre ou plus exactement avec une interruption minimale.

# 2.2 L'état de l'art

Historiquement, la virtualisation a commencé sur des gros systèmes avec un matériel conçu dans ce contexte.

Une fois que les processeurs sont devenus suffisamment puissants pour rendre des couches d'abstraction matérielle utilisables, la virtualisation a gagné les systèmes personnels, lesquels ont un matériel non prévu pour faire tourner plusieurs systèmes, d'où l'orientation technologique des premières solutions (émulation, machine virtuelle complète).

Aujourd'hui, plusieurs solutions de virtualisation existent, plusieurs approches technologiques différentes cohabitent, et les processeurs, à l'image du matériel de mainframes d'alors, ont adopté des instructions dédiées. Nous faisons ici un point sur l'état des divers logiciels de virtualisation.

#### 2.2.1 Fonctionnalités

Le strict minimum attendu d'un système de virtualisation, c'est qu'il permette d'exécuter deux instances d'un logiciel prévu pour fonctionner en instance unique, tout en séparant complètement ces deux instances. C'est l'approche d'un concept comme une prison 'chroot' pour les systèmes Unix, complété par une translation de ports. L'inconvénient principal de cette approche, c'est qu'il reste des éléments d'exécution communs, comme la libc. A ce niveau, la principale fonctionnalité attendue, c'est le lancement séparé de deux instances d'un logiciel sans télescopage dus à des ressources qui deviendraient partagées (fichiers de configuration par exemple). Notons que l'on lance toujours les processus sur la même machine ce qui peut impliquer quelques soucis dans le

LINAGORA GROUPE PARIS - TOULOUSE - LYON

Page: 14 / 61



cas d'utilisation de ressources plus subtiles que le développeur n'avait pas pensé à protéger d'exécutions multiples, comme des segments de mémoire partagée...

L'étape suivante consiste à généraliser la séparation à d'autres périphériques et à faire le nécessaire pour accepter deux systèmes différents, surtout 2 libc différentes, ceci afin de se rendre moins vulnérables aux anomalies éventuelles d'une libc, ou être plus souple dans le choix de ses éléments systèmes. Nous approchons là l'approche d'un BSD Jail ou d'un OpenVZ. Il reste encore une contrainte : le système bas niveau (kernel) est le même pour tout le monde, et ce dernier devient le talon d'Achille, mais on commence réellement à voir deux machines différentes au niveau logique.

L'approche suivante consiste à utiliser un logiciel d'émulation pour émuler une machine complète sur laquelle on installe un nouveau système. Cette approche permet l'utilisation simultanée de systèmes différents, au détriment des performances. C'est l'approche Qemu, VMWare ou VirtualBox. On peut cependant fortement atténuer la perte de performances due à l'empilement de plusieurs systèmes en se permettant des raccourcis surveillés. Là, la séparation est totale, les périphériques étant vus sous la forme d'une émulation. Seuls les périphériques émulés sont alors utilisable, ce qui est le cas des périphériques destinés aux serveurs mais moins des périphériques très spécifiques (cartes d'accélération 3D par exemple) qui proposeront souvent que des fonctionnalités limitées dans la machine virtuelle.

Concernant les fonctionnalités plus orientées gestion de parc, on mettra en avant les possibilités de :

- → migration en fonctionnement d'une VM d'une machine à une autre,
- → démarrage/arrêt/debug,
- → aides à la création, installation, sauvegarde, destruction de VM

# 2.2.2 Respect des standards

Nous avons sélectionné des logiciels émulant, de plus ou moins près, la plateforme PC IA-32/x86. Cependant, les derniers nés de cette famille de processeurs ont bénéficié d'ajouts de leur langage machine explicitement orientés virtualisation (AMD AM-V ou Intel VT). Si l'utilisation de ces instructions permet de simplifier l'écriture de programmes de virtualisation et d'améliorer fortement leur performance, elle contraint toutefois techniquement la plateforme de déploiement. Aussi les solutions utilisant explicitement ces instructions seront signalées, ainsi que celles les utilisant tout en leur gardant un caractère optionnel.

De plus, certains virtualiseurs nécessitent la modification du système invité. Si cela est facile pour un système open source, une telle modification n'est pas envisageable pour un système propriétaire. Ces contraintes seront signalées.

LIN AGORA GROUPE

Page: 15 / 61





Enfin, il reste la question du format des images (de disque, de machine etc.). Certains virtualiseurs montent des fichiers que l'on peut créer en loopback, d'autres demandent l'utilisation d'outils dédiés pour gérer un fichier optimisé (en performance ou en place : par exemple Qemu propose en option des disques utilisant le "copy on write"). Si les raisons qui poussent cette démarche sont parfaitement justifiées, la difficulté est après de porter les images de machine d'un virtualiseur à l'autre. Pour cette raison, l'option permettant, tel un *'chroot'*, de monter une sous-arborescence (qui pourra au passage être un montage d'un fichier par loopback), ou l'utilisation d'images de systèmes de fichiers loopback, est un plus appréciable.

Un mot maintenant concernant l'installation de systèmes : la plupart des virtualiseurs demandent une base pour démarrer, seuls Qemu et dérivés (dont VirtualBox, qui lui emprunte le code de boot) sont capables d'installer un PC virtuel avec des disques ou images de disques d'installation. Cette contrainte n'est pas gênante lorsqu'une image est disponible, mais cela complique fortement l'installation de systèmes "exotiques", tout comme l'installation à partir de disques amorcables.

#### 2.2.3 Scénarios d'utilisation

Les scénarios suivants d'utilisation sont communs à toutes les technologies de virtualisation. Selon les technologies et solutions, certains scénarios seront plus ou moins attrayants.

#### → Sécurité :

La virtualisation permet d'isoler chaque service réseau (comme Apache, le serveur mail, le serveur DNS etc.) dans un environnement virtuel séparé. Dans ces conditions, si un intrus trouvait une faille de sécurité dans une des applications, il ne pourrait maltraiter que ce service même ; puisque tous les autres services sont dans des VE (environnements virtuels) séparés, il ne pourrait pas y accéder.

#### → Consolidation de serveur :

Actuellement, la plupart des serveurs sont peu employés, de 10 à 20% selon les experts. En utilisant la virtualisation, de tels serveurs peuvent être consolidés en les migrant dans des environnements virtuels, un même serveur physique pouvant supporter de 2 à une dizaine de serveurs virtuels. Le gain est dans l'espace pris par les racks, les factures d'électricité, la mutualisation des équipements matériels de redondance et sûreté (RAID, climatisation), et les coûts de gestion.

#### → Plan de reprise d'activité :

Une autre utilisation de la virtualisation, c'est l'installation d'un système de secours situé dans un site distant. Ce système de secours étant destiné à embarquer tous les services à secourir à titre temporaire, il pourra donc être dimensionné moins généreusement que le système de production.



Page: 16/61



#### → Hébergement :

La virtualisation permet aux hébergeurs de proposer de l'hébergement virtualisé à bon marché. Chaque Environnement Virtuel (donc éventuellement, chaque client) a un accès root total, ce qui signifie que le propriétaire du VE peut réinstaller n'importe quoi, et employer des outils telles que les tables IP de Linux (règles de pare-feu).

#### → Développement et test, éducatif :

Les développeurs et testeurs ont souvent besoin d'accéder à un grand nombre de distributions Linux ; parfois ils doivent réinstaller ceux-ci à partir de zéro. Si toutefois les configurations sont d'une diversité maîtrisée, il est possible de disposer d'images virtuelles de chacune des installations prévues, et démarrer une installation se fait dans un temps de l'ordre de la durée de la copie d'un (gros) fichier.

# 2.2.4 Ergonomie

Un des buts de la virtualisation est l'économie, par une utilisation rationnelle (si possible optimale) des ressources matérielles. Cependant, si cette économie matérielle se fait au prix d'un surcoût trop important d'administration, de formation et par des concepts trop peu clairs, le jeu risque de ne pas en valoir la chandelle.

Le minimum exigé d'un outil de virtualisation, c'est la création, l'installation et la copie rapide et simples de machines virtuelles à l'arrêt, avec un recours minimum aux modes d'emplois, aides en lignes etc. Certaines solutions proposent de plus des facilités de surveillance des machines virtuelles, présentant sur un tableau synthétique ce que présenterait en fait un bon outil de surveillance de parc.

Les outils les plus aboutis en terme d'ergonomie permettent la création de machines virtuelles en quelques clics. Si c'est un plus appréciable aux utilisateurs occasionnels, il est bon de garder la possibilité de scripter la création de machines, si l'on veut en créer un grand nombre de manière automatique.

#### 2.2.5 Sécurité

De par leurs approches différentes, toutes les solutions proposent un compromis efficacité/sécurité différent, tout dépend du niveau d'isolation obtenu. Cependant, toutes les méthodes proposées ici assurent une séparation totale au niveau logiciel : une faille exploitée sur une machine virtuelle ne compromettra au pire que les autres services de cette machine virtuelle<sup>5</sup>.

LIN AGORA
GROUPE
PARIS - TOULOUSE - LYON

Page: 17/61

<sup>&</sup>lt;sup>5</sup>à moins d'une faille au sein même du kernel dans le cas particulier d'OpenVZ, puisque par conception de la virtualisation niveau OS toutes les VM partagent un même kernel. Mais de telles failles sont si rares, et la communauté du kernel si active, que ce danger peut être négligé.



# 2.2.6 Restrictions techniques

La virtualisation est aujourd'hui concentrée sur l'aspect serveurs et applications classiques, il reste des progrès à faire en terme de partage de matériel afin de faire bénéficier les environnements virtualisés de, par exemple, l'accélération 3D (utile pour l'utilisation de CAO/CFAO en environnement virtuel), ou d'autres accélérations matérielles (accélérateurs cryptographiques). Peut-être sur le modèle du partage de bus USB que certains logiciels font déjà? A noter, l'accélération 3D a déjà été réalisée dans le cadre d'un virtualiseur propriétaire sous MacOS, Parallels 3.0.



Page: 18 / 61



# 3. Les solutions

#### Candidats libres

On indique ici les logiciels de virtualisation fonctionnant sur architecture compatible x86 (ia32) ou x86-64, en mentionnant les éventuelles contraintes d'architecture. Les émulateurs destinés à fonctionner sur PowerPC ou autres architectures ont été écartés de cette étude.

Seuls les logiciels libres, c'est-à-dire ceux pour lesquels le code source est accessible, modifiable et le programme regénérable, en plus du libre déploiement, ont été retenus.

- Bochs
- Cooperative Linux
- FreeVPS
- ◆ Jail
- KVM (nécessite les extensions de virtualisation AMD ou Intel)
- Linux-VServer
- OpenVZ
- PearPC (émule un powerPC)
- Qemu
- ◆ View-OS
- User Mode Linux
- VirtualBox
- Xen
- Zones (licence libre Solaris)

# 3.1 Méthode de sélection

Les projets sont sélectionnés sur des critères de maturité du projet, utilisabilité dans l'environnement souhaité (Linux et Windows), dynamisme de la communauté, ergonomie. On écarte donc les solutions obsolètes, en forte perte de vitesse, ou cantonnés à l'émulation de systèmes anciens.

#### 3.1.1 Présélection

Sur les critères ci-dessus, on filtre donc les solutions suivantes, non retenues :

◆ Bochs : ce programme était au départ un émulateur de processeur type ia-32 (x86). S'il ne propose pas une émulation complète et se repose sur le système pour permettre le déploiement de plusieurs systèmes, Bochs est aujourd'hui dans une large partie réutilisé



Page: 19 / 61



par les projets récent, tout ou en partie. Cependant, Bochs n'est pas une solution d'avenir industrielle, mais une bonne base de départ.

- Cooperative Linux : solution de virtualisation à base de multitâche coopératif, cette solution sert surtout à utiliser un environnement Linux depuis un environnement Windows. Comme toute solution de multitâche coopératif, si le processus part en boucle, tout se bloque, ce qui ne convient pas à un environnement de production.
- FreeVPS (Virtual Private Servers): ce logiciel a une approche pour le moins originale. Plutôt que de gérer pénurie et concurrence, chacun récupère son allocation de ressource au départ (disque, bande passante,...). L'inconvénient de cette approche, c'est que les ressources matérielles ne sont plus utilisées de façon optimales mais partagées a priori. Si une application a besoin de place, elle ne la récupèrera pas d'une autre qui, elle, « nage » dans son espace. Cette solution, essentiellement à base de chroot() et généralisant les autres ressources sous l'appellation de « contexte », propose un niveau d'isolation aujourd'hui trop faible pour être pertinent.
- VServer : isolation moindre que la solution précédente. Ces solutions se justifient dans des contextes où l'on cherche juste quelque chose d'un peu plus abouti qu'un simple chroot().
- User Mode Linux : Projet initialement orienté debug kernel, il n'a pas reçu l'aval de Linus Torvalds et son utilisation est aujourd'hui trop peu répandue dans l'industrie. Ce projet semble peu vivant aujourd'hui. Il visait à permettre à un kernel (invité) Linux d'être exécuté comme un processus normal du kernel hôte, mais après une apparition furtive dans les sources officiels du noyau, sa réputation d'instabilité a eu raison de lui.
- ViewOS: Projet ambitieux et dispersé, View-OS vise, en retouchant les concepts sousjacents d'un système Unix, à permettre aux utilisateurs des droits aujourd'hui interdits par le modèle Unix. Un des effets de bords de ce projet est UMView, où un système invité passe en fait des appels systèmes par l'intermédiaire d'un démon contrôleur. Ce projet n'a pas l'air d'avoir encore atteint une maturité critique.
- ◆ PearPC : Solution d'émulation de PowerPC destiné aux Intel x86, ce logiciel sort fortement du cadre proposé (virtualisation des x86) et a donc été laissé de côté. Cependant c'est une solution très décente d'émulation d'un Macintosh à base de PowerPC.
- Zones: solution de virtualisation, basée sur un isolateur s'appuyant sur un support d'instructions adaptées, Zones est dédié au système Solaris: ce n'est pas le contexte souhaité.

LIN AGORA GROUPE

Page: 20 / 61



# 3.1.2 Comparaison des solutions sélectionnées

Au final, les solutions actuellement en état convenable pour un serveur de production, indiquées à la suite, seront comparées de la manière suivante :

- → essai global : installation de systèmes divers, pour déceler certaines difficultés,
- → analyse des fonctionnalités, apports, qualités, défauts
- → analyse de performance

Chaque solution est testée dans un contexte contenant deux machines virtuelles invitées dont une tournant à vide.

# 3.2 Solutions retenues

Solutions permettant de virtualiser un invité différent de l'hôte, en particulier un système Windows et un système Linux :

- → Qemu
- → KVM
- → Xen
- → VirtualBox

Solution fournissant plusieurs environnements cloisonnés d'un seul et même système d'exploitation (Linux en l'occurrence) :

→ OpenVZ (version libre de Virtuozzo)

OpenVZ est certes un peu en marge, puisqu'il est un isolateur. Dans sa catégorie, il est le plus abouti, permettant en particulier l'installation et l'exécution de plusieurs Linux en parallèle pourvu qu'ils partagent le même noyau. Il est conservé dans ce rapport en raison de son exceptionnelle scalabilité, conséquente de son approche : en effet, c'est le seul qui peut prétendre proposer des dizaines<sup>6</sup> de machines virtuelles.

Malheureusement pour la version Open Source, les **utilitaires** facilitant la migration d'un système réel vers un système virtuel sont dédiés à la **version commerciale**, Virtuozzo.

LIN AGORA GROUPE

Page: 21 / 61

<sup>&</sup>lt;sup>6</sup>le site avance une centaine avec seulement 768Mo de RAM



#### 3.2.1 Qemu

Application Application
OS OS OS F

Couche de virtualisation
OS hôte

Matériel

Machine virtuelle

Projet développé par Fabrice Bellard, autour de qui une petite communauté d'experts actifs s'est formée, Qemu est l'archétype d'une solution de type machine virtuelle complète.

Qemu ne nécessite pas de processeur spécifique, n'impose pas de modifier le système invité et marche sur un système hôte non modifié. Cependant, kQemu, approche d'accélération dédiée, permet d'accélérer

notablement les performances (toujours sans contrainte sur la génération de processeur).

#### Présentation

QEMU est une solution de virtualisation sur le modèle de la machine virtuelle, techniquement très aboutie, permettant depuis la virtualisation de machine cible identique à l'hôte, avec une virtualisation potentiellement accélérée (par le module noyau kQemu, récemment reversé à l'Open Source) jusqu'à l'émulation complète de machine, même d'architecture processeur différente (ARM, MIPS par exemple).

Qemu ne requiert pas une plateforme spécifique, n'utilisant pas de code spécifique d'un processeur particulier. De plus, parmi les exemples choisis, Qemu est la seule solution qui, avec une intervention totalement logicielle en mode utilisateur (sans patch ni module kernel), puisse être néanmoins fonctionnelle. Cependant l'utilisation d'un module dédié permet un gain notable de performances.

#### **Fonctionnalités**

En solution de virtualisation, Qemu propose, sur la base d'une émulation complète (optionnellement accélérée par un module kernel), une approche séduisante de la virtualisation native, et par un émulateur écrit sur le modèle d'un compilateur just-in-time, un outil appréciable pour faciliter le développement sur plates-formes ésotériques comme dans l'embarqué, par exemple. Il permet l'installation de systèmes invités différents de l'hôte, par exemple un Windows sur un hôte Linux.

Ce programme est très évolué techniquement, sur la piste d'une maturation rapide, et se trouve nanti de deux possibilités d'accélération différentes, l'une spécifique nommée kQemu, l'autre, Kvm, requérant un patch à QEMU. Cette partie testera Qemu seul puis accompagné de sa propre accélération kQemu, le module KVM faisant l'objet d'une section dédiée.



Page: 22 / 61



Parmi les fonctionnalités intéressantes qu'il propose, on note une émulation de disque selon une technologie copy-on-write (optionnel car non disponible pour tous les systèmes de fichiers, en particulier incompatible avec les FAT), ce qui permet à un disque émulé de n'occuper que la place effectivement prise et non toute la place allouée. Qemu supporte ainsi une grande variété de formats d'images disques.

En terme de support matériel, Qemu émule une carte graphique Cirrus Logic (SVGA) sans aucune accélération 3D, ce qui n'est pas gênant pour un serveur mais doit être noté, ainsi qu'une carte Ethernet de type NE2000. Cette carte est globalement la plus répandue, ce qui assure son support par l'OS invité.

#### **Evaluation**

#### Installation

Qemu se présente sous la forme de .tar.gz source et binaire, des distributions le packagent pour leur système de paquets (RedHat, ubuntu en particulier), et s'installe sans difficulté, il faut cependant noter qu'en cas de recompilation, il nécessite un gcc version 3.4.x (le module compatgcc des distributions récentes). L'installation par défaut (rpm ou sources) génère les Qemu pour à peu près toutes les plateformes supportées.

Qemu est une solution d'émulation par recompilation sur un modèle "just-in-time" utilisant déjà un bon nombre d'accélérations dans les cas sûrs (plate-forme native, instructions non dangereuses). Un module proposant une accélération supplémentaire (kQemu) est disponible et se compile et s'installe sans souci.

#### Scénarios

#### → Sécurité

Qemu étant à la base une solution d'émulation, la sécurisation du reste du serveur par rapport aux failles de la machine virtuelle est complètement assurée, à l'exception de l'effet d'éventuelles anomalies dans le module d'accélération noyau si on l'utilise.

#### → Consolidation de serveur

En consolidation de serveurs, Qemu est utilisable mais peut se montrer gourmand et un peu lent, en particulier dans une version non accélérée.

#### → Plan de reprise d'activité

Qemu est d'un déploiement aisé, que l'on dispose ou non des images prêtes à l'emploi, puisque l'installation d'un Linux à partir des supports se fait sans souci. On peut donc l'envisager en secours, mais sa gourmandise en RAM (inhérente au choix technologique) est à ne pas oublier.



Page: 23 / 61



#### → Hébergement

En terme d'hébergement, Qemu n'est probablement pas la solution la plus scalable à cause de la consommation mémoire (pour n systèmes invités, nous avons cohabitation de n+1 noyaux différents) mais son utilisation est néanmoins tout à fait pertinente dans certains contextes.

#### → Développement et test, éducatif

En ce qui concerne les considérations de test et développement, c'est probablement le domaine dans lequel QEMU excelle, étant la seule solution à proposer une émulation complète d'architectures même non x86, et permettant son utilisation (sans accélération) depuis un compte non privilégié. Qemu peut aussi s'utiliser pour déboguer le noyau (mais le cheminement temps réel n'étant pas identique à une plate-forme réelle, certaines anomalies très liés à des chronologies d'accès ne pourront être reproduites).

En terme de performances subjectives, Qemu sans accélération se montre un peu poussif, mais reste utilisable. Sa technologie particulière (compilateur just-in-time en émulation) permet même d'émuler une machine spécifique de manière utilisable. Le module spécifique kQemu lui offre une accélération conséquente.

#### Analyse de performances

Sans accélération, Qemu est lent et charge fortement le système hôte, même si configuré en monoprocesseur (donc complètement monotâche) il laisse un cœur d'un core2 duo au reste du système. L'accélération dédiée kQemu permet un gain de l'ordre d'un rapport 5, rapprochant les performances de celle de l'hôte (hors I/O et accès mémoire).

# Retours d'expérience

Tout d'abord, notons que VirtualBox et Kvm reposent sur Qemu : le premier se replie sur son code pour les cas complexes (boot mode réel et quelques cas sporadiques), le second l'utilise patché, cela souligne l'aboutissement de ce logiciel, dont le numéro de version n'indique toutefois pas une version finale (version actuelle : 0.9.0).

Avec une image de disque préchargée, nous avons testé le boot en émulation d'architecture ARM : la mise en route d'une émulation de plate forme a été très simple. A cet égard, Qemu est un outil d'une redoutable efficacité. Des images exemples de boot Linux existent aussi, mais générer son image propre n'est pas difficile, il n'a que l'inconvénient de la ligne de commande.

Thalès Optronique utilise Qemu en environnement R&D, en profitant de la simplicité avec laquelle on obtient des émulations de plates-formes d'ailleurs.

LIN AGORA GROUPE

Page: 24 / 61



Linagora a utilisé Qemu en production, surpassé par VMWare Serveur, en raison de la qualité des outils d'administration : c'est effectivement le point faible de cette solution. Des outils graphiques d'administration sont cependant en développement chez Red Hat Software par exemple, ces outils ayant la particularité intéressante de vouloir couvrir plusieurs solutions de virtualisation par une seule interface.

S'il souffre encore un peu de sa réputation de produit « artisanal » en raison de son origine non-industrielle, **sa qualité parle pour lui** et son utilisation totale ou partielle par d'autres virtualiseurs, y compris industriels, est la meilleure preuve.

#### Liens

Gestionnaire de achines virtuelles

• Page de Qemu : <a href="http://fabrice.bellard.free.fr/Qemu/">http://fabrice.bellard.free.fr/Qemu/</a>

Application

OS

Application

OS

Hyperviseur

Matériel

Hyperviseur complet

Wikipedia : <a href="http://fr.wikipedia.org/wiki/QEMU">http://fr.wikipedia.org/wiki/QEMU</a>

3.2.2 KVM

KVM est en fait vu comme un périphérique (/dev/kvm) dédié à la virtualisation, utilisant les instructions de virtualisation des derniers AMD et Intel. Il n'est donc disponible que sur les plates-formes proposant ces extensions. Technologiquement, il s'apparente à un hyperviseur complet, le périphérique "/dev/kvm" s'occupant de la gestion de la pagination des systèmes hôtes, et le noyau Linux

accompagné d'utilitaires système jouant le rôle de gestionnaire de machines virtuelles. Il s'appuie largement, pour la partie hors instructions spécifiques, sur Qemu : le virtualiseur réel associé à Kvm est en réalité une version patchée de Qemu.

#### **Présentation**

KVM (Kernel-based Virtual Machine) est un module du kernel définissant un périphérique (/dev/kvm) dédié à la virtualisation. Il a été choisi pour intégrer la version 2.6.20 du noyau Linux.

Ce module, accompagné d'une version patchée de Qemu, propose une solution de virtualisation qui fonctionne avec des systèmes d'exploitation invités non modifiés. Cependant, il impose d'utiliser des processeurs x86 récents, équipés d'extensions pour faciliter la virtualisation : VT chez Intel et AM-V chez AMD pour les architecture X86.

Le module KVM a l'intérêt d'affecter très peu le noyau, son intégration a demandé peu de modifications, ce qui réduit la complexité et les risques d'instabilité.



Page: 25 / 61



Il propose un périphérique dédié dont l'interface, au fichier spécial nommé /dev/kvm, sert à être utilisé par un programme utilisateur (hors kernel), afin de mettre en place l'espace d'adressage, d'IO et l'affichage de la machine virtuelle dans l'espace de l'hôte. Le seul programme qui s'en sert aujourd'hui est une version modifiée de QEMU (dont l'exécutable s'appelle d'ailleurs Kvm sous Ubuntu). Ces modifications sont destinées à être intégrées au projet Qemu.

#### **Fonctionnalités**

KVM est une infrastructure de support de la virtualisation au niveau du noyau Linux proposant, par l'utilisation des instructions de virtualisation des processeurs Intel (extensions Vanderpool, flag vmx) et AMD (extensions Pacifica), la virtualisation complète (de machines Linux) et un support partiel de paravirtualisation (pour des systèmes hôtes différents). En pratique, il permet d'installer, en système invité, les systèmes de type Windows. Cependant ces systèmes devront être installés dans une configuration PC spécifique « PC standard » après avoir désactivé l'IO APIC.

Comme indiqué, cette extension au kernel (actuellement sous forme de module) nécessite le support des extensions ad hoc au langage machine x86, donc l'un des processeurs suivants :

- → Chez Intel: certains Pentium 4 6x1 et 6x2, Pentium D 9x0, Xeon 3xxx/5xxx/7xxx, Core Duo (sauf T2300E) et Core 2 Duo (sauf T5200, T5500, E4x00).
- → Chez AMD : tous les K8 dont la révision (stepping) est F et ultérieure. En fait les processeurs sur socket post-939 (AM2 pour les Athlons-64).

La virtualisation elle-même s'appuie sur Qemu, on retrouve les fonctionnalités de ce dernier :

- → Images de disques selon une technologie copy-on-write (non activée par défaut),
- → Import des images de machines virtuelles VMWare .vmx,

Par l'intermédiaire de la console de Qemu, Kvm propose la migration à chaud de machine virtuelle : on peut ainsi passer un service d'une machine à une autre plus puissante, ou équipée de plus de disque, sans l'arrêter. Cette fonctionnalité impose cependant quelques contraintes (plus de changement mode réel/mode utilisateur en cours, et quelques contraintes sur l'homogénéité du parc, la migration étant délicate si le serveur d'origine supporte le bit NX mais pas le serveur destination).

LIN AGORA GROUPE

Page: 26 / 61



#### **Evaluation**

#### → Installation

KVM propose une formalisation de l'utilisation des instructions de virtualisation des processeurs récents, mais n'est pas un logiciel de virtualisation. Pour utiliser les services qu'il fournit, on n'a à disposition aujourd'hui qu'une version modifiée de Qemu.

KVM se pose plutôt en alternative à kQemu, la grosse différence entre ces deux projets, c'est que Kvm a été choisi pour intégrer les sources officielles du noyau Linux (le noyau "Vanilla" de kernel.org), ce qui lui donne une assise intéressante en terme de qualité de support et de vivacité de la communauté. Il a aussi beaucoup gagné en visibilité.

Son autre grosse différence par rapport à kQemu, c'est l'utilisation des instructions de virtualisation ; cela limite les plateformes intéressées aux plateformes récentes, mais celles-ci se généralisent. Ces approches différant, il est intéressant d'en comparer les performances.

Installer KVM à partir d'un noyau 2.6.20 ou plus récent consiste à sélectionner l'option Kvm et le support spécifique adapté (Intel ou AMD), puis de recompiler le kernel. Le module KVM et la version modifiée de Qemu qui l'accompagne font l'objet d'un paquetage rpm commun.

A noter à l'utilisation, un bug connu concernant la gestion de l'IO APIC impose l'utilisation de l'option boot du kernel "NO\_APIC" ou, moins restrictif, "NO\_L\_APIC".

#### → Scénarios

KVM ne propose "que" la possibilité d'accéder aux possibilités de virtualisation des processeurs récents, et utilise une version modifiée de Qemu. Les résultats ici seront proches de ceux de Qemu.

#### Sécurité

Qemu étant à la base une solution d'émulation, la sécurisation du reste du serveur par rapport aux failles de la VM est complètement assurée<sup>7</sup>.

#### Consolidation de serveur

En consolidation de serveurs, Kvm est utilisable mais peut se montrer gourmand en mémoire. Les performances sont par contre excellentes, proches d'un système natif.

• Plan de reprise d'activité

<sup>&</sup>lt;sup>7</sup>à l'exception de l'effet d'éventuelles anomalies dans le module d'accélération noyau si on l'utilise



Page: 27 / 61





Le déploiement reste tout aussi aisé, que l'on dispose ou non des images prêtes à l'emploi, puisque l'installation d'un Linux à partir des supports se fait sans souci. Sa gourmandise en RAM (inhérente au choix technologiques) est son talon d'Achille, sa performance est un avantage.

#### ◆ Hébergement

En terme d'hébergement, Qemu/Kvm souffre de la même consommation mémoire, par contre cette solution propose des performances en nette hausse par rapport à Qemu. On peut donc y voir une compensation si l'on peut se permettre l'investissement en RAM.

#### • Développement et test, éducatif

En ce qui concerne les considérations de test et développement, QEMU excelle dans ce domaine, KVM en hérite, mais l'émulation complète est dans ce cas hors-jeu.

Si l'intégration du support de Kvm à la ligne principale de développement du kernel Linux est récente, cette solution est néanmoins remarquablement stable et ne déprécie pas Qemu sur ce point, à l'exception de la limitation déjà citée d'IO APIC. Si, maintenant, l'accélération proposée par le module permet des performances proches de celles d'une machine seule, la scalabilité n'est pas la meilleure en raison de l'obligation d'embarquer autant de systèmes complets que de machines virtuelles, ce qui consomme une forte quantité de mémoire.

En terme de performances subjectives, Kvm est sensiblement plus rapide que kQemu, lui-même accélérant notablement Qemu. On est effectivement proche des performances natives.

#### Analyse de performance

Résolvant le problème des accès mémoire lents de Qemu par l'emploi du « Ring -1 », on gagne un rapport d'environ 8 par rapport à Qemu non accéléré, et un rapport de presque 2 par rapport à kQemu pour des applications utilisant intensivement la mémoire.

## Retours d'expérience

Peu de retour d'expérience aujourd'hui, puisqu'il a été intégré au kernel dans la version 2.6.20, cette version est très récente. Seules les distributions sorties récemment l'intègrent (Fedora 7), tout comme les distributions dynamiques (Zenwalk par exemple).

Ce qui ressort des tests, c'est que cette solution est plus rapide que Qemu (même accéléré par kQemu), mais présente les mêmes inconvénients ergonomiques et d'administration que ce logiciel. C'est cependant à notre avis la solution technique la plus prometteuse.

#### Liens

→ Site officiel de Kvm : <a href="http://kvm.gumranet.com/kvmwiki">http://kvm.gumranet.com/kvmwiki</a>



Page: 28 / 61



#### 3.2.3 VirtualBox

Application Application

OS OS OS In

Couche de virtualisation

OS hôte

Matériel

VirtualBox propose un support optionnel des instructions de virtualisation Intel VT et AMD AM-V, ce dernier support étant annoncé expérimental d'après la documentation de la version en cours (1.4.0). S'il en est de fait utilisable sur toutes les plates formes x86 32 et 64 bits, l'utilisation de ces instructions dédiées accélère notablement les systèmes invités.

Il permet une émulation de tout système PC, donc des systèmes d'exploitation invités de types Windows, moyennant une petite limitation

#### **Présentation**

développée plus loin.

VirtualBox est un produit qui se présente comme "seule solution professionnelle de virtualisation distribuée sous licence GPL". Il se présente accompagné d'utilitaires destinés à faciliter la création de machines virtuelles, disques et montages CD/DVD (fichiers .iso testés). Cependant, le site de Innotek est très clair sur les limites de la version Open Source, et l'USB ne s'y trouve que depuis peu, réservé auparavant à la version commerciale. La version Open Source est disponible packagée, il faut le noter, pour un nombre important de distributions (dont OpenSuse, Mandriva, Fedora, RHEL et Ubuntu).

Techniquement, VirtualBox est une solution de virtualisation efficace, potentiellement accélérée sur architecture équipée des extensions virtualisation x86, se repliant sur Qemu dans certains cas (mode réel c'est-à-dire au boot, puis quelques cas sporadiques).

#### **Fonctionnalités**

VirtualBox émule un PC complet, permettant une variété en théorie totale des systèmes d'exploitation invités. Une limite à savoir : le modèle de carte réseau émulé, dont le support a été supprimé de Windows Vista, impose à ce dernier un démarrage sans réseau et l'installation d'un driver afin d'être pleinement fonctionnel.

A part cela, ce logiciel, produit de la société allemande Innotek GmbH, propose une version Open Source (OSE, Open Source Edition) pleinement opérationnelle mais sans les interfaces graphiques les plus efficaces. A noter, le boot ia32 (qui passe par le vénérable mode réel x86) et quelques situations délicates font opportunément appel au code de Qemu.

A l'instar de ce dernier, VirtualBox propose de plus une accélération par l'intermédiaire d'un module kernel dédié. Cette accélération profite des instructions de virtualisation.



Page: 29 / 61



#### **Evaluation**

#### → Installation

Il est en théorie possible d'utiliser VirtualBox pour virtualiser tout système d'un PC, même si certains systèmes (FreeBSD) ont posé des problèmes par le passé. Notons l'aspect sécurisant de l'approche virtualisation : c'est le système invité qui plante, sans aucune conséquence fâcheuse sur le système hôte. Ce défaut a été corrigé depuis, la version 1.5.2 d'octobre dernier permet d'installer FreeBSD.

La seconde remarque concernant VirtualBox, c'est qu'il est relativement gourmand en RAM, bien plus qu'en CPU : démarrer simultanément deux machine virtuelle Linux équipées respectivement de 384Mo et 256Mo a nécessité 1Go de swap en plus du giga de RAM qui équipe la machine, sur une Fedora Core 6. Il faudra donc penser à dimensionner généreusement les machines. La consommation en RAM est déjà le talon d'Achille des virtualiseurs complets, mais VirtualBox se montre particulièrement gourmand.

En ce qui concerne son ergonomie, elle est fortement inspirée des aspects agréables des habitudes "à la Windows", avec un assistant de création de machines virtuelles utilisable avec un recours minimum aux aides en ligne. Démarrer, arrêter ou gérer plus généralement des VM est aussi d'une simplicité bien pensée. Cependant, cette ergonomie intéressante gagnerait à s'équiper aussi de fonctionnalités de contrôle à distance, qui manqueront aux administrateurs. Pour cela, le modèle reste WMWare dont les outils sont remarquablement bien pensés (pas de dépendance excédentaire sur la machine serveur par exemple).

Par contre la mauvaise nouvelle : VirtualBox a été écrit pour être en partie compilé avec bcc, un compilateur C qui n'est pas l'habituel gcc. La raison vient des parties du code traitent du mode réel, qui est du code 8086 d'époque et qui nécessite compilateur et assembleur adaptés. Recompiler VirtualBox impose donc une dépendance très spécifique.

#### → Scénarios

#### Sécurité

L'isolation totale des programmes étant garantie par le choix de la technologie, aucun souci à se faire sur ce point : un seul serveur « plante » et tout le reste est conservé.

#### Consolidation de serveur

Cette solution s'est montrée la plus gourmande, la limite se trouvera certainement en terme de densité, en raison de la consommation forte en RAM. D'ailleurs pour cette utilisation particulière, une virtualisation complète consommera toujours plus de RAM qu'un isolateur. VirtualBox en consomme cependant beaucoup, plus que Qemu. Si une console d'administration à distance sans



Page: 30 / 61





nécessité de support graphique sur le serveur venait à voir le jour, VirtualBox gagnerait beaucoup sur cet aspect.

#### Plan de reprise d'activité

La consommation en RAM de VirtualBox sera le frein le plus important à son utilisation dans ce cadre, mais les utilitaires d'arrêt sur image (snapshot), et arrêt/redémarrage seront bien utiles dans un contexte aussi sensible, le retour sur investissement en RAM est immédiat en terme de sécurité

#### Hébergement

VirtualBox semble trop gourmand en mémoire pour être utilisé dans un contexte de déploiement en masse, cela occasionnerait un surcoût significatif. Cette utilisation de la virtualisation est cependant le terrain sur lequel les virtualiseurs niveau OS sont favoris.

#### • Développement et test, éducatif

Aucun problème pour l'aspect test, développement et éducatif : à l'image des déclinaisons de Qemu, c'est l'avantage majeur des virtualiseurs complets que d'offrir une séparation nette et franche entre les VM, à tous les niveaux logiciels qui leur sont accessibles. L'accélération matérielle offre de plus un surcroît de confort appréciable.

En terme de performances subjectives, tout va bien tant que la consommation de la RAM ne ralentit pas notablement la machine hôte puis par conséquent toutes les machines virtuelles. On conseillera donc d'utiliser cette solution sur une machine généreusement dotée. Heureusement, une version pour architecture x86-64 existe ce qui permet de l'installer sur une machine confortablement dotée.

#### Analyse de performances

VirtualBox montre des performances comparables à Kvm, mais ces performances s'écroulent dès que le swap entre en jeu, ce qui arrive un peu vite en raison de sa gourmandise. Il faudra prévoir une machine généreusment dotée, et pourquoi pas envisager l'architecture x86-64 avec laquelle VirtualBox est compatible et qui dispose d'un espace adressable très vaste.

# Retours d'expérience

VirtualBox a été le plus simple à utiliser, le démarrage d'une machine virtuelle demandant simplement quelques clics, et l'installation à partir d'un CD n'ayant pas posé le moindre problème pour le socle 2005. Par contre la version 1.3.6 a curieusement été incapable d'installer un FreeBSD sur une exception mémoire. Le test n'a pas été retenté sur la version 1.4.0 sortie entre temps (la correction était officiellement incluse).



Page: 31 / 61

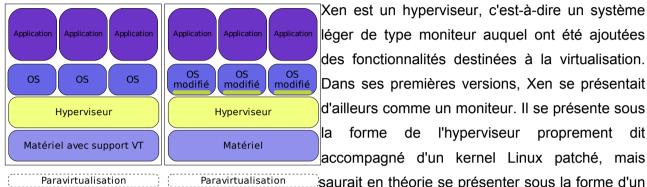


#### Liens

→ Site officiel: <a href="http://www.virtualbox.org/">http://www.virtualbox.org/</a>

#### 3.2.4 Xen

#### **Présentation**



de l'hyperviseur proprement accompagné d'un kernel Linux patché, mais saurait en théorie se présenter sous la forme d'un

autre système plus réduit.

Dans les versions actuellement libres et en l'absence de l'utilisation des extensions de virtualisation des fondeurs de x86, Xen nécessite de patcher les systèmes hôtes. Depuis la version 3.0 (pour Intel, et 3.03 pour AMD), Xen offre de plus la possibilité, sur les machines supportant les instructions de virtualisation, de faire tourner des systèmes d'origine.

Il existe une version commerciale de Xen, offrant plusieurs niveaux de support, un produit packagé, le support d'invités non modifiés sans instructions de virtualisation. L'un de ces niveaux est dédié aux serveurs Windows.

#### **Fonctionnalités**

Xen se présente sous la forme de l'hyperviseur lui-même, accompagné d'un kernel patché embarquant l'hyperviseur, que la terminologie Xen nomme Dom0, faisant tourner un ou plusieurs OS invités, nommés DomU. En théorie, Xen peut, dans sa version Open Source, faire tourner tout OS adapté pour se reloger dans l'espace d'adressage réservé et dans des niveaux de priorités processeur non maximum, ce qui signifie en pratique que la plupart des systèmes actuels ont besoin d'être patchés. C'est en particulier le cas de Windows.

De plus, exécuté sur un processeur disposant des extensions de virtualisation d'Intel et AMD, la version 3.1 de Xen est capable de démarrer des systèmes d'exploitation non modifiés, dont d'après la documentation, les versions de Microsoft Windows XP, 2000, Vista, sous réserve que les périphériques émulés se trouvent supportés par l'OS invité.



Page: 32 / 61





Notons une limitation de Xen : il est fortement ralenti par l'utilisation des libc embarquant la tls (thread local storage), actuellement utilisée par la plupart des distributions. Xensource propose, en guise de contournement, de la désactiver en renommant « /lib/tls » en « /lib/tls.disabled », la correction consistant à recompiler la libc sans support tls. Cette limitation est signalée lors du boot, dès que Xen détecte l'utilisation de la tls.

#### **Evaluation**

#### → Installation

L'installation de Xen est assez particulière, non qu'elle soit complexe mais elle requiert certaines versions très spécifiques du kernel. De l'aveu de XenSource et des laboratoires de l'Université de Cambridge d'où il vient, Xen n'est pas facile à porter d'une version du kernel Linux à une autre - ou plus exactement, le patch requis pour adapter un kernel Linux à devenir dom0 n'est pas trivial. Fort heureusement, Xen a fait son chemin jusqu'aux sources officielles du kernel, dans la déclinaison domU.

Xen a été intégré à la Fedora Core 6 et s'installe sans souci ; de plus, sa recompilation à partir des sources ne pose pas de difficulté particulière, à part que certains utilitaires requièrent bcc, le compilateur de Bruce Ewans générant du code x86 mode réel. C'est le même compilateur que celui requis pour recompiler VirtualBox Open Source Edition.

Concernant l'installation d'une distribution à partir du CD, il y a un détail qui mérite une certaine attention : l'installation par installateurs graphiques utilisant la lib curses ne marche pas sous Xen, il faut donc passer si possible en mode texte ou carrément par un installeur sous X11. L'alternative : passer par l'image tirée d'un système installé, par la commande dd par exemple, ou par copie pour le cas du déploiement d'une image sur plusieurs domU.

#### → Sécurité

La paravirtualisation, tout comme la virtualisation complète, garantit la séparation logique effective des machines virtuelles, pas de souci sur ce point. Cependant, dans le cas particulier de Xen, le système privilégié nommé Dom0 ou domain-0 est assez sensible : le mettre en difficulté (par une réservation mémoire inadaptée par exemple) peut conduire à une catastrophe. La **bonne pratique**, qui de plus conserve la similitude des hôtes virtualisés, consiste à réserver Dom0 pour l'administration du parc virtuel et placer les serveurs virtuels dans les DomU.

#### → Consolidation de serveur

A l'instar des Qemu et VirtualBox, installer n systèmes en tenant compte de la recommandation précadente demande n+1 noyaux. On aurait pu se dire, cependant, que le noyau Xen est notablement plus léger et limite un peu cette considération, mais, pour garantir la stabilité de



Page: 33 / 61





l'ensemble de la machine, on portera une attention particulière à la surveillance des services tournant dans le domain-0 si l'on insiste pour en laisser.

#### → Plan de reprise d'activité

La souplesse de Xen et sa possibilité de prendre en compte de multiples OS est un avantage, sa consommation en RAM, liée à la technologie utilisée, et liée à l'avantage dans le domaine de la sécurité, est son point faible. Par contre, installer des machines virtuelles à partir des supports n'est pas évident, un **plan de reprise d'activité** devra donc s'anticiper pour disposer des fichiers utiles en avance de phase.

### → Hébergement

Pour de l'hébergement de services en masse, on va être limité par la consommation en RAM, mais cet usage s'accomode mieux de virtualisation au niveau OS. En ce qui concerne la mise à disposition de machines virtuelles destinées à des utilisateurs, Xen peut être une solution.

### → Développement et test, éducatif

Xen vient du monde universitaire, et l'aspect développement et test sont certainement parmi les priorités du projet. Cependant, il reste relativement difficile d'installer une VM à partir des supports, en raison d'une séquence de boot moins bien maîtrisée que Booch et Qemu, en particulier en ce qui concerne la transition mode réel - mode protégé. Donc pour déployer des serveurs de développement ou de test à partir d'images préalablement générées, pas de problème, mais installer une distribution à partir d'un support se fera indirectement.

#### Analyse de performances

Parmi les meilleures de ces tests. On est proche des sensations d'une machine native, l'utilisation simultanée, au-dessus d'un Dom0 Linux, de 2 Linux DomU est fluide tout en ne chargeant pas exagérément l'hôte (1 CPU sur un core2 Duo).

De plus des astuces de configuration permettent de dédier des périphériques à un domaine, évitant le recours à une pile de couche d'émulation (non testé).

# Retours d'expérience

Le plus gros frein à Xen sera sans doute sa relative complexité de configuration, puisque le parti a été pris, eu égard au nombre de paramètres impressionnant qu'il propose, de se servir d'un fichier de configuration en sus d'un paramétrage partiel en ligne de commande, le dernier surchargeant le précédent. L'ergonomie est donc un peu passée de mode, même si elle se prête du coup assez bien à un déploiement de masse.



Page: 34 / 61



Autre inconvénient, la relative difficulté à installer un système d'exploitation depuis les supports, et le peu d'informations concernant la configuration elle-même. Le support des hôtes non adaptés ressemble à ce titre à une fonctionnalité non terminée : il faut déclarer un élément « kernel » spécifique nommé hvm (se chargeant de la virtualisation) et les périphériques de boot, pour un hôte Windows, Linux ou autre.

Le service OSSA de LINAGORA utilise Xen, c'est la solution la plus efficace et d'administration la plus rationnelle pour virtualiser des systèmes sur un hôte ne disposant pas des extensions processeur de virtualisation, à utiliser si l'on se contente de virtualiser des OS Open Source (puisqu'une modification de l'OS est nécessaire en l'absence de ces extensions).

#### Liens

- → Site officiel du support de Xen : <a href="http://www.xensource.com/">http://www.xensource.com/</a>
- → Site à l'University of Cambridge : <a href="http://www.cl.cam.ac.uk/research/srg/netos/xen/">http://www.cl.cam.ac.uk/research/srg/netos/xen/</a>

# 3.2.5 OpenVZ

## **Présentation**

Application

os

Application

OS

OpenVZ est une technologie de virtualisation de niveau OS basée sur le Application noyau Linux. OpenVZ permet à un serveur physique d'exécuter de os multiples instances de système d'exploitation isolés, connus sous le nom Couche de virtualisation de serveurs privés virtuels.

OS hôte En comparaison aux machines virtuelles telles que VMware et aux technologies de paravirtualisation telles que Xen, OpenVZ offre moins de Matériel flexibilité dans le choix du système d'exploitation : le système d'exploitation invité et hôte doivent être de type Linux. Cependant, la

virtualisation au niveau OS d'OpenVZ offre une meilleure performance, une meilleure scalabilité, une meilleure densité, une meilleure gestion de ressource dynamique, et une meilleure facilité d'administration que ses alternatives. Selon le site Web d'OpenVZ, cette méthode de virtualisation introduirait une très faible pénalité sur les performances : 1 à 3% de pertes seulement par rapport à un ordinateur physique.

OpenVZ est la base de Virtuozzo, un produit propriétaire fourni par SWsoft, Inc. OpenVZ est distribué sous la Licence publique générale GNU (GPL) version 2. L'entreprise à l'origine des deux produits, SWSoft, inc., vend du support sur OpenVZ et la suite Virtuozzo, qui propose des fonctionnalités supplémentaires comme migration d'une machine vers une VM et vice-versa, une



Page: 35 / 61



gestion par fermes et par serveurs nommés plutôt que par IP (plus besoin d'allouer des IP), des outils de surveillance, de statistiques, d'installation, de sauvegarde et d'administration par interface Web.

# **Fonctionnalités**

OpenVZ, de par sa conception, offre une excellente scalabilité, et permet une densité de serveurs très élevée, et offre en outre des outils facilitant la gestion de fermes de machines virtuelles.

**Scalabilité**: Comme OpenVZ utilise un modèle de noyau unique, il se comporte comme le noyau 2.6 de Linux: il supporte, sur architecture ia-32, jusqu'à 64 CPU et jusqu'à 64 GB de RAM. Un environnement virtuel unique peut être étendu jusqu'à la machine physique dans sa totalité, c'est-à-dire tous les CPU et toute la RAM.

En effet, certains déploient un unique environnement virtuel OpenVZ. C'est étrange à première vue, mais étant donné qu'un VE unique peut employer toutes les ressources matérielles avec une performance proche du natif, tout en bénéficiant d'autres avantages tels que l'indépendance du matériel, la gestion des ressources et la migration à chaud, ceci est un choix évident dans beaucoup de scénarios.

**Densité**: OpenVZ peut accueillir des centaines d'environnements virtuels sur un matériel décent (les limitations principales sont la RAM et le CPU). Des tests ont permis d'évaluer qu'une machine équipée de 768Mo de RAM pouvait être utilisée pour héberger jusqu'à 120 systèmes faisant tourner xinetd, init et les services de base plus apache, servant des pages statiques, avant que le swap excessif ne rende les temps de réponse rédhibitoires.

**Gestion de masse** : OpenVZ offre la possibilité de faire tourner un simple script qui mettra a jour tous les VE immédiatement, ou juste certains VE choisis.

#### → Sécurité

Peut être le moins fort des points d'OpenVZ, le fait de faire tourner tous les serveurs sur le même code kernel fait de ce code un point de défaillance potentiel important. On évitera donc naturellement les kernels non considérés comme stables, mais ces derniers ne sont de toutes façons pas à envisager en production. Les kernels stables, quand à eux, ont une stabilité digne de la réputation de Linux en la matière, aussi ce risque est-il très limité. En tous cas aucune crainte concernant les données intrinsèques à chacune des VM, la séparation est assurée pour celles-ci.

#### → Consolidation de serveur

De par son excellente scalabilité et en raison de son nombre minimal de couches de virtualisation, OpenVZ pourra optimiser l'utilisation de serveurs tous basés sur un système identique. En gardant le chiffre de 15% d'utilisation des serveurs actuels, si l'on considère que le surcoût dû à la



Page: 36 / 61





présence d'un système d'exploitation s'élève à 3% donc un coût unitaire de service d'environ 12%, ce sont 8 serveurs virtuels par CPU qui peuvent être simulés sur le modèle de OpenVZ, contre 5 serveurs plus l'hôte pour les autres modèles, sans même considérer l'aspect mémoire vive.

#### → Plan de reprise d'activité

A condition d'avoir une image prête à l'avance (comme pour Xen), l'utilisation de OpenVZ pour ce cadre est particulièrement adaptée si l'on a la chance que les services se basent tous sur le même kernel. En effet, on retrouve de nouveau l'argument scalabilité, OpenVZ est le seul système raisonnablement adapté à une virtualisation massive, donc à même de prendre à bras le corps le relais de gros serveurs. De plus des outils facilitent la mise à jour et la migration, ce qui est intéressant pour redéployer le matériel lors de l'intervention, ou déployer suite à mise à niveau matérielle. Cependant, la réalisation d'une image à partir d'une configuration opérationnelle est peu documentée et rentre clairement dans le cadre du produit commercial Virtuozzo.

#### → Hébergement

Là encore, à condition que l'aspect kernel commun ne soit pas rédhibitoire, cette solution s'utilise remarquablement pour de l'hébergement massif grâce à ses qualités de scalabilité, dus rappelons-le à une légèreté logicielle par conception et à la non multiplication des noyaux.

#### → Développement et test, éducatif

Peut-être le domaine pour lequel la condition de noyau commun est la plus gênante, et peut éventuellement être rédhibitoire : pas question ici d'installer une VM Windows, FreeBSD ou autre dans un Linux, tout le monde partage le même noyau. Ce produit n'est pas le plus souple pour cet usage qui demande justement une adaptabilité en priorité. Cependant, on peut être intéressé par une virtualisation des seuls serveurs d'application, et vouloir en faire simplement tourner plusieurs sur une même machine physique pour comparaison, et dans ce cadre OpenVZ apporte une réponse technique utile et très impartiale.

#### **Evaluation**

#### Installation

OpenVZ nécessite un noyau patché, ce qui détermine la version de kernel à utiliser de manière assez rigide, nous avons testé la dernière version d'OpenVZ, destinée au kernel 2.6.20. La configuration des options du kernel est aussi assez directive, à tel point que le site d'openVZ propose une configuration de base.

A cela, il faut ajouter les outils en ligne de commande vztools, disponibles sous la forme de paquets rpm sur le site dédié.



Page: 37 / 61





Cependant, OpenVZ demande, pour pouvoir démarrer un « Environnement Virtuel » (VE), toute une préparation d'arborescence (ou un cache tout prêt). Or, la préparation de ce genre d'environnement à partir de distributions non préparées par SWSoft, inc. n'est simplement pas documentée.

#### Scénarios

La technologie choisie par OpenVZ lui donne de notables avantages en termes de scalabilité, d'où des scénarios préférés de consolidation de serveurs, hébergement, plan de reprise d'activité.

Par contre ce choix technologique impose le système invité (nécessairement Linux), la version du kernel et virtualise "seulement" l'environnement machine (processus, libc, distribution, drivers en cours).

En terme de sécurité, le caractère dangereux d'une anomalie d'un serveur dépend directement des patches kernel de OpenVZ.

#### • Analyse de preformances

Les distributions particulières ne peuvent être facilement installées sans les dénaturer de façon trop importante, ce qui compromet un tes réellement objectif, il n'a donc pas été réalisé. On touche effectivement là à une limite du modèle logiciel libre jumelé avec une offre commerciale : les outils de mise en place de socle pour virtualisation font partie opportunément de l'offre commerciale. Ce qui peut être regretté, c'est qu'aucune documentation même manuelle n'est proposée, et si le choix de l'approche par système réduit et bootstrap s'applique globalement bien pour des distributions de type Debian ou Gentoo, cette approche s'accomode mal d'une distribution préparée et travaillée ou d'une distribution moins répandue. Aucun problème par contre pour les distributions majeures pour lesquelles des images prêtes sont distribuées non loin d'OpenVZ.

#### Retours d'expérience

Dès que l'on sort des chemins balisés pour lesquels les fichiers sont préparés, l'utilisation de OpenVZ demande des gymnastiques peu rassurantes et non documentées : en l'état, cette solution, en dépit de son impressionnante scalabilité, n'est pas aisément utilisable. Nul doute que c'est un chemin vers lequel SWSoft sera poussé à orienter son produit, mais en l'état, cette solution est trop peu praticable.

#### Liens

http://openvz.org/

http://fr.wikipedia.org/wiki/OpenVZ



Page: 38 / 61



## 4. Synthèse de l'étude

## 4.1 Comparaison des solutions retenues

Si une case est vide cela signifie que la fonctionnalité n'est pas présente.

Compa	ratif de	s solutions	étudiées		
	Qemu	VirtualBox	Xen	KVM	OpenVZ
Machine virtuelle	Х	Х			
Virtualisation OS					Х
Paravirtualisation			X		
Hyperviseur			Х	Х	
Manipu	ulations o	de Machines \	/irtuelles		
Création graphique		Х			
Sauvegarde					Х
Restauration					Х
Migration		Х	Х	Х	Х
		(commercial)	(commercial)		
	Invité	s Windows			
Système invité Windows sans	Х	Х	Х	N/A	
virtualisation matérielle			(commercial)		
Système invité Windows avec	Х	Х	Х	Х	
virtualisation matérielle					
	Dé <sub>l</sub>	ploiement	I		
Déploiement multiple	Х	Χ	Χ	Х	Х
Interface réseau virtuelle avec	Х	Х	Х	Х	Х
hôte					
OS invités multiples	Х	Х	X	Х	
Création scriptée	Х	Х	Х	Х	Х
Administration à distance		Х	Х		X
		(commercial)	(commercial)		(commercial)



Page: 39 / 61



## 4.2 Fiche d'identité des logiciels

### 4.2.1 Qemu

Propriétés génériques		
Cartouche		
Référence	Qemu 0.9.0	
Date d'évaluation	13/6/2007	
Auteur de l'évaluation	BDO	
Informations générales		
Nom du logiciel open source	Qemu	
Domaine fonctionnel du logiciel	Machine virtuelle	
Description succincte du logiciel	Méthode de virtualisation par émulation de périphériques et/ou processeur	
Licences auxquelles/à laquelle est soumis le logiciel	GPL (y compris l'accélérateur dédié)	
Site principal du projet open source	http://fabrice.bellard.free.fr/qemu/	
Systèmes d'exploitation compatibles	Tout système supportant gcc et bcc.  Qemu permet des machines virtuelles sous tout OS PC non modifié, y compris Windows.	
Démonstrations du logiciel (par	Plusieurs images de disques disponibles sur	
exemple portail démo pour un produit de portail)	http://www.oszoo.org/wiki/index.php/Category:OS_imag es	
Services existants		
Documentation	Toute la documentation utile est présente, mais en anglais uniquement.	
Nombre d'offres de support contractuel	Aucune dédiée sur le Web. Support OSSA est possible (licence GPL)	
Nombre d'offres de prestation de formation	Linagora	
Nombre d'offres de prestation de conseil	Linagora	
Aspects fonctionnels et techniques		



Page: 40 / 61



Technologie(s) d'implémentation	Machine virtuelle, émulation éventuelle à base de technologie Just-in-Time
Pré-requis techniques	La lib sdl est nécessaire si l'on veut une image graphique de la console (utile pour installer une distribution qui utilise ncurses, comme les Red Hat Entreprise)
Fonctionnalités détaillées	Emulation CPU, carte VGA Cirrus Logic 5440, Ethernet, boot mode réel
Plan de développement (roadmap)	Intégration des patches Kvm à venir.
Critères qualitatifs	
Pérennité	Ce projet, dont l'accélérateur dédié était auparavant sous licence spécifique, est intégralement GPL, bien écrit et intéresse suffisamment de monde pour être pérenne.
	4/5
Maturité	Issu de Bochs, Qemu est remarquablement stable et propose des options d'accélération intéressantes.
	5/5
Dynamisme de la communauté	5/5  Communauté réduite, mais efficace. On pourrait la souhaiter plus vaste, ou bénéficiant du support d'un industriel.
Dynamisme de la communauté	Communauté réduite, mais efficace. On pourrait la souhaiter plus vaste, ou bénéficiant du support d'un
Dynamisme de la communauté  Exploitabilité	Communauté réduite, mais efficace. On pourrait la souhaiter plus vaste, ou bénéficiant du support d'un industriel.
	Communauté réduite, mais efficace. On pourrait la souhaiter plus vaste, ou bénéficiant du support d'un industriel.  3/5  Tout en ligne de commande, en local sur la machine, l'exploitation de Qemu est perfectible pour des besoins d'administration, mais pour une utilisation R&D, tout est
	Communauté réduite, mais efficace. On pourrait la souhaiter plus vaste, ou bénéficiant du support d'un industriel.  3/5  Tout en ligne de commande, en local sur la machine, l'exploitation de Qemu est perfectible pour des besoins d'administration, mais pour une utilisation R&D, tout est là.
Exploitabilité	Communauté réduite, mais efficace. On pourrait la souhaiter plus vaste, ou bénéficiant du support d'un industriel.  3/5  Tout en ligne de commande, en local sur la machine, l'exploitation de Qemu est perfectible pour des besoins d'administration, mais pour une utilisation R&D, tout est là.  3/5  Sans accélération, on reste souvent en mode émulation, c'est très lent, surtout lors des entrées/sorties, mais néanmoins fonctionnel.



Page: 41 / 61



Adaptabilité	Modérée à forte, il lui manque la prise de contrôle à distance et une interface graphique, 4/5
Industrialisation	Pas de prise de contrôle à distance, pas de déploiements massifs, par contre tout le reste est fonctionnel, stable et robuste.
Synthèse	4/5
Synthese	
	Qemu est techniquement la solution libre de référence,
Tendance générale	surtout en raison de son émulation globale de PC très
Tonidanio generale	aboutie, mais il lui manque des outils d'administrateur
	pour emporter une plus large adhésion.
Notre commentaire	Solution techniquement impressionnante concernant l'émulation comme l'exécution native, nanti d'une accélération optionnelle efficace, il ne lui manque que les outils de l'administrateur et une interface graphique. Ces deux points sont en développement chez Red Hat : il pourrait donc devenir <i>la</i> solution de virtualisation. Qemu souffre pourtant d'une réputation de produit « bricolé » en raison de son origine non-industrielle, mais des solutions industrielles comme VirtualBox s'appuient en partie sur lui sans crainte.

#### 4.2.2 Kvm

Propriétés génériques		
Cartouche		
Référence	Qemu-Kvm	
Date d'évaluation	13/6/2007	
Auteur	BDO	
Informations générales		
Nom du logiciel open source	Kvm kernel module	
Domaine fonctionnel du logiciel	Virtualisation	
Description succincte du logiciel	Interface assistant à la virtualisation.	
Licences auxquelles/à laquelle est	GPL (intégré au code officiel du kernel et soumis à la	
soumis le logiciel	même licence)	



Page: 42 / 61



Site principal du projet open source	http://Kvm.qumranet.com/Kvmwiki	
	Hôte : Linux	
Systèmes d'exploitation compatibles	Invité : en théorie tout OS, Windows XP et Vista ont été	
	testés	
Démonstrations du logiciel (par		
exemple portail démo pour un produit	N/A	
de portail)		
Services existants		
	Jeune, à l'image du projet, la documentation s'appuie	
	largement sur celle de Qemu, qui constitue sa partie	
Documentation	contrôleur userland, mais elle suffit à démarrer une	
	machine sans peine.	
	<5 (produit trop récent)	
Nombre d'offres de support contractuel	Linagora TM2L OSSA possible (licence GPL)	
Nombre d'offres de prestation de	<5 (id.)	
formation	. ,	
	Linagora	
Nombre d'offres de prestation de	<5 (id.)	
conseil	Linagora	
Aspects fonctionnels et techniques		
	Kvm est un module kernel permettant un contrôle, par	
	les instructions de virtualisation, de contextes de MMU,	
Technologie(s) d'implémentation	ce qui permet à un programme ainsi contextualisé de se	
	croire seul sur une machine physique, sans intervention	
	logicielle.	
	Kvm est un module du kernel Linux donc ne tourne que	
Pré-requis techniques	sous Linux. Il utilise de plus les instructions de	
Tre-requis techniques	virtualisation donc un processeur comportant ces	
	instructions acivées est nécessaire.	
   Fonctionnalités détaillées	ld Qemu : émulation CPU, carte VGA Cirrus Logic 5440,	
1 onotionnantes detanices	Ethernet, boot mode réel	
	Optimisations en cours. Le support des instructions AM-	
Plan de développement (roadmap)	V de AMD est très jeune, mais apparemment de très	
Train de developpement (roadmap)	bonne qualité en raison d'une approche un peu plus	
	récente de la virtualisation.	
Critères qualitatifs		



Page: 43 / 61



Pérennité	Intégré récemment aux sources officielles du kernel, ce module kernel dispose d'une excellente réputation et se trouve intégré dans la cour des grands.  5/5
Maturité	Il reste quelques détails spécifiques à régler (en particulier, l'installation peut être difficile : générer une première image hors accélération peut s'avérer nécessaire) mais globalement la base est bonne. La maturation est en outre très rapide.  3/5
Dynamisme de la communauté	Module kernel récent mais réputé, officiel, dynamique, Kvm dispose de tous les atouts sur ce plan.  5/5
Exploitabilité	Exactement celle de Qemu, la performance en plus. En l'absence d'outils dédiés administration, il rate la note maximale.  4/5
Performance	Performance testée sur Intel: on est proche des performances natives. Les sites semblent dire qu'en raison d'une meilleure conception au niveau du silicium, une solution à base d'AMD pourrait être légèrement meilleure.  4/5
Adaptabilité	De nouveau celle de Qemu, si ce n'est qu'il nécessite un processeur supportant les instructions dédiées virtualisation pour fonctionner.  3/5
	Pas de prise de contrôle à distance, pas de
Industrialisation  Synthèse	déploiements massifs, quelques détails (ACPI) subsistent, par contre tout le reste est fonctionnel, stable et robuste.  3/5



Page: 44 / 61



Tendance générale	Produit à maturation rapide, ce module accélère Qemu
	de manière impressionnante.
	Kvm est l'accélération matérielle la plus efficace
	disponible pour Qemu, et s'il est encore jeune, il arrive
	très vite à maturité, comme le montre son intégration
Notre commentaire	précoce dans les sources officielles du kernel Linux. Un
	must pour Qemu sur plate-forme avec instructions de
	virtualisation. Cependant, cette solution pose parfois de
	petites difficultés, surtout dans la phase d'installation
	des systèmes invités (ACPI à déconnecter par
	exemple).

#### 4.2.3 VirtualBox

Propriétés génériques		
Cartouche		
Référence	VirtualBox OSE 1.4.0	
Date d'évaluation	13/6/2007	
Auteur	BDO	
Informations générales		
Nom du logiciel open source	VirtualBox OSE (Open Source Edition) version 1.4	
Domaine fonctionnel du logiciel	Machine virtuelle accélérée	
Description succincte du logiciel	VirtualBox est un logiciel de virtualisation par machine virtuelle.	
Licences auxquelles/à laquelle est soumis le logiciel	GPL (produit Open Source Edition), des parties propriétaires (très clairement identifiées sur le site dédié : <a href="http://www.virtualbox.org/wiki/Editions">http://www.virtualbox.org/wiki/Editions</a> )	
Site principal du projet open source	http://www.virtualbox.org/	
Systèmes d'exploitation compatibles	Hôtes: Linux Invités: Linux, Windows (une petite difficulté à contourner pour Vista), FreeBSD (version 1.4.0 impérative).	
Démonstrations du logiciel (par exemple portail démo pour un produit de portail)	Télécharger le logiciel et créer une machine est rapide et simple, c'est la meilleure démonstration.	



Page: 45 / 61



Services existants  Documentation	En anglais, mais l'effort de traduction a été démarré.		
Documentation			
Nombre d'offres de support contractuel	Innotek GMBH pour la version commerciale		
	OSSA possible pour la version Open Source		
Nombre d'offres de prestation de	Innotek offre le support de la version commerciale		
formation	innotest offic to dapport do la version commerciale		
Nombre d'offres de prestation de	Non trouvé		
conseil	Tron dedice		
Aspects fonctionnels et techniques			
Technologie(s) d'implémentation	Machine virtuelle avec accélération par module dédié.		
Duć vo svije to abvijeva	Pour l'installation :		
Pré-requis techniques	Pour la recompilation : as86, bcc, ld86, gcc		
	Emulation d'un PC complet pour x86, support du mode		
Fonctionnalités détaillées	PAE et de l'x86-64, interface de contrôle local.		
Plan de développement (roadmap)	Non publié		
Critères qualitatifs			
	Innotek GMBH semble bien tenir ce produit, qui évolue		
	régulièrement. L'approche open source avec la		
Pérennité	commercialisation de fonctionnalités nettement		
	identifiées est un plus au niveau transparence.		
	4/5		
	Capable d'émuler un PC assez bien pour installer des		
	systèmes d'exploitation hors Linux, proposant une		
	interface agréable et simple, il est très fonctionnel.		
Maturité	Quelques difficultés pour l'installation de OpenBSD		
	cependant.		
	Peu visible hors de Innotek, mais bien organisée et		
	reu visibie nois de innotek, mais dien organisee et		
Dynamisme de la communauté	vivement encouragée. La transparence de leur position		
Dynamisme de la communauté	vivement encouragée. La transparence de leur position vis-à-vis de leur double licence aide.		
Dynamisme de la communauté	vivement encouragée. La transparence de leur position vis-à-vis de leur double licence aide.  3/5		
Dynamisme de la communauté	vivement encouragée. La transparence de leur position vis-à-vis de leur double licence aide.  3/5  En R&D, le logiciel est excellent et permet de goûter à la		
Dynamisme de la communauté  Exploitabilité	vivement encouragée. La transparence de leur position vis-à-vis de leur double licence aide.		



Page: 46/61



Performance	Globalement bonne, le produit bénéficie d'une accélération dédiée. 4/5
Adaptabilité	Son plus gros talon d'Achille est sa gourmandise en RAM, ce qui rend difficile son utilisation en tant que parc de serveurs virtuels. Donc à réserver à la R&D.
Industrialisation	Stable, efficace, mais gourmand et peu administrable à distance, ce logiciel se destine tout d'abord aux laboratoires. Si toutefois la prise d'instantanés et la migration sont essentielles, l'effort nécessaire pour le maintenir en opération sera productif. Nul doute que Innotek va investir dans les logiciels d'administration, et le coût de la RAM rend petit à petit acceptable le premier point.
Synthèse	3/5
Tendance générale	Excellent produit pour laboratoire, en particulier accompagné par une interface permettant de créer une VM sans se tromper et sans le moindre recours ni au mode d'emploi, ni au manuel.
Notre commentaire	Ce produit est idéal pour goûter à la virtualisation : facile et efficace, il est juste un peu coûteux en mémoire pour le solliciter à virtualiser massivement un pool de serveurs. Il propose un niveau de performances en phase avec son orientation R&D ou tests, et une stabilité pleinement satisfaisante. Pour l'envisager en production, il faudra simplement qu'apparaissent les outils d'administration à distance qui lui font défaut. Le besoin de fonctionnalités d'instantanés ou de migration peuvent toutefois justifier qu'on le déploie industriellement.



Page: 47/61



#### 4.2.4 Xen

Propriétés génériques	
Cartouche	
Référence	Xen 3.1
Date d'évaluation	14/6/2007
Auteur	BDO
Informations générales	
Nom du logiciel open source	Xen
Domaine fonctionnel du logiciel	Paravirtualiseur
Description succincte du logiciel	Xen, dans sa version 3.1, est un moniteur de paravirtualisation pour les processeurs sans instructions de virtualisation, et fonctionne en paravirtualiseur complet, c'est-à-dire qu'il permet à un système non modifié d'être virtualisé, avec ces instructions.
Licences auxquelles/à laquelle est	
soumis le logiciel	GPL
Site principal du projet open source	http://www.xensource.com/
Systèmes d'exploitation compatibles	Hôte (Dom0): Linux (certaines versions spécifiques)  Invités (DomU):  Tout système sur AM-V ou VMX,  systèmes adaptés sans ces instructions (ce qui exclut Windows).
Démonstrations du logiciel (par exemple portail démo pour un produit de portail)	
Services existants	
Documentation	En anglais sur le site <a href="http://www.xensource.org">http://www.xensource.org</a> , un site français existe mais il n'est plus à jour.
Nombre d'offres de support contractuel	Xensource: 3 niveaux de produit, une entrée de gamme, un produit orienté Windows et un produit serveur haut de gamme.  La version open source est sous GPL donc un support Linagora/TM2L OSSA est possible.



Page: 48 / 61



Nombre d'offres de prestation de formation	Linagora		
	Des tutoriels Web existent		
Nombre d'offres de prestation de conseil	Plusieurs prestataires, dont LINAGORA.		
Aspects fonctionnels et techniques			
Technologie(s) d'implémentation	Solution de paravirtualisation logicielle, complète (avec support d'instructions de virtualisation) ou partielle.		
	A l'origine accompagné d'un BSD, Xen se présente		
Pré-requis techniques	aujourd'hui avec un noyau Linux patché, lequel sert de Dom0.		
	Xen permet le contrôle (démarrage, pause, arrêt) de		
	machines virtuelles. Les versions soumises à licence		
Fonctionnalités détaillées	ajoutent les fonctionnalités de portage (d'une machine		
	vers une VM), migration (d'un Xen à un Xen d'une autre		
	machine), etc.		
Plan de développement (roadmap)	Optimisation de l'utilisation des opérations de virtualisation.		
Critères qualitatifs			
	C'est le moniteur de paravirtualisation le plus connu.		
	Issu d'un projet universitaire du Computer Laboratory de		
Pérennité	Cambridge, il est maintenant industrialisé par une		
	société dédiée, XenSource.		
	4/5		
	La version 3.0.3 a fini d'apporter ce qui manquait : le		
	support d'OS non modifiés par utilisation des		
	instructions de virtualisation. Le matériel adéquat étant		
Maturité	destiné à se généraliser, on a aujourd'hui tout ce qu'il		
	faut, au prix éventuel d'une configuration complexe.		
	5/5		



Page: 49 / 61



Dynamisme de la communauté	L'Université de Cambridge propose toujours des projets autour de Xen, XenSource assure le suivi industriel par la commercialisation de package supportés, c'est une excellente entente institutions/industrie qui gère et dynamise Xen.  4/5	
Exploitabilité	Xen est extrêmement configurable, mais les configurations nécessitent de bien être comprises et demandent une expertise ou une très grande patience. L'apport de Red Hat, qui propose une interface facilitant cette mise en œuvre, est bienvenu.  3/5	
Performance	Parmi les meilleures. Comparables à Kvm, avec le bénéfice supplémentaire de pouvoir, moyennant une configuration aux petits oignons, dédier des ressources matérielles aux domaines. Changer ces allocations n'est toutefois pas dynamiques (aucun système ne saurait d'ailleurs assurer ce changement sans arrêt/démarrage du périphérique).	
Adaptabilité	Sur le papier, c'est la solution la plus souple. La performance est au rendez-vous. Dans la réalité, le seul frein c'est le besoin de maîtriser les subtilités de sa configuration.  5/5	
Industrialisation	En version Open Source, il est déjà largement utilisé, et Red Hat l'a inclus dans la Fedora Core 6 en confiance. Mais la configuration complexe ternit le tableau. Une fois le système prêt, il est cependant sans faille.  3/5	
Synthèse		
Tendance générale	Produit excellent mais configuration trop complexe et peu pratique, nécessitant recours à un expert ou à de nombreuses recherches sur la toile.	



Page: 50 / 61



	Un produit complet, qui remplit bien sa mission, et qui,
	depuis la version 3 (stabilisé en 3.1.0 pour AMD et
	Intel), virtualise les systèmes d'exploitation invités natifs,
	mais dont la configuration, certes riche, est peu lisible et
	désuète. Ne pas s'embarquer dans Xen sans expert
	disponible

## **4.2.5 OpenVZ**

Propriétés génériques				
Cartouche				
Référence	OpenVZ Linux 2.6.20			
Date d'évaluation	14/6/2007			
Auteur	BDO			
Informations générales				
Nom du logiciel open source	OpenVZ			
Domaine fonctionnel du logiciel	Virtualisation d'OS			
Description succincte du logiciel	Couche de virtualisation d'OS par isolation logicielle en			
Description succincte du logiciei	espace utilisateur			
Licences auxquelles/à laquelle est	GNU GPL V2			
soumis le logiciel	GNU GPL V2			
Site principal du projet open source	http://www.openvz.com			
	Hôte : Linux			
Systèmes d'exploitation compatibles	« Invité » : kernel identique à l'hôte.			
Démonstrations du logiciel (par	Des configurations de démonstrations sont disponibles.			
exemple portail démo pour un produit	-			
de portail)	embarquant leur solution.			
Services existants				
	Sur le site, en Anglais. Introduction en Français.			
Documentation	Manque la méthode de génération d'une VM à partir			
	d'une arborescence système			



Page: 51 / 61



Nombre d'offres de support contractuel  Nombre d'offres de prestation de formation  Nombre d'offres de prestation de	SWSoft, inc. (qui distribue aussi Virtuozzo, version commerciale accompagnée d'outils de déploiement, gestion et administration à distance, migration)  Linagora  Non présentes sur le Web  Linagora	
conseil	Linagora	
Aspects fonctionnels et techniques		
Technologie(s) d'implémentation	Virtualisation d'OS par isolation : les machines virtuelles sont en fait des groupes de processus disposant d'un contexte à l'image de celui d'une machine (root filesystem, propre table des processus, propres entrées dans /dev, propres utilisateurs y compris root etc.)	
Pré-requis techniques	Ce logiciel ne tourne que sur un Linux, seules certains kernels sont patchés, la configuration de base est fournie car assez spécifique.	
Fonctionnalités détaillées	Virtualisation par isolateur Espaces d'exécution séparés Allocation RAM et CPU dynamiques QOS garantie (chacun des environnement virtualisé bénéficie d'un temps CPU garanti par un algorithme de tourniquet)	
Plan de développement (roadmap)	Intégration en cours dans le kernel officiel	
Critères qualitatifs		
Pérennité	Bien supporté par l'entreprise initiatrice, cette solution originale dispose des moyens d'exister.  4/5	
Maturité	La version Open Source est trop difficile de mise en place pour être crédible, ce produit ne trouve pas ses utilisateurs en l'état. C'est probablement un choix marketing, voué à évoluer.  2/5	



Page: 52 / 61



Dynamisme de la communauté	OpenVZ est bien géré par SWSoft, inc. mais la communauté semble peu nombreuse à l'extérieur.  3/5			
Exploitabilité	La préparation d'une plate-forme spécifique est trop complexe, non documentée, c'est le point faible de cette solution. Notons que la version commerciale inclut opportunément des outils de migration.			
Performance	Très proche des performances natives sur le papier, mais la plate-forme de test n'a pu être montée sur la même base que les autres logiciels.  N/A			
Adaptabilité	L'approche technique est déjà spécifique, le produit délibérément orienté distributions rpm renforce son caractère un peu solitaire.  2/5			
Industrialisation	L'industrialisation est difficile en l'état à moins de vouloir se baser sur une des machines déjà prêtes. Ce produit est-il délibérément bridé au profit de la version commerciale ? Il faut faciliter la création de VE plus variés.  1/5			
Synthèse				
Tendance générale	Une approche spécifique (pas de machine Windows virtuelle ici, sauf à utiliser Wine), mais intéressante, focalisée sur la densité et la performance. Ce produit est intéressant pour des batteries de services.			



Page: 53 / 61



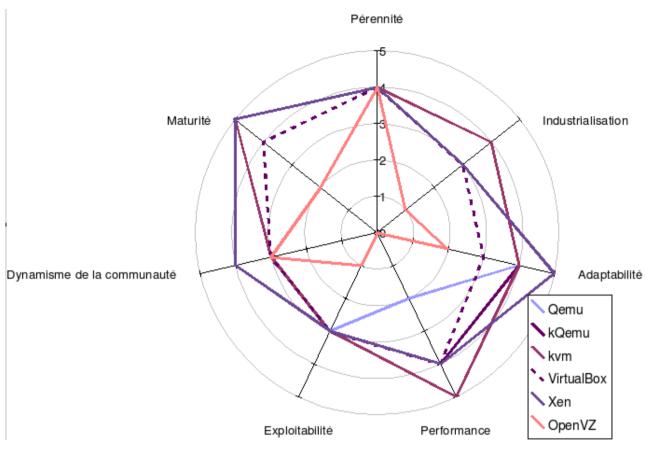
	Son plus gros handicap est la difficulté à démarrer une
	VM « maison ». La méthode la plus utilisable pour cela,
	c'est de partir d'une distribution minimale déjà prête et
	d'additionner les installations spécifiques par-dessus
	(bootstrap) : c'est une approche inacceptable pour les
	équipes qui ont fait l'effort de packager une distribution.
Notre commentaire	La raison de cette difficulté semble cependant d'ordre marketing.
	Une fois la configuration prête, cette solution a des
	usages tous désignés : la virtualisation massive de
	serveurs pour hébergement de services est sa première

cible.



Page: 54 / 61





	Pérennité	Maturité	Dynamisme	Exploitabilité	Performance	Adaptabilité	Industrialisation
Qemu	4	5	3	3	2	4	4
kQemu	4	5	3	3	4	4	4
kvm+Qemu	5	3	5	4	5	3	3
VirtualBox	4	4	3	3	4	3	3
Xen	4	5	4	3	4	5	3
OpenVZ	4	2	3	1	N/A	2	1

## 4.3 Perspectives des technologies

L'intégration récente de Kvm au kernel Linux ouvre de nouvelles voies aux projets de virtualisation.

Actuellement utilisée par le seul Qemu patché, cette extension (sous la forme d'un pilote de périphérique) permet d'envisager une utilisation de Linux comme hyperviseur comme l'est Xen, avec l'avantage de permettre le démarrage d'un kernel non modifié spécifiquement (à l'instar d'ailleurs de Xen 3.0 et suivants sur une machine supportant les extensions de virtualisation).





Il semble que toute l'offre technique est en place et mature, proposant des performances de tout premier plan. Ce qui fait encore défaut aux logiciels open source de virtualisation, c'est essentiellement une suite d'outils facilitant l'administration des machines virtuelles tout d'abord, et l'accès simultané aux accélérations matérielles ensuite. Le premier point est globalement très nettement à l'avantage des solutions commerciales, le second point commence à peine à exister, dans une solution commerciale et dans un cadre très spécifique d'ailleurs.

Notons que Red Hat Software a récemment initié un travail d'interface globale de virtualisation, dans le but de faciliter le contrôle et le déploiement des virtualiseurs proposés par leurs distributions. Si ce travail est encore jeune et manque de maturité, il montre la voie à suivre et indique bien que les acteurs de l'Open Source, conscients des attentes des utilisateurs, partent dans le bon sens.

Il reste un aspect sur lequel les solutions techniques hors isolateur pourraient gagner en performance globale de plateforme par mutualisation de ressources : actuellement, la mémoire d'échange (swap) est conçue pour un système unique sans concurrence. Placer du swap dans une machine virtuelle peut aujourd'hui se faire de deux manières différentes. Soit on dédie une partition de swap à chacune des machines virtuelles, ce qui peut raisonnablement se faire au regard de la taille des disques aujourd'hui mais qui exige techniquement de piloter une partition d'un disque depuis une machine virtuelle : cela demande que la solution de virtualisation est explicitement prévu ce cas. Soit, en seconde solution, on place le swap dans un fichier image de disque monté en loopback, solution plus simple mais terriblement coûteuse : la mémoire virtuelle, déjà plus lente que la RAM d'un facteur 100 en accès natif, doit alors passer au travers d'une couche périphérique et une couche « filesystem » supplémentaires. Il reste une solution contestable : affecter une grande partie de RAM à la machine virtuelle et aucun Swap, laissant le soin au système hôte de paginer. Le comportement sera peu prévisible alors, le système hôte n'ayant pas vraiment les bonnes informations quand à l'utilisation des pages.



Page: 56 / 61



#### 4.4 Comment faire?

En raison de la variété des solutions et de leur évolution, LINAGORA peut vous accompagner pour accéder aux bénéfices des technologies de virtualisation en deux temps : un audit préalable pour définir les besoins et sélectionner la solution la plus adaptée, puis la mise en œuvre de la solution optimale et des outils complémentaires. LINAGORA est également en mesure de réaliser en mode communautaire des d'outils d'administration complémentaires.

#### 4.4.1 Réalisation d'un audit des besoins de virtualisation

L'objet de cette première étape est de bien mesurer les enjeux de la virtualisation dans votre environnement et les différents contextes d'emploi actuels et potentiels.

Cette étape devrait permettre de déterminer les usages précis, la cible exacte et, donc, la possibilité et l'intérêt de la virtualisation dans votre entreprise. Il convient alors de sélectionner les solutions de virtualisation open source les plus adaptées aux besoins identifiés, et de proposer un plan d'action. Cet audit pourra proposer le cas échéant le développement ou l'accélération du développement d'une suite d'administration.

# 4.4.2 Configuration du ou des outils les plus appropriés et réalisation d'une suite d'outils d'administration

Une fois la première phase réalisée, l'étape suivante vise à déployer les solutions de virtualisation open source choisies et, le cas échéant, d'engager le développement d'une suite d'outils portant cette ou ces solutions à un niveau permettant une utilisation performante et confortable en utilisation industrielle. Cette étape prendra en compte la question du reversement des solutions développées.



Page: 57 / 61



#### 4.5 Méthodologie

Le modèle et la démarche proposés pour l'évaluation des produits open source est basée sur l'Argus du Libre ce qui permet de :

- → Déterminer si un produit Open Source est convenable pour une organisation,
- → Définir et/ou identifier en 7 critères permettant de déterminer si un logiciel libre répond aux besoins.

#### L'Argus du Libre permet de :

- 1. Déterminer la maturité du logiciel libre.
- 2. Trouver un produit répondant aux critères fonctionnels attendus.
- 3. Évaluer l'existence d'une communauté assurant le support et la pérennité du produit.

#### 4.5.1 Définition des critères communs

Les critères communs habituellement utilisés dans l'Argus du Libre sont les suivants, ceux marqués par (\*) ne sont pas utilisés dans le cadre de ce livre blanc ou sont combinés (\*\*) :

- → La pérennité : plus un logiciel est en développement actif depuis longtemps moins il a de chance de péricliter. Parce que le plus souvent, la première année est décisive, la majorité des projets n'aboutissent pas au bout d'un an, il est important d'évaluer l'age du logiciel et l'activité de ses développeurs.
- → La maturité : les projets qui ne dépendent que d'un nombre restreint de personnes, voir d'une seule ressource, tendent à mourir très vite. La maturité des projets est souvent liée au nombre de membres actifs dans la communauté ainsi qu'à leur organisation.
- → Le dynamisme de la communauté : ce critère est mesurable en évaluant la quantité et la qualité des canaux d'échanges autour du produit – listes de diffusion, groupes de discussions, sites d'information ...
- → L'exploitabilité : la capacité de déploiement peut être mesurée à la documentation disponible ainsi que l'offre de formation disponible sur le marché pour cet outil.
- → La capacité d'intégration avec d'autres produits : ce critère peut être mesuré en évaluant la compatibilité de l'outil avec les normes et standard en vigueur.

LIN AGORA
GROUPE

Page: 58 / 61



- → L'adaptabilité : l'architecture du logiciel est un critère important pour déterminer s'il est possible et avec quelle difficulté, d'intervenir pour adapter le comportement à un usage spécifique ou pour ajouter une fonctionnalité manquante.
- → Industrialisation : il s'agit d'un critère de regroupement nous permettant de donner notre avis global sur la possibilité d'utiliser le logiciel en environnement de production ou de le proposer à des utilisateurs.



Page: 59 / 61

## 4.5.2 Définition des niveaux pour chaque critère.

Indicateur	Score: 1	Score: 3	Score: 5
Pérennité	mainteneurs principaux ou un mainteneur unique depuis sa création et peut se retrouver	composé d'un petit groupe de personnes actives. Un nombre	Le groupe de développeurs est stable et la technologie récente ou largement stabilisée. L'organisation est très clairement identifiée et structurée.
Maturité	démarrer. La stabilité du groupe	quelques années. Les objectifs sont identifiés pour la prochaine	Le projet est actif depuis plusieurs années. La stabilité du code est avérée et les objectifs à court et moyen terme sont définis clairement
Dynamisme de la communauté	•	existent avec chacune leurs objectifs (liste d'utilisateur, liste de développement, liste d'annonce et	Les listes de diffusion sont très actives et certaines existent hors des listes « officielles » du projet. On trouve des correctifs et des améliorations de sécurité maintenus par d'autres développeurs.
Exploitabilité	La documentation se compose d'un document non traduit, n'est pas forcément à jour et s'adresse à des administrateurs avertis.	jour. Elle est orientée principalement vers les	,
Intégration	Le projet ne communique par	Le projet dispose de quelques	Le projet est intégré avec des

Indicateur	Score: 1	Score: 3	Score: 5
	nécessite des versions spécifiques. Il ne tourne que dans un ensemble restreint	d'expériences sur l'intégration avec d'autres composants qui lui sont tout naturellement associés (sauvegarde et supervision par exemple,)	standards à d'autres projets / produits. Il se déploie sur différents systèmes et supportent toute ou partie des particularités de ces derniers. Il existe sous forme de paquets divers voir d'installeurs.
Adaptabilité	projet nécessite une excellente	projet nécessite quelques compétences mais dispose d'une documentation simplifiant les	La modularité du code autorise une connaissance parcellaire du code via, notamment des mécanismes de greffons et une documentation de développement.
Industrialisation	précédents : le logiciel est recommandé pour un usage particulier en prenant en	peut être satisfaite par ce logiciel même s'il a des limites pour des usages spécifiques (grande volumétrie, haute disponibilité,	