REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
ECOLE NATIONALE POLYTECHNIQUE.

ECOLE NATIONALE POLYTECHNIQUE
D'ALGER
DEPARTEMENT D'ELECTRONIQUE

Mémoire de fin d'études

En vue de l'obtention du
Diplôme d'Ingénieur d'Etat en Electronique

Thème :

Couche de Virtualisation Xen pour le Déploiement de Serveurs Internet

Réalisé par:

BOUSSA Samir
BOUTELLIS Nabil

proposé par : **M. SADOUN Rabah**

Soutenu le 28 juin 2008 devant le jury composé de

Dr. KACHA	Président
Mr. SADOUN	Rapporteur
Mr. ZERGUI	Examinateur

-Promotion Juin 2008-

Remerciements

On tient à exprimer nos vifs remerciements pour notre promoteur M. Rabah SADOUN pour avoir proposé ce sujet, prodigué ses précieux conseils tout au long de notre travail, son aide et sa confiance.

On remercie très chaleureusement les membres de jury pour l'honneur qu'ils nous ont fait en acceptant d'être les rapporteurs de ce mémoire.

On remercie aussi tous nos enseignants de l'Ecole Nationale Polytechnique d'Alger, pour le savoir qu'ils nous ont transmis. Leur disponibilité et leur gentillesse.

On tient à remercier tous ceux qui ont participé de prés ou de loin à la réalisation de ce modeste travail. Plus particulièrement l'équipe du centre de calcul.

Enfin, on aimerait adresser nos plus fervents remerciements a nos parents qui sans eux nous aurions pas atteint ce stade.

BOUTELLIS Nabil

BOUSSA Samir

Dédicace

Je dédie ce modeste travail à toutes les personnes que j'aime :

- *A ma très chère mère qui a veillée sur mois pendant toute ma vie.*
- *A mon père.*
- *A mes frères et mes sœurs : Kamel, Nacer, Razika, Djamila, Toufik, Chahira, Sofiane, Omar. Sans oublier Naoual.*
- *A Sidali, Amina, Sara, Issam, Ikrame, Inesse, Wissame, ferial, oussama, Maroua, Faress. Et surtout la petite Hadile*
- *A tous mes amis intimes : Moh, Redouan, Charika sans oublier Sofiane et Hicham.*

A toute la promotion 2008.

BOUSSA Samir

Je dédie ce modeste travail à :

Mon cher oncle défunt M. Boutellis Boualem que Dieu accueille son âme dans son vaste paradis.

Mes très chers parents.

Mon frère billel et ma sœur Amina.

Toute ma famille

Tous mes amis et plus particulièrement a Krimou et Lydia.

Tous les profs qui m'ont soutenu de l'école nationale polytechnique.

BOUTELLIS Nabil

SOMMAIRE

page

CHAPITRE 3 : Mise en œuvre de l'hyperviseur et tests de performances

Liste des figures

Page

Liste des tableaux

Table des Abréviations

VM : Virtual Machine.

VMM : Virtual Machine Monitor.

OS : Operating system.

VT : Virtual Technology .

GPL / GNU : General Public License .

IP : Internet Protocol.

MAC : Media Access Control.

NAT : Network Address Translation.

NIC : Network Information Center.

DHCP : Dynamic Host Configuration Protocol.

VLAN : Virtual Local Area Network

VPN : Virtual private Network.

DNS : Domain Name System.

AMD : Advanced Micro Devices.

CPU : Central Process Unit.

RISC : Reduced Instruction Set Computer.

CISC : Complex Instruction Set Computer .

PV : Para virtualization.

RAM : Random Access Memory**.**

I /O : Input Output.

VMCS : Virtual Machine Control Structure.

VPD : Virtual Processor Descriptor.

VAC : Virtualization Acceleration Control.

DMA : Directs Memory Access .

SVM : Support Vector Machine.

VDI: Virtual Disk Image.

OSE: Open Source Edition.

TPS: Transfer Page Sharing.

LVM: Logical Volume Manager.

Introduction générale

L'informatique n'a pas cessé de progresser ces dernières années laissant pour unique limite l'évolution de l'outil Hardware. Par contre la miniaturisation et la mise au point de processeur de plus en plus performants et de moins en moins volumineux a créé une nouvelle problématique qui est la sous exploitation des ordinateurs actuels. D'ou la nécessitée de la virtualisation.

Cette dernière est une technologie logicielle éprouvée qui transforme rapidement le paysage informatique et change radicalement l'approche de l'informatique.

À l'origine, le puissant matériel informatique x86 dont nous disposons actuellement a été conçu pour n'exécuter qu'un seul système d'exploitation et qu'une seule application. La virtualisation dépasse ces limites en permettant d'exécuter simultanément plusieurs systèmes d'exploitation et plusieurs applications sur le même ordinateur, ce qui accroît l'utilisation et la flexibilité du matériel.

Dans ce projet de fin d'études on a fait une étude sur les différentes solutions de virtualisation comme VirtualBox, VMware en testant leurs performances dans des environnements différents tels que Windows et Linux. Sans oublier bien sur la solution de paravirtualisation qui est le Xen. Tout cela dans le but de choisir une solution adaptée a notre application.

Ce rapport est structuré comme suit, dans le chapitre 1 et 2 on a fait une étude théorique détaillées sur la virtualisation. Suivi dans le chapitre 3 de la mise en œuvre et les testes de performances des trois solutions sélectionnées. Pour le chapitre 4 nous allons vous montrez comment on a pu virtualiser des serveurs Internet.

On espère que ce modeste travail aidera quiconque s'intéressant à la solution de virtualisation.

On vous souhaite une très bonne lecture.

CHAPITRE 1
Historique de la virtualisation

1.1 Introduction

Il n'y a pas aujourd'hui un article, un analyste ou un éditeur qui ne vous parle de virtualisation. La médiatisation du concept cache mal la révolution en marche qu'est en train de vivre notre industrie informatique. Les récents événements de l'été 2007 en témoignent : Citrix a racheté Xen Source; VMware est entré en bourse ; Microsoft vient de sortir sa console de gestion des infrastructures virtuelles ; on vient de mettre sur le marché des microprosseurs à quatre cœurs supportant mieux les infrastructures virtuelles.

On nous parle de virtualisation des serveurs, des postes de travail et même de virtualisation des applications ! Mais au-delà de ces concepts et de ces mots qui sonnent bien et qui font pro, qu'est-ce que réellement la virtualisation et plus particulièrement la virtualisation des serveurs ?

Dans ce premier chapitre on donnera un bref aperçu historique de la virtualisation et les différentes techniques existantes.

1.2 Historique

Les premiers ordinateurs, qui occupaient plusieurs pièces d'un bâtiment, n'étaient pas faits pour exécuter *plusieurs* programmes à la fois. On concevait un programme (qui était à l'époque une simple succession de calculs), on le mettait dans la file d'attente des programmes, et quand le système d'exploitation avait fini de traiter un programme, on lui donnait le suivant dans la liste.

Premiers pas

Très vite, dès la fin des années cinquante, l'idée de pouvoir exécuter plusieurs programmes en parallèle voit le jour. On parle de temps partagé *(time sharing)*, de multiprogrammation, etc. L'idée était de pouvoir faire cohabiter plusieurs programmes au même moment, ayant tous accès au même matériel, sans qu'ils ne se gênent mutuellement. La virtualisation est très proche de concept.

Au milieu des années soixante, IBM effectue des recherches sur les systèmes virtualisés les invités étaient gérés par une simple multiprogrammation. En 1967 est lancé, toujours par IBM, le système CP-40, le premier système offrant une virtualisation complète.

À partir de ce moment là, les technologies ont vraiment progressé, avec l'arrivée de nouveaux acteurs toujours prêts à innover pour se démarquer des concurrents.

Évolutions récentes

Du côté du logiciel propriétaire, l'arrivée du Microsoft comme éditeur de solutions de virtualisation a relancé la course au développement de technologies performantes. En effet, la plupart des concurrents de la société Microsoft ont peur qu'il ne réitère ce qu'elle avait fait pour Netscape Navigator, alors en concurrence avec son logiciel Internet Explorer : diffuser en masse, avec son système d'exploitation, un produit techniquement inférieur dans l'espoir d'affaiblir, voire d'éliminer, leurs concurrents.

Cette stratégie lui avait plutôt réussi par le passé, aussi les craintes des éditeurs de produits de virtualisation pour la plate-forme Windows sont légitimes. Microsoft a également annoncé que la prochaine version de son système d'exploitation pour serveur, pour l'instant encore dénommée *Longhorn*, pourra fonctionner avec un hyperviseur, développé en collaboration avec la société XenSource. [1.5]

En parallèle, la communauté du logiciel libre continue elle aussi à faire évoluer les différents projets. Récemment, une architecture visant à faciliter la paravirtualisation, nommée paravirt_ops, a été intégrée au noyau Linux. Elle permet à toutes les solutions de virtualisation de faire fonctionner un système GNU/Linux avec paravirtualisation sans avoir à modifier le code du noyau.

De même, une interface pour la virtualisation des Entrées/Sorties, nommée virtio, est actuellement en cours de conception par les développeurs du noyau Linux. Cette interface

permettra d'offrir une base commune pour les E/S, pour toutes les solutions de virtualisation, y compris pour la paravirtualisation et les hyperviseurs.

Les fondeurs de processeurs de la famille PC—c'est à dire Intel et AMD—ont eux aussi compris l'intérêt qu'il y avait à développer les possibilités de virtualisation. En effet, les processeurs utilisés sur les architectures PC 3 sont notoirement difficiles à émuler de manière performante, du fait du nombre d'instructions nécessitant un niveau de privilège élevé (donc hors d'atteinte d'une machine virtuelle s'exécutant comme un processus utilisateur), ainsi que de la façon dont la mémoire est gérée.

Toutefois, Intel et AMD vont encore plus loin, car récemment, ils ont tous les deux mis au point des technologies pour décharger encore plus la machine virtuelle, avec l'introduction de la virtualisation pour les E/S.

La société Qumranet a diffusé il y a peu le logiciel KVM, qui est basé sur QEMU pour la machine virtuelle. Puis Le tour de force de Qumranet a été de faire intégrer KVM au noyau Linux, lui assurant ainsi une grande diffusion et un fort succès auprès de la communauté.

Tous ces investissements et acquisitions en l'espace de quelques mois montrent qu'il y a un réel mouvement de fond autour de la virtualisation. Toutes les entreprises veulent en faire partie et proposer une solution à leurs clients. Le marché de la virtualisation est en pleine croissance, et cela ne fait que commencer. [2.2]

1.3 Concept de la virtualisation

La virtualisation n'est pas un concept inventé récemment, mais a fait ses premières apparitions il y a quarante ans avec les gros systèmes d'IBM. En réalité, nous utilisons déjà aujourd'hui plusieurs techniques de virtualisation dans nos serveurs x86 telles que les RAID capables d'agréger plusieurs disques physiques en un disque logique (ou virtuel). Dans le monde du réseau, on parlera de VPN pour Réseau Virtuel Privé, de VLAN pour Lan Virtuel.

On pourrait définir la virtualisation des serveurs comme un ensemble de Techniques permettant de faire fonctionner de manière simultanée plusieurs serveurs logiques sur un même serveur physique. L'objectif de la virtualisation des serveurs est de permettre à un serveur logique de pouvoir accéder nativement aux ressources matérielles comme s'il était seul à utiliser ces ressources.

Le principe de la virtualisation est un principe de *partage* : les différents systèmes d'exploitation se partagent les ressources du serveur.
Pour être utile de manière opérationnelle, la virtualisation doit respecter deux principes fondamentaux :

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

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

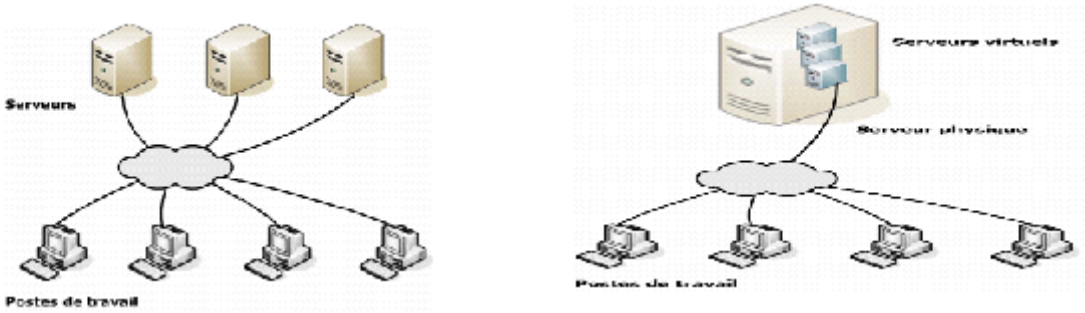


Figure 1 : Architecture traditionnelle vs Architecture virtualisé

En fait, les applications tournant sur un même serveur, en l'absence de virtualisation, se partagent déjà les ressources du serveur. C'est l'une des missions du système d'exploitation que de permettre et d'administrer ce partage : plusieurs applications se partagent les disques, le processeur, la mémoire, les accès réseau, et le système d'exploitation est le chef d'orchestre, gérant les règles de ce partage. [1.5]

Alors, pourquoi ce partage ne suffit-il pas ? Pourquoi a-t-on besoin de virtualisation A cela, deux réponses.

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

Il arrive couramment que les applications requièrent un système d'exploitation particulier, ou bien une configuration particulière du système, ou encore des composants logiciels majeurs qui ne peuvent pas cohabiter sur un même système d'exploitation. Dans tous ces cas de figure, le partage de ressources offert par le système lui même ne convient plus : on veut partager les ressources *en dessous* du système d'exploitation, de manière à faire cohabiter plusieurs systèmes d'exploitation sur le même serveur physique.

1.4 Intérêts et inconvénients de la virtualisation

Intérêts

❖ Un système d'exploitation virtualisé permet d'observer le comportement d'un logiciel malveillant *(malware*)* — virus, ver, *spyware*, etc. — dans un système sain sans avoir à infecter une machine physique. De plus, le processus d'infection est reproductible.

❖ Les technologies de virtualisation permettent de séparer des applications et des systèmes de manière logique, quand les prérequis des applications sont mutuellement exclusives.

❖ La virtualisation de plusieurs systèmes d'exploitation permettra aux développeurs de tester le rendu de plusieurs navigateurs sur plusieurs plates-formes sans avoir à changer de machine — et donc d'environnement de travail — en permanence.

❖ Toutefois, cette dispersion a un coût qui n'est pas nul pour l'entreprise, que ce soit en espace occupé (location au mètre carré dans les *Datacenter**), en énergie (consommation électrique) ou en maintenance (plus de machines physiques implique plus de risques de pannes matérielles).

❖ la plupart des services fournis sur un réseau local (DHCP, DNS, Intranet, ...) ne consomment qu'une très faible partie des ressources offertes par une machine récente. Tous ces facteurs font qu'il n'est plus pertinent aujourd'hui d'utiliser des machines séparées pour héberger des services ne nécessitant qu'une fraction de la puissance d'une machine.

❖ La virtualisation peut apporter beaucoup en termes de réactivité et de flexibilité.

❖ L'importance parfois critique des applications « anciennes » pour le fonctionnement de l'entreprise fait qu'il est souvent plus facile de continuer à maintenir un système et une machine obsolètes (et donc avec un risque de panne matérielle plus important) que d'entamer une migration vers une nouvelle plate-forme. La virtualisation permet dans ce cas d'exécuter l'application comme dans son environnement d'origine, mais sur du matériel récent (une application de comptabilité ou un progiciel quelconque) utilisée depuis des années mais non portée sur la nouvelle version d'un système d'exploitation ou encore un logiciel de pilotage de machine industrielle. [1.1]

Inconvénients

Performances et rendement

A l'évidence, puisqu'il y a partage des ressources physiques, chaque environnement virtuel dispose de ressources plus limitées que s'il avait un serveur physique dédié. Le problème qui s'impose dans ce cas c'est le surcoût (*« overhead »*) en termes de CPU, car les autres ressources sont en général moins précieuses.

En effet, ce problème est rencontré dans toutes les virtualisations reposant sur des Processeurs qui ne disposent pas d'instructions spécialisées. C'est dernières permettent un surcoût en performances qui est aujourd'hui négligeable. (On peut éviter ce problème facilement)

Administration

La mise en œuvre et l'exploitation des solutions de virtualisation requièrent une vraie expertise, en plus les solutions open source de virtualisation ont un packaging moins abouti, et ne fournissent pas d'outils graphiques aussi avancés que leurs concurrents propriétaires.

Mais même si l'apprentissage des outils d'administration en ligne de commande nécessite un certain niveau de formation, ils permettent une maîtrise plus importante et plus grande souplesse d'utilisation.

Contrôle des ressources

Une des grandes problématiques dans un environnement virtualisé est le contrôle dans l'attribution et dans le partage des ressources du serveur physique. Il arrive qu'un serveur ne pénalise les autres en consommant toute les ressources de la machine physique sur laquelle ils s'exécutent.

On peut souhaiter répartir les ressources disponibles soit de façon équitable, soit en privilégiant certains environnements par rapport aux autres.
Les règles dépendent bien sûr du domaine d'application. Si 10 sites Internet se partagent un serveur physique et que l'un connaît un pic de trafic, on peut souhaiter lui laisser prendre 90% de la CPU tant que les autres n'en ont pas usage. A l'inverse, si un hébergeur a vendu 1/10ème de serveur à l'un de ses clients, il doit être en mesure de garantir que le client aura toujours son quota, quelle que soit la demande des autres clients. [1.2]

1.5 Les catégories de la virtualisation

On distingue trois grandes catégories de solutions de virtualisation :

1.5.1 Isolation

Présentation

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

L'isolation est utilisée sous Unix depuis longtemps pour protéger les systèmes. Via des mécanismes comme *chroot* ou *jail* il est possible d'exécuter des applications dans un environnement qui n'est pas celui du système hôte, mais un « mini système » ne contenant que ce dont l'application a besoin, et n'ayant que des accès limités aux ressources. Il est possible également de lancer des programmes dans une autre distribution que celle du système principal.

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

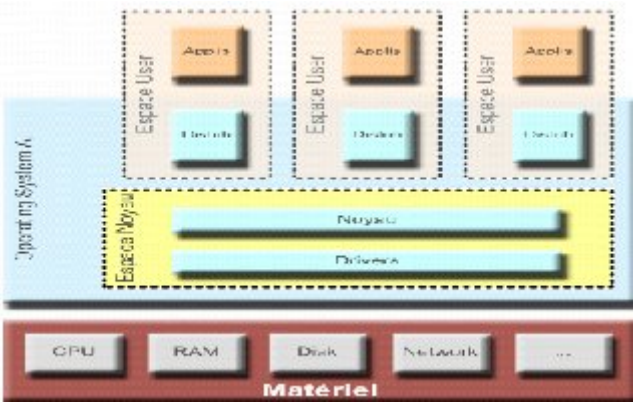


Figure 2 : schématisation de l'isolation

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.

1.5.2 Paravirtualisation

1.5.2.1 Hyperviseur

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

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

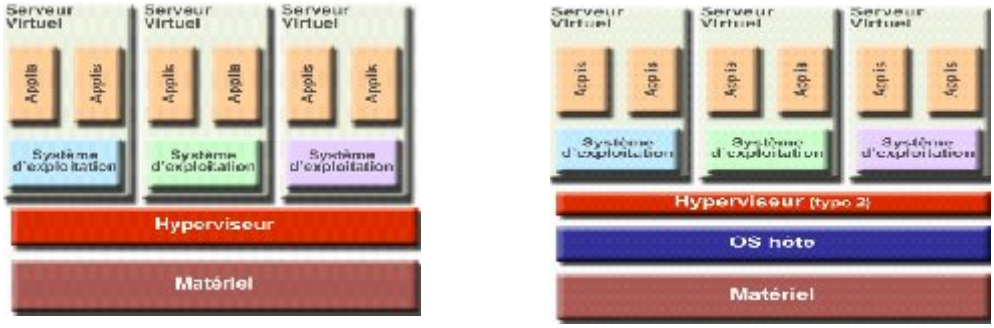


Figure3 : emplacement de l'hyperviseur

1.5.2.2 Présentation de la paravirtualisation

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

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

Chaque système virtuel doit donc être modifié de façon à utiliser cette interface pour accéder au matériel, en revanche, contrairement à l'isolation, plusieurs OS de familles différentes peuvent fonctionner sur un même serveur physique. Il est ainsi possible de faire fonctionner GNU/Linux, NetWare, Solaris (et d'autres) simultanément sur une même machine.

La nécessité de petites modifications au système d'exploitation invité exclut le support de systèmes « fermés », et en particulier de Microsoft Windows. [1. 3]

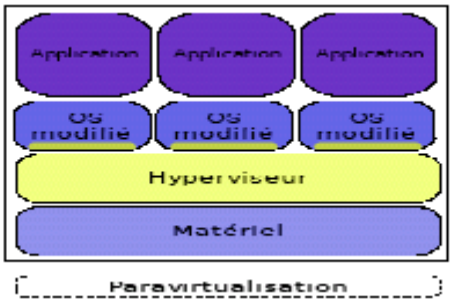


Figure 4: schématisation de la paravirtualisation

Environnements utilisées

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

1.5.3 Virtualisation complète

Présentation

Au sens strict, le terme 'machines virtuelles' désigne les systèmes virtuels exécutés via des technologies de virtualisation dites *complète* ou encore *native*.
Dans ce cas de figure, c'est le matériel d'un ordinateur complet qui est présenté au système d'exploitation par le produit, de sorte que la virtualisation est alors réellement transparente pour le système d'exploitation invité.

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

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

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

Les produits modernes tirent partie des nouveaux jeux d'instructions spécialisés des dernières générations de processeurs pour assurer des performances quasi identiques aux performances natives, alors que l'interface matérielle est gérée au plus bas niveau par un hyperviseur. Sur une machine virtuelle, il est possible d'installer n'importe quel OS non modifié, et donc aussi bien commercial qu'open source du moment qu'il dispose des pilotes pour le matériel générique que lui présente l'hyperviseur.[1.3]

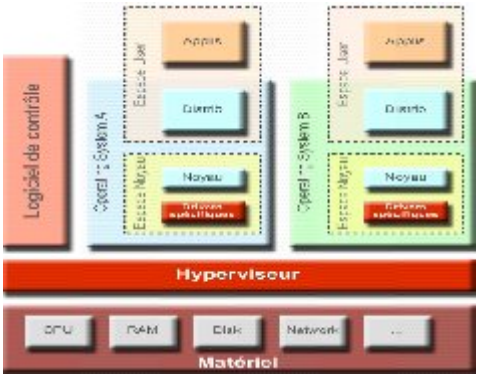


Figure 5 : schématisation de la virtualisation complète

Les principaux exemples de ce principe sont
KVM : Intégré au noyau GNU/Linux à partir de la version 2.6.20.
Xen sur une machine offrant le support des instructions AM-V. (voir les détailles dans le chapitre 2)

1.5.4 Emulation

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

L'émulation est la technique qui offre le plus haut niveau d'abstraction de la plateforme. Il faut rappeler en effet que toutes les autres techniques de virtualisation citées ont une exigence en commun : tous les exécutables doivent être compilés pour le processeur physiquement disponible sur le serveur. Elle lève cette contrainte car les instructions ne sont jamais exécutées par le processeur, elles sont interprétées en simulant le processeur. [1.1] – [1.2]

1.6 Différents solutions de virtualisation

Nous allons présenter ici les différents systèmes connus dans la virtualisation en mettant en évidence leurs principaux avantages et inconvénients.

1.6.1 QEMU

Présentation

QEMU est le projet de virtualisation complète libre le plus abouti actuellement. Il a été fondé par Fabrice BELLARD et diffusé à la communauté en 2003. Fabrice BELLARD est également connu pour avoir fondé le projet FFmpeg, une suite d'utilitaires dédiés au traitement et à la conversion de flux numériques (vidéo et audio). À l'heure actuelle, M. BELLARD est toujours le mainteneur et développeur principal de QEMU, secondé par une communauté active.

Technologies

Techniquement, QEMU utilise la virtualisation complète, et supporte de nombreuses architectures cibles (*i.e.* émulées), parmi lesquelles les processeurs x86 et l'architecture PC. Il fonctionne sur les plates-formes les plus courantes (Microsoft Windows, GNU/Linux, Mac OS X) et est très simple d'utilisation

QEMU peut en option utiliser un module d'accélération système pour améliorer les performances. Ce module s'appelle kqemu *(kernel-QEMU)* et est disponible pour Windows et GNU/Linux. Il s'intègre au noyau (ou dans les services dans le cas de Windows) et permet à QEMU de contourner certaines couches d'abstraction du système hôte, amenant ainsi un gain de performances.

Virtualisation Complète	✓
Paravirtualisation	
Hyperviseur	
Cloisonnement	

Tableau 1 : Qemu

Fonctionnalités

Ce qui distingue QEMU de solutions plus intrusives (comme Linux-VServer ou Xen) est sa grande simplicité d'usage. En effet, comme c'est un projet utilisant la virtualisation complète, il y a un simple programme à exécuter sur le système hôte pour obtenir une nouvelle machine virtuelle contenant un système invité.

Un des gros points forts de QEMU est la flexibilité qu'il offre au niveau des options pour la configuration de la machine virtuelle. La section suivante détaillera la gestion des interfaces réseau et du trafic réseau en général, car c'est un des points forts de QEMU. L'étendue de ses possibilités dans ce domaine a été l'un des critères pour sa sélection dans ce comparatif.
QEMU est diffusé sous la licence GNU GPL

Inconvénients

Tout d'abord, du fait de la grande complexité du code réalisant l'émulation du processeur, il sera difficile de comprendre et modifier les sources du programme, si jamais le besoin se fait sentir. L'évolution du projet est donc aux mains des personnes à même de comprendre l'architecture bas niveau de QEMU.

Les performances de la machine virtuelle sont un autre facteur limitatif. Comme toutes les solutions de virtualisation complète, QEMU pêche au niveau de la rapidité des Entrées/Sorties. Le module d'accélération kqemu améliore certes les performances, mais on reste loin des performances « natives ». Il faut toutefois signaler que même si les performances sont éloignées de celles d'une véritable machine, QEMU se place parmi les solutions de virtualisation complète les plus performantes.[1.3]

1.6.2 KVM

Présentation

Le projet KVM (*Kernel-based Virtual Machine* – machine virtuelle dans le noyau) a été créé en 2006 par la société Qumranet. KVM a su très vite attirer les contributions externes et le code source du projet a été intégré au noyau Linux dès février 2007. Le mainteneur principal de KVM est Avi KIVITY. À l'heure actuelle, le projet KVM est financé principalement par Qumranet, mais IBM participe aussi au développement en salariant un développeur pour travailler sur KVM.

Technologies

KVM est un projet de virtualisation complète qui utilise les instructions de virtualisation des processeurs x86 récents. Techniquement, KVM se compose d'un module noyau et un programme utilisateur, qui utilise le module noyau pour toutes les opérations privilégiées.

La partie utilisateur de KVM est une version légèrement modifiée de QEMU, simplement adaptée pour que les opérations pouvant bénéficié des instructions de virtualisation du processeur fassent appel au module noyau.

Virtualisation Complète	
Paravirtualisation	✓
Hyperviseur	
Cloisonnement	

Tableau 2 : KVM

Fonctionnalités

KVM étant basé sur QEMU, il reprend naturellement toutes les fonctionnalités de ce dernier, il y a toutefois une différence notable : grâce à l'utilisation des instructions de virtualisation des processeurs récents, les performances des systèmes invités sont bien plus élevées qu'avec QEMU. KVM est majoritairement sous licence GNU GPL.

Inconvénients

Étant basé sur QEMU, KVM souffre des mêmes inconvénients sur la complexité du code et la difficulté de manipuler les images disques. En plus de cela, la partie propre à KVM, c'est à dire le module noyau utilisant les instructions de virtualisation, est encore jeune et manque de maturité. La stabilité de l'interface de programmation *(API)* vient tout juste d'être atteinte, mais il n'est pas exclu que l'*API* change à nouveau. Cela signifie qu'un outil très proche de KVM peut ne plus marcher après une mise à jour.

1.6.3 Linux-VServer

Présentation

Le projet Linux-VServer est un projet de virtualisation par cloisonnement pour la plateforme GNU/Linux. Au départ, ce projet s'appelait *Virtual private servers and security contexts* et était géré par la société Solucorp. Il a été ouvert à la communauté fin 2001, et a eu une évolution lente mais constante depuis. La version actuelle est la version 2.2.0.3, qui a été diffusée au début du mois de juillet 2007.

Technologies

Le projet Linux-VServer utilise le cloisonnement, c'est à dire qu'il isole des instances du système d'exploitation par dessus un système hôte.

Virtualisation Complète	
Paravirtualisation	
Hyperviseur	
Cloisonnement	✓

TABLEAU 3 : LINUX- VSERVER

Du point de vue technique, il se compose de *patches* à appliquer sur le code source du noyau Linux et d'utilitaires fonctionnant dans l'espace utilisateur pour contrôler la création et l'administration des systèmes invités.

Linux-VServer se base sur les fonctionnalités déjà implémentées au sein du noyau pour la séparation des processus et les droits d'accès pour implémenter une séparation complète au niveau du noyau. Les modifications apportées utilisent et étendent les fonctionnalités déjà présentes dans le noyau. Il y a un seul noyau Linux, qui contrôle toujours l'accès au matériel

Fonctionnalités

Comme le projet utilise une technologie de cloisonnement, il n'y a quasiment pas de perte de performances par rapport à un système natif, mais il y un net gain en sécurité et simplicité d'administration du fait de la séparation entre les systèmes invités.

Contrairement à QEMU et KVM, Linux-VServer ne travaille pas avec une image disque du système, mais avec un simple ensemble de fichiers dans un répertoire, qui représentera la racine du système invité. Cela implique une plus grande facilité d'utilisation.

Inconvénients

Malgré des performances très élevées, le projet Linux-VServer souffre de quelques inconvénients. Tout d'abord, la séparation entre les systèmes invités est trop « faible », dans le sens où il y a un seul noyau Linux pour plusieurs systèmes invités. Si un système invité trouve un moyen de contourner les protections et affecte le noyau, tous les autres systèmes de la machine en pâtiront. C'est une faiblesse dont souffrent toutes les solutions de virtualisation utilisant le cloisonnement. [1.3]

1.6.4 OpenVZ

Présentation

OpenVZ est un projet de virtualisation par cloisonnement géré par la société SWsoft.
La société SWsoft a été créée aux États-Unis en 1997. En 2001, le produit Virtuozzo est diffusé, avec son pendant libre : OpenVZ. SWsoft rachète Parallels en 2004, une société spécialisée dans la virtualisation pour les particuliers, très populaire sur Mac OS X.

Technologies

Le principe de fonctionnement d'OpenVZ est très similaire à celui de Linux-VServer, car ils se basent tous les deux sur une modification du noyau Linux pour implémenter un système de cloisonnement au niveau du système d'exploitation.

Virtualisation Complète	
Paravirtualisation	
Hyperviseur	
Cloisonnement	✓

TABLEAU 4 : OPENVZ

OpenVZ modifie le noyau Linux plus en profondeur que le *patch* de Linux-VServer. Il y a donc des fonctionnalités spécifiques à OpenVZ. Parmi ces nouvelles fonctionnalités, on peut notamment citer l'ajout d'un niveau supplémentaire d'indirection pour les ordonnanceurs du noyau.

Ce nouveau niveau d'indirection permet de gérer les priorités entre les systèmes invités, puis ensuite au sein d'un système invité. C'est à dire que l'ordonnanceur de bas niveau décide d'abord à quel système invité passer la main, puis ensuite l'ordonnanceur traditionnel du noyau Linux décide à quel processus passer la main, au sein de ce système. La répartition de charge entre les systèmes invités est donc plus fine qu'avec Linux-VServer, qui utilise un système d'ordonnancement bien moins complexe qu'OpenVZ.

Fonctionnalités

Les fonctionnalités de haut niveau d'OpenVZ sont quasiment identiques à celles de Linux-VServer : arrêt, instanciation et contrôle des systèmes invités sont d'un fonctionnement similaire. Seule une étude approfondie permet de noter quelques différences, qui viennent pour la plupart d'une intégration plus en profondeur. Le projet OpenVZ est distribué sous la licence GNU GPL.

Inconvénients

Le principal problème avec OpenVZ est que SWsoft ne « joue pas le jeu» du logiciel libre. En effet, le projet libre dispose de moins de fonctionnalités que la version propriétaire et payante. La version payante est montrée comme la version complète, alors que la version libre est juste là pour appâter la communauté et attirer les contributions. [1.2]

1.6.5 VirtualBox et VMware :

Ces deux inclus avec les solutions de la virtualisation complète, et grâce à leurs importances dans le monde de la virtualisation on va les étudier dans le chapitre 3 en détail.

1.6.6 Xen

Présentation

Xen (dont le nom vient du grec *xenos*, étranger) est un projet de virtualisation par hyperviseur géré par la société XenSource. Le projet était à l'origine mené au sein de l'Université de Cambridge, sous le nom de Xenoserver. Le but était alors d'héberger 100 systèmes invités sur une seule machine physique, avec les meilleures performances possibles.
En 2003, les initiateurs du projet ont fondé la société XenSource et ont lancé le projet Xen en se basant sur le code source de Xenoserver.

La solution de virtualisation Xen est séparée en plusieurs produits, ayant tous des finalités différentes. Il y a tout d'abord la version libre, nommée Xen 3.0, qui concentre toute la technologie de virtualisation. Les autres versions (propriétaires) de la gamme se distinguent uniquement par le support proposé, les nombres de machines virtuelles supportées, les systèmes

invités supportés et les logiciels annexes. Ainsi, la version Xen Enterprise se base intégralement sur Xen 3.0, mais rajoute des outils de contrôle pour gérer plusieurs dizaines d'instances de Xen.

Technologies

Xen est un hyperviseur, c'est à dire qu'il vient s'insérer entre le matériel et le noyau. C'est donc Xen qui a l'accès exclusif au matériel, et les systèmes d'exploitation fonctionnant par dessus doivent obligatoirement passer par l'hyperviseur pour y accéder. Le projet Xen est diffusé sous la licence GNU GPL. [1.4]

Virtualisation Complète	
Paravirtualisation	✓
Hyperviseur	✓
Cloisonnement	

TABLEAU 5 : XEN

On va parler de xen avec plus de détaille dans le chapitre 4.

1.7 Conclusion

La virtualisation est un domaine en pleine croissance, il évolue très rapidement. Les entreprises peuvent se servir de la virtualisation pour différents usages, aux besoins variés.
Dans ce chapitre on a parlé d'une façon générale sur la virtualisation en tant que concept et intérêts.

Le contenu de ce chapitre est très important pour bien comprendre la virtualisation en tant que solution majeure pour le déploiement des multiples serveurs d'un coté, et d'un autre coté comme un nouveau concept dans le monde de l'informatique.

En plus des possibilités techniques déjà citées, la virtualisation est également une technologie clef pour l'avenir de l'entreprise. En effet, elle ne permet pas seulement de contourner les limitations matérielles des ordinateurs, mais elle peut aussi fournir un avantage décisif sur la concurrence dans le milieu très disputé qu'est l'informatique de services.
Dans ce qui suit, on va parler d'une façon approfondie des solutions de virtualisation.

CHAPITRE 2

Types d'implémentation de la virtualisation

2.1 Introduction

Dans ce chapitre, on va parler brièvement des deux types majeurs de virtualisation, tels que la virtualisation logicielle et la solution développée par Intel et AMD qui consiste en la virtualisation matérielle.

Dans un premier temps, on va justifier le passage de l'implémentation logicielle à celle matérielle ainsi que les inconvénients et les avantages de chaque solution tout au long de notre étude.

2.2 Implémentations logicielles:

On remarque deux grands types d'implémentations logicielles :

2.2.1 Implémentation native

L'hyperviseur (VMM) se met juste sur la couche Hardware.

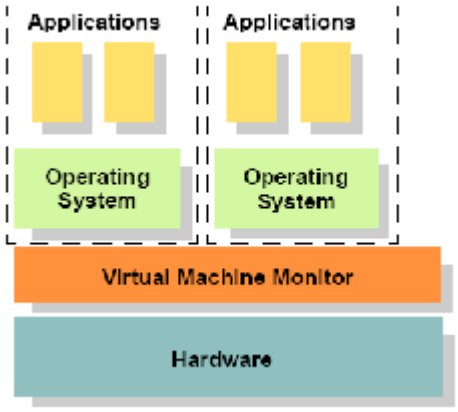


Figure 6 : Implémentation native

Intérêt majeur c'est la rapidité d'exécution vs Inconvénient difficulté de mise en place

2.2.2 Implémentation hébergée

Ou l'hyperviseur est Hébergé sur un Système d'exploitation Hôte.

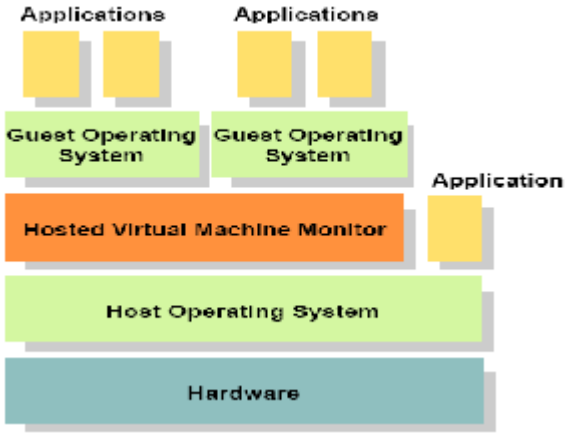


Figure 7: Implémentation Hébergée.

L'intérêt se résume dans la facilité de mise au point ***vs*** Inconvénient c'est la perte du temps d'exécution. Et pour chaque type d'implémentation nous pouvons distinguer ces trois cas :

1. **l'isolation d'application,**
2. **la virtualisation complète (et par extension, l'émulation),**
3. **la para-virtualisation.**

Nous allons présenter et expliquer les concepts qui sont derrière chaque technique de virtualisation, parce que les maîtriser, permettra à tout un chacun de savoir et de comprendre de quelle virtualisation il a besoin et il peut vouloir ; plutôt que de se laisser imposer un système qui ne lui conviendrait pas obligatoirement par des gens qui n'ont qu'une vue parcellaire et étriquée, biaisée, ou intéressée de ce domaine de l'informatique.

2.2.3 Concepts

2.2.3.1 L'isolation

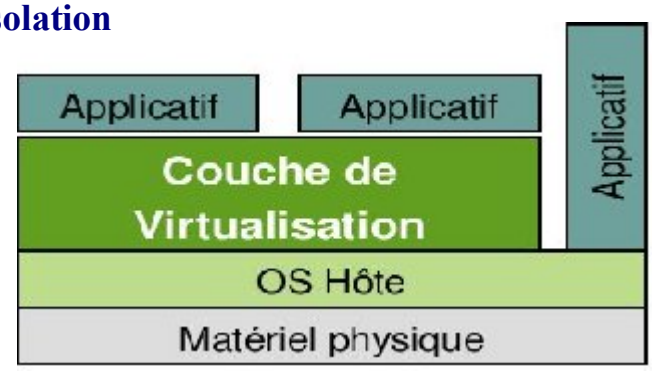


Figure 8. Isolation

Un isolateur est un logiciel permettant d'isoler l'exécution des applications dans des contextes ou zones d'exécution. L'isolateur permet ainsi de faire tourner plusieurs fois la même application prévue pour ne tourner qu'à une seule instance par machine.

2.2.3.2 La virtualisation complète

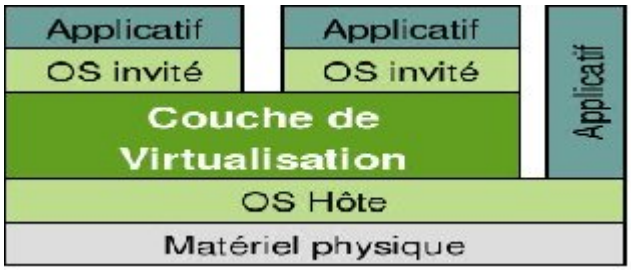


Figure 9 : Virtualisation complète

Ici, c'est simple : on virtualise le « matériel » donc on peut prendre le CD fourni par le distributeur (Microsoft, RedHat, Novell, etc…), le mettre dans le lecteur de cdrom (ou en faire une image ISO), et on installe son système « comme si » on était sur une machine physique.

Ce concept apparu plus « récemment » est finalement le plus « évident » à concevoir (mais pas forcément le plus simple à implémenter). L'avantage énorme de ce système de virtualisation est évident : on ne modifie rien à son CD d'installation, tout doit être vu par le système comme s'il était sur une machine physique. L'inconvénient majeur c'est une perte de performance assez phénoménale (minimum 20%) du fait même du concept utilisé : on va analyser tous les appels passés par le système d'exploitation au « matériel » qu'il croit être réel. [2.2]

2.2.3.3 La Paravirtualisation

Figure 10 : Para-virtualisation

Là où c'est devenu compliqué c'est que le succès et les améliorations formidables des plates-formes x86 aidant, ce matériel est devenu de plus en plus commun dans les grands centres informatiques, au point de se rendre majoritaire et indispensable dans toutes les DSI. Et quelque chose qui était possible et même commun avec les vrais grands processeurs professionnels de l'époque (le partage de temps entre les systèmes d'exploitation), était impossible avec ce matériel là, alors que sa puissance le permettait.

C'est le concept de la para-virtualisation : on développe une couche minuscule de virtualisation qui sera directement posée sur le matériel (un hyperviseur), et on modifie le système d'exploitation invité pour le rendre « virtualisable ».

L'avantage fondamental de cette technique c'est la puissance. Il y a une perte infime en matière de performances et l'inconvénient c'est qu'il faut que le système d'exploitation soit modifié : exit donc Microsoft qui ne fournit pas de version « para-virtualisée » de son système d'exploitation. [2.1]

2.2.4 Architecture de l'hyperviseur Xen

Nous allons parler de l'architecture de l'hyperviseur Xen d'une manière générale.

Composants de Xen

L'environnement de la virilisation Xen se compose de plusieurs éléments qui assurent le bon déroulement de la virtualisation:

- L'hyperviseur Xen
- Domaine 0
- Domaine de Management et de contrôle (Xen DM&C)
- Domaine U (Dom U) Para Virtualisation invité (PV Guest)
- Domaine U (Dom U) Virtualisation Totale invité (HVM Guest).

Le diagramme suivant montre l'organisation classique de ces composants

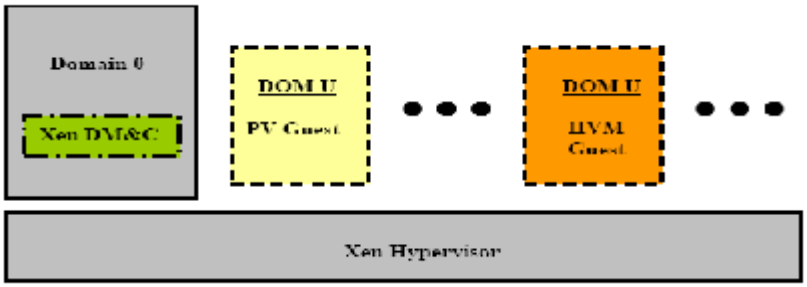


Figure 11: Organisation classique de Xen

2.2.5 L'Hyperviseur Xen

L'hyperviseur Xen est l'abstraction de la couche software qui se met directement après la couche physique, il organise le fonctionnement de la CPU ainsi que le partage de la mémoire des différentes machines virtuelles. L'hyperviseur ne se contente pas uniquement de l'abstraction Hardware des machines virtuelles mais il contrôle et veille sur le bon fonctionnement de ces dernières dans cet environnement partagé.

Il n'a aucune connaissance du réseau ni des appareils de stockages externe, ni des fonctions entrée/ sortie de l'ordinateur.

Domaine 0 : Il s'agit d'un système d'exploitation linux dont le noyau a été modifié, c'est l'unique machine virtuelle se déployant sur l'hyperviseur Xen qui bénéficié de certaine autorisation d'accès physique des ressources entrées / sorties lui permettant d'interagir avec les autres domaines invités (Domain U: PV and HVM Guests).

La présence du domaine 0 est indispensable pour l'environnement de virtualisation Xen car il permet aux autres machines virtuelles de démarrer et fonctionner correctement.

On remarque deux drivers dans le Domaine 0 qui sont Le Network backend Driver et le Block Backend Driver. L'un pour l'accès au réseau et l'autre pour l'accès au disk dur des Domaines (Domain U PV et HVM invité). Le Network backend Driver communique directement avec le réseau physique local permettant ainsi aux autres domaines d'accéder au réseau, pareil pour le Block Backend Driver qui lui a accès au disck locale physique.

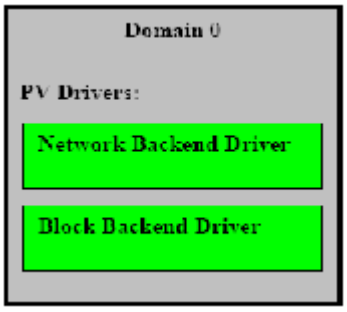


Figure 12 : PV Drivers Domain 0

Domaine U: Les Domaines U se composent de deux catégories, Systèmes d'exploitation para visualisés **Domain U PV invité** dont le noyau subit des modifications citant Linux modifé, Solaris, FreeBSD et les systemes d'exploitation totalement virtualisés **Domain U HVM invité** dont le noyau du système d'exploitation ne subit aucun changement comme Windows…

Les Domaines para virtualisés **Domain U PV Guests** ont conscience qu'ils n'accèdent pas directement à la couche physique, ils peuvent même détecter la présence des autres domaines Tandisque les Domaines Totalement virtualisés n'ont aucune conscience des autres domaines ils se croient totalement seule dans la machine

Les Domaines para Virtualisés contiennent deux types de Drivers

- PV Network driver
- PV Block driver

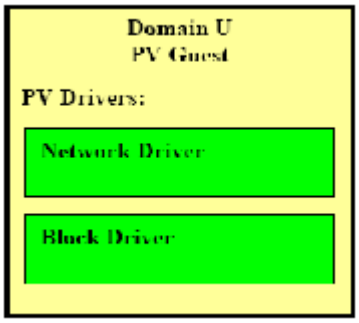


Figure 13: PV Drivers Domin U

Les domaines totalement virtualisés **Domain U HVM Guest,** ne possèdent pas de PV Drivers a la place un logiciel spéciale (Deamon) se situant dans le domaine 0 le **Qemu-dm** il supporte les requêtes des **domaines U HVM** pour avoir accès au réseau et au disk dur.
Pour avoir un démarrage sans reproche pour les domaines totalement virtualisés un software a été ajouté dans chacun d'eux pour jouer le rôle du BIOS au démarrage.

Le ***Xen virtual firmware***

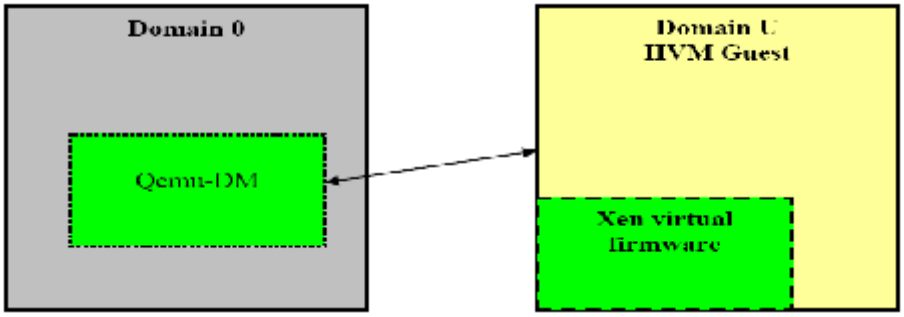


Figure 14 : Xen virtual firmware

2.2.6 Domaine de Management et de Contrôle (*Domain Management and Control):*

Des séries de distributions linux (deamons) ont été classées par la communauté Open source comme **domaine de Management et de contrôle.** Ces services supportent tout le Contrôle et le Management des environnements virtuels et se situent dans le domaine 0 Le diagramme ci-dessous nous les montres séparées pour avoir une meilleure compréhension de l'architecture.

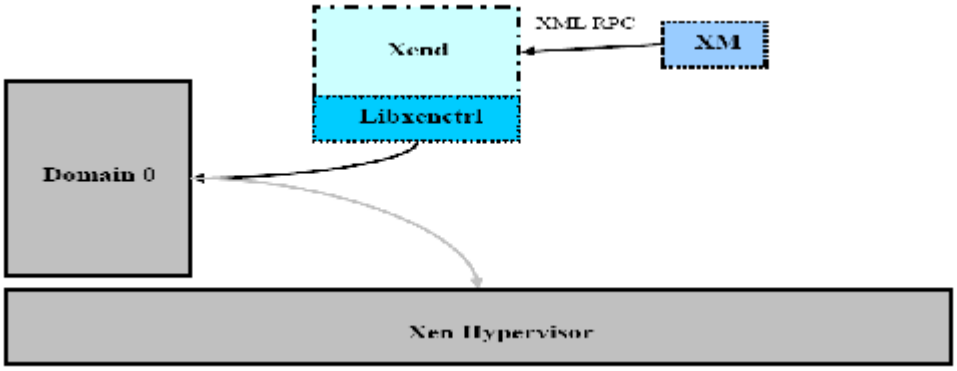


Figure 15 : Domaine de Management et de contrôle

XM: c'est la ligne de commande qui permet à l'utilisateur d'accéder au **Xend** par le billet du **XML RPC**

Xenstored: maintien un registre d'information incluant la mémoire et le canal de transfert entre le domaine 0 et tous les autres domaines invités.

Libxenctrl: C'est une bibliothèque de **C** qui donne au **Xend** la capacité de communiquer avec l'hyperviseur **Xen** via le domaine 0. Sans oublier le Driver **privcmd** se situant dans le domaine 0 qui délivre les requêtes à l'hyperviseur.

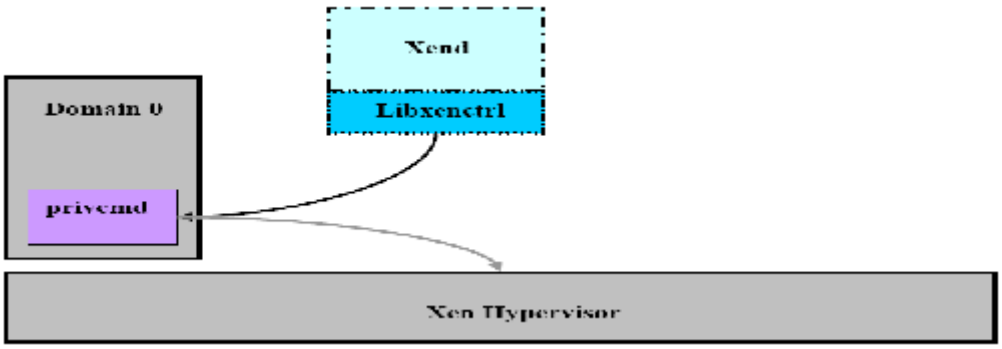


Figure 16 : Communication Xend Domain 0

Qemu-dm: Chaque domaine totalement virtualisé (**Domain U HVM**) requière son propre **Qemu deamon**. Cet outil s'occupe des requêtes de réseau et de disk dur reçu par les domaines **U HVM**. Le **Qemu** doit exister en dehors de l'hyperviseur **Xen** et cela pour qu'il y ait accès au réseau et au entrées /sorties (**I/O**).

Xen Virtual Firmware : C'est un BIOS virtuel injecté dans chaque **domaine U HVM** ayant pour but d'assurer au système d'exploitation tous les éléments de Boot dont 'il a besoin pour un fonctionnement correcte.

2.2.7 Xen Opération

2.2.7.1 Communication Domaine 0 et domaine U

L'hyperviseur Xen n'a pas été conçu pour gérer les requêtes du réseau et du disk.
C'est pour cela que les Domaine U PV doivent communique via L'hyperviseur Xen par le billet du domaine 0. Pour accomplir les requêtes des réseaux et du stockage.
L'exemple suivant présente comment un domaine U PV invité arrive à écrire sur le disk.

- Le Domaine U PV invité envoie la requette au PV Block Driver pour ecrire sur le Disque.
- Le PV Block Driver écrit cette information sur la mémoire avec l'assistance de l'hyperviseur Xen. Cette dernière est partagée avec le domaine 0.
- Il y a un canal de transmission qui permet la communication entre le domaine 0. et le domaine U PV invité asynchrone.
- Le Domaine 0 reçoit une interruption de l'hyperviseur Xen permettant au PV Block Backend Driver d'accéder à la mémoire partagée et d'écrire son contenue sur le disque dur dans l'emplacement approprié. [2.2]

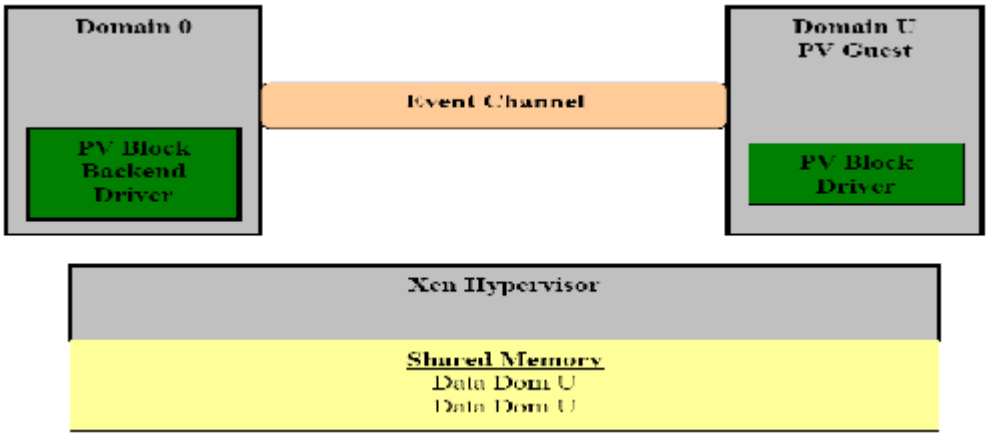


Figure 17: Communication domain 0 domain U

2.3 Architecture X86 et les anneaux de protection (ring)

2.3.1 Les Ring

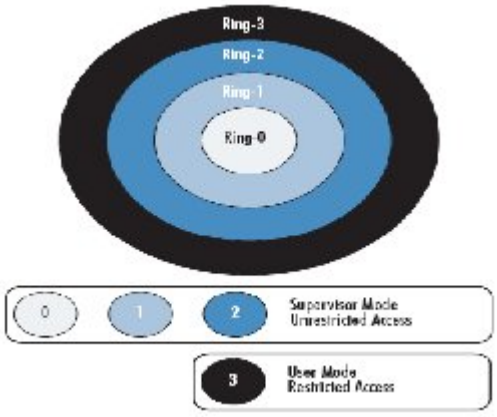


Figure 18 : représentation des rings en anneaux

Un **anneau de protection** (ou **ring**) est l'un des niveaux de privilèges imposé par l'architecture d'un processeur.

De nombreuses architectures modernes de processeurs (architectures parmi lesquelles on trouve le populaire Intel x86) incluent une certaine forme de protection en anneau, bien que les logiciels d'exploitation ne l'exploitent pas toujours entièrement.

Les *rings* étaient parmi les concepts les plus révolutionnaires mis en œuvre par le système d'exploitation Multics, un prédécesseur fortement sécurisé de la famille actuelle des systèmes d'exploitation UNIX.

Description

Les anneaux sont arrangés dans une hiérarchie allant du plus privilégié (celui qui est le plus sécurisé, habituellement le numéro zéro dit *Ring0*) au moins privilégié (le moins sécurisé, habituellement l'anneau le plus élevé). Le système d'exploitation Multics original possédait huit anneaux, mais beaucoup de systèmes modernes en possèdent moins. Le matériel connait l'anneau de privilège courant des instructions qui s'exécutent à tout moment, grâce à des registres spéciaux de la machine.

Le matériel limite sévèrement les manières dont la main peut être passée d'un anneau à l'autre, et impose également des restrictions aux types d'accès mémoire qui peuvent être exécutés aux travers des anneaux. En général il y a une instruction spéciale d'appel qui transfère le contrôle d'une manière sécurisée vers des points d'entrée prédéfinis dans des anneaux de plus bas niveaux (plus sécurisés) ; ceci fonctionne comme un appel supervisé dans beaucoup de systèmes d'exploitation qui emploient l'architecture en anneau, ces restrictions matérielles étant conçues pour limiter les occasions d'infractions accidentelles ou malveillantes envers la sécurité du système.

2.4 Architecture x86

C'est la dénomination de la famille de microprocesseurs compatibles avec le jeu d'instructions de l'Intel 8086. Les différents constructeurs de microprocesseurs pour PC se doivent de maintenir une compatibilité ascendante afin que les anciens logiciels fonctionnent sur

les nouveaux microprocesseurs. L'architecture de la série x86 à partir du Pentium a été nommée IA-32 par Intel.

À l'origine de conception CISC, les nouvelles générations ont été de plus en plus conçues comme des processeurs RISC, les instructions complexes étant transformées dans le microprocesseur en instructions plus élémentaires. Cette famille de processeurs, dont le Pentium est emblématique, est en train de passer au 64 bit.

Architecture

La conception de la gamme x86 a mis l'accent sur la compatibilité ascendante. Ainsi, les générations successives de processeurs admettent plusieurs modes de fonctionnement, qui sont diffèrent en particulier du point de vue de l'accès à la mémoire. [2.13]

2.5 Virtualisation matérielle

2.5.1 Introduction

L'environnement d'exploitation x86 n'a jamais été conçu pour être un environnement partagé, De sorte qu'il est difficile de créer un hyperviseur optimisé qui peut efficacement Allouer des ressources matérielles et gérer plusieurs machines virtuelles et leur Systèmes d'exploitation.

La solution à ce type de scénario est de lancer l'hyperviseur dans le ring 0, l'OS invité dans le ring 1, et les applications en ring 3 (modèle 0/1/3). Sinon, certaines implémentations fonctionnent à la fois le OS invité et l'application dans le ring 3 (modèle0/3/3). Ces deux Modèles sollicitent l'hyperviseur pour effectuer occasionnellement des traductions binaires

Pour résoudre ce problème, *AMD* et Intel ont mis en application une nouvelle technique de virtualisation, ce qui a présenté un nouveau niveau de privilège appelé le ring -1.

Le *hypervisor* fonctionne en ring -1, l'OS de control (par exemple, *Linux*) fonctionne dans le ring 0, et Les applications de l'OS contrôlé (par exemple, fonction de service *VMware -cmd*) fonctionnent en ring 3 (voir le schéma). Les machines virtuelles fonctionnent dans les ensembles *virtuels* de ring de sorte que chaque OS invité dans une VM croit qu'il fonctionne seul en ring 0.

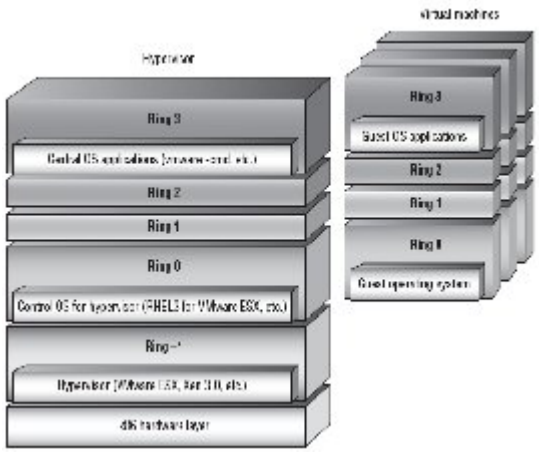


Figure 19 : nouveau ring pour l'implémentation de l'hyperviseur

Ces processeurs donnent à l'OS invité l'autorité d'accès direct aux ressources de plateforme sans partager le control du matériel. Précédemment, la VMM a comme rôle d'émuler le matériel à l'OS invité tandis qu'elle a gardé le control sur la plateforme physique. Ces nouveaux processeurs donnent le VMM et l'OS invité l'autorité pour que chacune doit fonctionner sans besoin d'émulation du matériel ou de modification d'OS. [2.1]

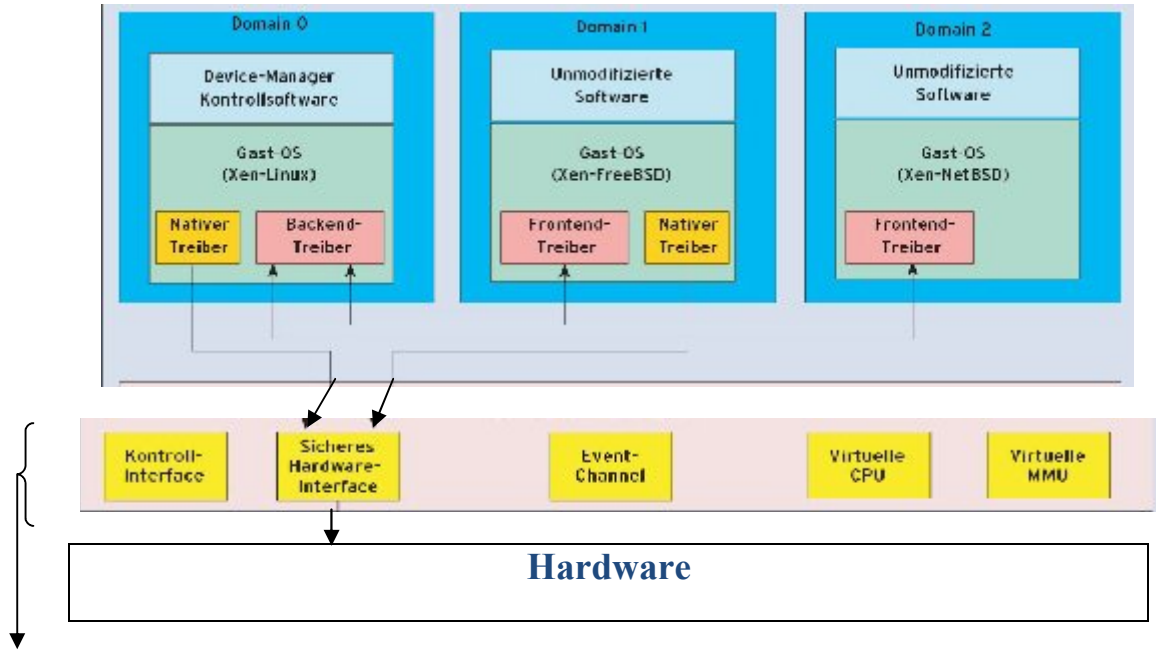


Figure 20 : fonctionnement de la VMM

Politiques de virtualisation des ressources

- **Politique de Partage :** Avec la politique de partage, le matériel réel et les ressources du système seront partagé (temps multiplexé) entre les multiples processeurs virtuels. Le VMM aura besoin de mettre en application les mécanismes de scheduling/switching /sharing pour soutenir cette politique.

- **Politique Consacrée:** Avec cette politique, les ressources matérielles réelles et de système sont consacrées à un processeur virtuel particulier. Il n'y aura aucun partage de ce matériel particulier et de ressource de système entre les processeurs virtuels. Le processeur virtuel aura le control directe de la plateforme matérielle.

Le VMM décide les politiques de virtualisation pour les processeurs virtuels au moment de leurs créations, les politiques sont applicables jusqu'à ce que le processeur virtuel soit terminé. [2.2]

2.5.2 La solution rapportée par Intel

Intel® VT signifie Intel Virtualization Technology (nom de code Vanderpool) et désigne un ensemble de fonctions de virtualisation implémentées dans le processeur. Ces dernières servent à obtenir des logiciels de virtualisation plus légers et plus robustes en déplaçant des fonctions importantes vers le matériel. Ces techniques rendent la virtualisation de Serveur plus rapide et plus flexible ; par exemple, certaines solutions logicielles prévoient une adaptation du système d'exploitation virtualisé.

Avec la mise en oeuvre de VT, cette limitation disparaît et il est ainsi possible, par exemple, de faire tourner un système Windows® non modifié sur un serveur Linux en tant qu'invité.

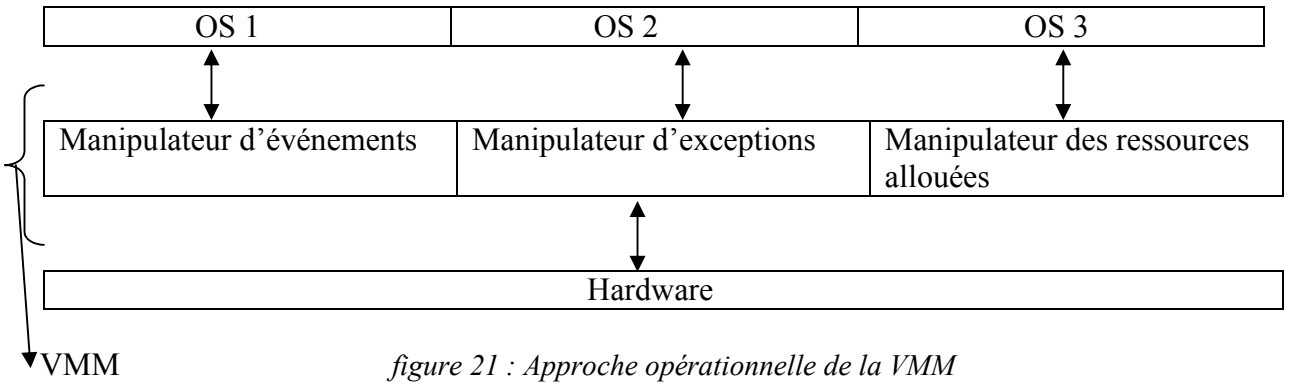


figure 21 : Approche opérationnelle de la VMM

La virtualisation matérielle permet d'optimiser au mieux les solutions de virtualisation logicielles. Elle améliore en outre la robustesse : comme en a déjà dit Intel®VT permet aux MMV d'exécuter des systèmes d'exploitation invités sans modification tout en augmentant le degré d'isolation des machines virtuelles entre elles. Concrètement, les processeurs Intel®VT offrent une prise en charge de la virtualisation pour les composants matériels suivants (Processeur, RAM, E/S, IOMMU = I/O Memory Management Unit)

Il existe actuellement trois composants VT :

- VT-x: architecture IA-32, p.ex. Processeurs Xeon®
- VT-i: architectures Itanium®
- VT-d : virtualisation E/S

La composante VT-x

Ces processeurs connaissent les trois rings décrits par la zone Nonroot dans laquelle sont exécutés les invités, ainsi qu'une zone Root parallèle et de construction identique pour les machines virtuelles. Cette architecture crée des nouvelles structures de contrôle afin de garantir que l'invité et le superviseur ne se croisent pas et que chaque invité bénéficie d'un espace d'adresse distinct. La zone Root est prévue pour l'hyperviseur ; son comportement correspond à celui d'un processeur normal sans composants VT. La zone Non-Root met à disposition un environnement IA-32 supplémentaire contrôlé par l'hyperviseur. [2.6]

La technologie de virtualisation d'Intel® est conçue pour permettre une performance élevé du VMM sans besoin de changements de la Paravirtualisation ou des techniques binaires de traduction [2.7]

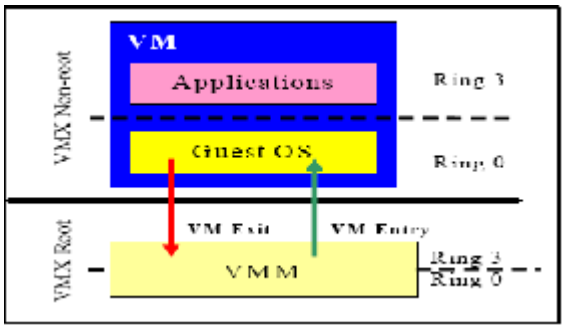


Figure 22: Intéraction VMM / VM

La composante VT-i

Ajoute aux processeurs Itanium® les composants cités ainsi que la couche d'abstraction du processeur, Processor Abstraction Layer (PAL). Cette dernière ouvre à l'hyperviseur une interface grâce à laquelle il peut par exemple définir des processeurs logiques à l'aide d'un descripteur virtuel (VPD). [2.4]-[2.5]-[2.6]

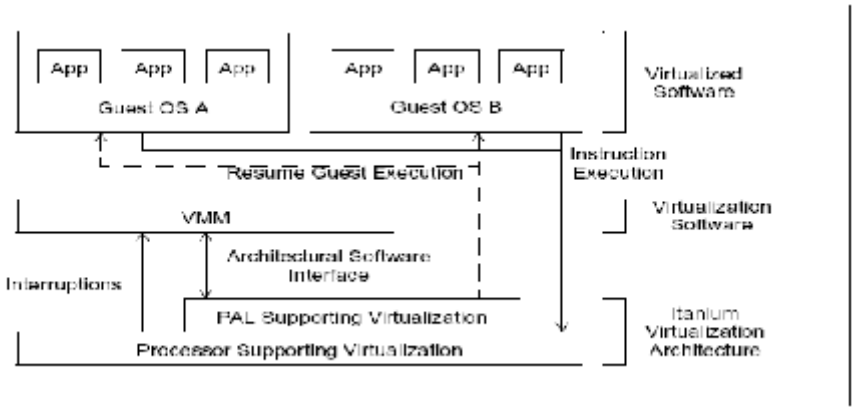


Figure 23: Interactions dans l'environnement de Virtualisation

La solution VT-d

VT-d (d pour directed I/O) est la future technologie qui permettra de compléter les solutions VT existantes avec une virtualisation E/S mise en oeuvre dans le matériel.
Un VMM doit supporter la virtualisation des demandes d'I/O de système invité. La virtualisation d'I/O peut être soutenu par un VMM par l'un des modèles suivants :

- **Emulation :** VMM émule la fonctionnalité de l'unité d'E/S Dans le système au-dessus du quel les dispositifs physiques sont disponibles sur la plateforme physique.

- **Nouvelles Interfaces Logiciel:** Ce modèle est semblable à l'émulation d'I/O, mais au lieu d'émuler les dispositifs, la VMM expose une interface synthétique de dispositif au système d'invité.
- **Tâche** : Un VMM peut directement assigner les unités d'E/S Physiques à la VMs. Dans ce modèle le driver des dispositifs d'E/S, fonctionne dans la VM à laquelle il est assigné.

- **Partage d'unité E/S** : Dans ce modèle, qui est un allongement au modèle de tâche, un dispositif d'I/O soutient les interfaces fonctionnelles multiples, dont chacune peut être indépendamment assignée à une VM.

Une condition générale pour tous les modèles ci-dessus de virtualisation d'I/O est la capacité d'isoler et limiter l'accès aux ressources par les dispositifs gérant la partition. Pour diriger I/O. Intel VT® fournit au logiciel VMM les possibilités suivantes :

- **Tâche de dispositif d'E/S** : pour assigner avec souplesse des unités d'E/S À VMs et prolonger la protection et les propriétés d'isolement de VMs pour des opérations d'I/O.

- **DMA remapping**: pour supporter les translations d'adresses indépendantes pour des accès mémoire directs (DMA) des dispositifs.

- **Interrupt remapping:** pour l'isolement et le routage des interruptions des dispositifs et des contrôleurs externes d'interruption pour une VM approprie.

Fiabilité: pour l'enregistrement et le reportage du logiciel système DMA et aux erreurs d'interruption qui peut autrement modifier la mémoire ou isoler la VM cible. [2.7] – [2.8]

2.5.3 Avantages de VT

Les nouvelles approches décrites offrent de nombreux avantages en améliorant considérablement les techniques de virtualisation de Serveur :

- Sécurité de fonctionnement grâce à un hyperviseur plus petit et plus simple.
- Économies grâce à l'utilisation des OSs invités « normaux » et non modifiés.
- Sécurité accrue grâce à une bonne isolation des MVs dès le niveau du processeur.
- Évolution simplifiée des futurs logiciels hyperviseurs.

2.5.4 Limitation de la solution Intel® VT

Il y a également quelques sujets de préoccupation dans l'utilisation de la technologie VT. VMM consomment l'objet valable qui traitant des ressources pour contrôler des opérations. C'est des frais généraux permanents qui doivent être acceptés par des utilisateurs et des concepteurs de système. En outre, la performance de système peut prendre un coup de la commutation de contexte qui se produit quand le VMM commute entre le mode VMX root et le mode VMX non-root.

Pour finir, les performances d'I/O du jour actuel VMM et les plateformes de VT sont limitées par l'architecture, entraînant une plus grande latence et une exécution inférieure de la sortie I/O. La technologie courante de VT (seulement VT-x est soutenu) pourra seulement accéder aux unités d'E/S Virtuelles, qui sont tracées aux unités d'E/S Physiques, au lieu de tracer directement aux unités d'E/S Physiques.

Les perfectionnements doivent déjà en cours fixer ces reculs. La commutation de contexte VMM et les frais généraux de ressource informatique s'améliore pendant que les fournisseurs VMM découvrent comment la virtualisation aidée par matériel des processeurs d'Intel peut augmenter l'exécution. Le goulot d'étranglement d'I/O est adressé par les chipsets qui soutiennent la virtualisation d'I/O (VT-d). La technologie de virtualisation pour I/O dirigé fournit au VMM les possibilités suivantes :

- Assigner les unités d'E/S À travers VMs : Assigner avec souplesse I/O à VMs et prolonge des propriétés de protection et d'isolement de VMs pour des accès d'I/O.
- Remap DMA : L'accès mémoire direct des dispositifs peut être directement traduit.
- Erreurs de disque et de rapport DMA

Ces dispositifs permettent au VMs de fournir une meilleure exécution d'I/O par une nouvelle interface logiciel, qui a moins de frais généraux comparés à l'émulation ; et une unité d'E/S Physique assignée directe, qui fournit des performances améliorées pour des applications intensives d'I/O. En outre, VMM peut également soutenir I/O aidé par dispositif partageant, qui fournit les interfaces fonctionnelles multiples, dont chacune peut être indépendamment assignée à une VM, permettant des dispositifs plus virtuels que les dispositifs physiques dans une plateforme. [2.3]

2.6 AMD virtualisation

2.6.1 Introduction

L'architecture du virtualisation AMD ™ (AMD-v™) est conçue pour fournir une technologie logicielle qui facilite le développement et le déploiement de virtualisation.

Fondamentalement, VMM travaille par intercepting et emulating (émulation) dans des opérations sensibles d'une façon sûre, dans l'invité (tel que : changer les tables de page, qui pourraient donner à un invité l'accès à une mémoire (inaccessible auparavant). AMD SVM fournit des aides matériels pour améliorer les performances et pour faciliter l'implémentation de la virtualisation.

Le support de processeur SVM fournit un ensemble des extensions matériel conçus pour permettre une économie et efficacité d'implémentation des systèmes à machines virtuelles. D'une manière générale, le support matériel rentre dans deux catégories complémentaires : support de virtualisation et sécurité.

2.6.2 Support De Virtualisation

L'architecture de la machine virtuelle AMD est conçue pour fournir :

- Les mécanismes pour lesquels le monde rapide commute entre VMM et invité
- Les capacités d'arrêter les instructions ou les événements choisis dans l'invité
- Protection de l'accès externe (DMA) pour la mémoire.
- Aides pour la manipulation d'interruption et le support virtuel d'interruption
- Un guest/host étiqueté TLB pour réduire des frais généraux de virtualisation.

Protection d'Accès Externe

L'invité peut accorder des accès direct aux unités d'E/S choisies. Le support matériel est conçu pour empêcher les dispositifs d'un invité d'accéder à la mémoire par un autre invité (ou par le VMM).

En fixant le mécanisme virtuel de translation d'adresses, le VMM peut limiter les accès du CPU invité à la mémoire. Si l'invité a l'accès direct aux dispositifs DMA, un mécanisme additionnel de protection est exigé. SVM fournit les domaines de protection multiples qui peuvent limiter l'accès du dispositif à la mémoire physique sur une base de per-page

Note : pour AMD on a préféré de ne pas citer les différents mécanismes conçus pour renforcer la virtualisation matérielle mais seulement on vous donne les liens officiels de AMD qui traitent ces instruction d'une façon détaillée. [2.9] - [2.10]

2.7 Interaction Virtualisation VS plateforme matérielle

Virtualisé votre infrastructure ou même un petit nombre de machines peut avoir de nombreux avantages, mais elle peut aussi affecter les performances du serveur, Poste de travail, du matériel ou des machines mobiles, même avec des progrès comme les processeurs multiCore. Il est important de comprendre certaines prescriptions qui se produisent Au niveau matériel avec la virtualisation. Cette section décrit les Composante par composante.

RAM, CPU, espace disque, carte réseau jouent tous un rôle pour Déterminer si une machine est prête à lancer une application dans un environnement virtuelle.

Préparer correctement votre hôte avant d'exécuter les machines virtuelles, elles vous aideront à assurer une meilleure stabilité, évolutivité et Performance à long terme pour vos machines virtuelles. Lors de la sélection d'un hôte, Vous devez vous assurer qu'il satisfait au Besoins de l'application minimale de la machine virtuelle en matériel et en outre que des ressources suffisantes, en particulier les Mémoires sont disponibles pour le nombre de machines virtuelles que vous voulez exécuter Simultanément sur l'hôte.

Par la suite on va présenter les diverses insuffisances produites au niveau du matériel, et les solutions proposées pour ce genre de problème.[2.11]

2.7.1 CPU

L'unité centrale de traitement est l'un des goulots d'étranglement (bottlenecks) les plus significatifs dans le système lors d'exécution. Tous les systèmes d'exploitation invitée exécutée sur un Host concurrencent pour l'accès au CPU.
Une solution efficace à ce problème est d'employer un **multi-processor**, ou bien un processeur Multi-cœur où l'on peu consacrer un noyau ou plus à une machine virtuelle.

La technologie pour assigner un noyau donné à une machine virtuelle, n'est pas encore entièrement fournie par les fournisseurs courants de virtualisation mais elle est prévue pour être disponible dans un proche avenir. En l'absence des processeurs multi -cœur, la meilleure solution est de trouver l'unité de traitement la plus rapide disponible pour satisfaire vos besoins.

2.7.2 Mémoire

La mémoire peut également être un bottlenecks significatif, d'une part, en choisissant le meilleur fournisseur pour votre solution de virtualisation parce que les divers fournisseurs manipulent différemment l'utilisation de mémoire.

Indépendamment du fournisseur que vous choisissez, vous devez avoir une quantité significative de la mémoire qui est rudement équivalente à la quantité que vous aurez assigné à chaque machine pour quelle soit exécuté en tant que machine physique. Par exemple, pour faire fonctionner le Windows XP Professional sur une machine virtuelle, vous pourrez assigner 256 megabytes (MB) de mémoire.

2.7.3 Disque physique

Souvent dans la virtualisation, l'utilisation de la globalité espace disque pour chaque machine virtuel n'est pas aussi un grand souci que l'utilisation intelligente de chaque commande physique. Un point important additionnel à considérer est la rotation de vitesse de la commande en service. Puisque on peu utiliser de multiples machines virtuelles sur une commande simple, la vitesse de rotation de la commande peut avoir un effet dramatique sur la performance avec une grande vitesse d'entraînement.

La meilleure performance à travers la plupart des produits de virtualisation aujourd'hui, c'est celle qui met en application multiples disques et emploie la commande la plus rapide, en termes de sa vitesse de rotation, pour chaque commande.

Le sens unique pour amplifier la performance de la solution de virtualisation, c'est juste d'avoir une commande plus rapide qui doit assurer que l'ordinateur Host et le système d'exploitation associé ont une commande physique dure consacrée, et cela pour toutes les machines virtuelle ou potentiellement chaque machine virtuelle a un disque dur physique séparé assigné à elle.

2.7.4 Network

L'utilisation du réseau peut également présenter des issues de bottlenecks, semblables à ceux de la mémoire. Quoique la machine virtuelle n'ajoute aucune quantité significative de latence dans l'équation de réseau, l'ordinateur Host doit avoir la capacité pour entretenir les besoins réseau de toutes les machines virtuelles courantes sur lui même. Cependant comme avec la mémoire on a besoin toujours d'appeler la quantité appropriée de largeur de bande de réseau et de ressources de network que l'on aurait si les machines fonctionnaient sur un matériel physique séparé.

On pourrait améliorer votre carte réseau si on fait fonctionner de multiples machines virtuelles dans un environnement VT et toutes les machines ont une expérience lourde concurrente le trafic network. Mais dans la plupart des scénarios du virtualisation desktop on constatera que le réseau n'est pas le problème. Très probablement le répréhensible est l'unité centrale de traitement, disque, ou mémoire. [1.2]

2.8 Conclusion

On a essayé d'introduire les différents types d'implémentations, en les classant et expliquant les intérêts de chaque structure, et ces inconvénients. Puis on a expliqué un peu la structure x86 et les rings de privilège pour leur intérêt dans le domaine de virilisation.

En comparant les deux techniques d'implémentation on constate que l'implémentation logicielle est économique mais très difficile à mettre au point, vu qu'il fallait tout adapter au matériel utilisé. Elle reste cependant plus performante.

Dans ce qui suit on va maître au point les différentes solutions de virtualisation et choisir la mieux adaptée par l'application des tests de performances.

CHAPITRE 3

Mise en œuvre et tests de performances

3.1 Introduction

Dans ce chapitre on va faire une étude comparative concernant les deux solutions de virtualisation hébergées virtualBox et VMware et la solution native qui est le Xen.

Ce chapitre fournit une étude détaillée sur les trois solutions de Virtualisation, discutant leur histoire, terminologie, et identifiant également les meilleures ressources pour leur mise en œuvre. Tout en présentant les méthodes et les systèmes les plus appropriés pour leur installation.

Pour conclure, on va appliquer un premier test de performance pour les trois solutions déjà citées afin de voir celle qui s'approche le plus des performances des machines réelles.

3.2 Virtual Box

Virtual Box est un logiciel permettant de lancer des machines virtuelles, tout comme VMware, Virtual PC et XEN …etc. Virtual Box permet d'installer de nombreux systèmes d'exploitation invités. Deux versions sont proposées par l'éditeur : la version de base est gratuite pour un usage personnel ou éducatif (licence PUEL), mais payante pour les entreprises. La version Open Source est entièrement libre mais amputée de certaines fonctionnalités pour les entreprises, et est disponible actuellement dans le dépôt Ubuntu universel. Virtual Box est disponible pour Windows, Linux, et Mac.

Virtual Box possède une interface (en QT) qui vous permettra de gérer très simplement vos différentes machines virtuelles. (Il existe une interface en SDL : VBoxSDL)
Ce qui différencie Virtual Box des autres virtualiseurs commerciaux, c'est qu'il offre une très bonne base GPL (et laisse augurer un avenir alléchant dans ce domaine), ainsi qu'une rapidité d'exécution bien supérieure à VMware par exemple. Ce qui en fait le virtualiseur idéal (à ne pas confondre avec un paravirtualiseur comme xen), malgré quelques bugs de jeunesse.

Figure 24 : Machine virtuelle sous VirtualBox

3.2.1 Le support OS:

3.2.1.1 Le système d'exploitation Hot:

Aujourd'hui Virtual Box est valable pour les systemes d'exploitations **Windows** 32-bit :

- Windows 2000, service pack 3 et Windows XP, all service packs
- Windows Server 2003 et Windows Vista

En plus virtual Box 1.5 peut être hébergé sous Windows Vista 64-bit. Et on trouve aussi le système d'exploitation **Linux** 32-bit:

- Debian GNU/Linux 3.1 ("sarge") and 4.0 ("etch")
- Fedora Core 4 to 8
- Gentoo Linux
- Redhat Enterprise Linux 3, 4 and 5
- SUSE Linux 9 and 10, openSUSE 10.1, 10.2 and 10.3
- Ubuntu 5.10 ("Breezy Badger"), 6.06 ("Dapper Drake"), 6.10 ("Edgy Eft"), 7.04

("Feisty Fawn"), 7.10 ("Gutsy Gibbon")
• Mandriva 2007.1 and 2008.0
Virtual Box 1.4, tourne aussi sur:
• 64-bit Linux
• Apple Mac OS X
Il est possible de tourner Virtual Box sur plusieurs systèmes basée sur Linux kernel 2.4 ou 2.6

3.2.1.2 Le système d'exploitation invité:

Comme Virtual Box est désigné pour fournir un environnement de virtualisation pour système x86, dans la suite on va lister les déférents systèmes d'exploitation qui peuvent être utilisé comme des invitées :

Operating system	Support status
Windows NT 4.0	All versions/editions and service packs are fully supported (but see remark 1 below). Guest Additions are available with a limited feature set.
Windows 2000 / XP / Server 2003 / Vista	All versions/editions and service packs are fully supported. Guest Additions are available.
DOS / Windows 3.x / 95 / 98 / ME	Limited testing has been performed. Use beyond legacy installation mechanisms not recommended. No Guest Additions available.
Linux 2.4	Limited support.
Linux 2.6	All versions/editions and service packs are fully supported (but see remark 2 below). Guest Additions are available.
FreeBSD	Limited support. Guest Additions are not available yet.
OpenBSD	Versions 3.7 and 3.8 are supported. Guest Additions are not available yet.
OS/2 Warp 4.5	Requires VT-x hardware virtualization support to be enabled. We officially support MCP2 only; other OS/2 versions may or may not work. Guest Additions are available with a limited feature set.

Tableau 6 : OS invités supportés par Virtualbox

3.2.3 Eléments importants de Virtual Box :

3.2.3.1 Virtual Disk Image (VDI) files:

Par, défaut, Virtual Box utilise son propre format pour le disque dur invité – Virtuel Disque Image (VDI) classe.
Le fichier VDI réside sur le système hôte, est vu par le système invité comme un disque dur. Au moment de leur création on spécifie la dimension du disk. Il est donc impossible de transformer ces dimensions après. Pour créer un disque dur virtuel il faut spécifier 2 choses :

- **size image**
- **dynamically expanding image (immutable or normal images)**

3.2.3.2 VMDK image files :

La version 1.4 de Virtual Box supporte aussi le format VMDK, celle supportée par la plupart des solutions de virtualisation. Cela nous permet de récupérer une image existante et la faire fonctionner. L'outil qui gère ces applications est le Virtual Disk Manager :

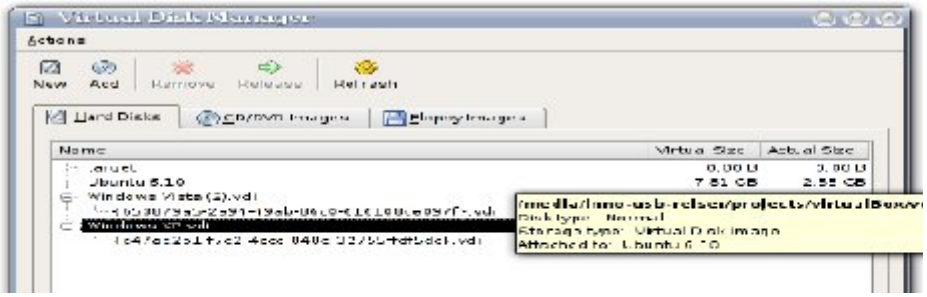


Figure 25 : Virtual disk Manager

Remarque :

Dans cette partie on a parlé des éléments importants dans l'utilisation de Virtual Box pour plus d'informations vous pouvez consulter la référence suivante

3.2.5 Installation de Virtual Box :

3.2.5.1 Sous Windows :

L'installation du programme est tout ce qu'il y a de plus classique. Après avoir téléchargé Virtual Box, un simple double clic sur le fichier exécutable lance l'installation :

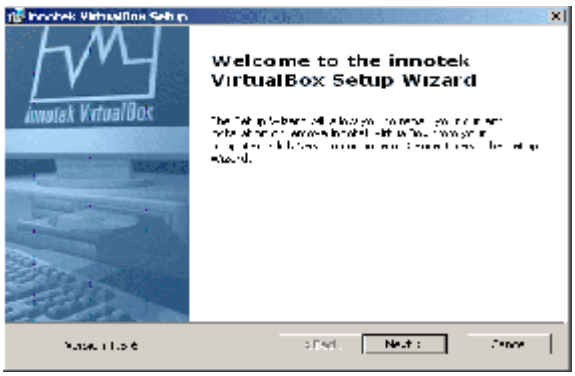


Figure 26 : Installation Virtualbox

Après avoir accepté le contrat de licence et spécifier l'emplacement de l'installation Cliquez sur le bouton **Next**. Vous obtenez le message suivant, pas de panic c'est tout a fait normale il suffit de continuer

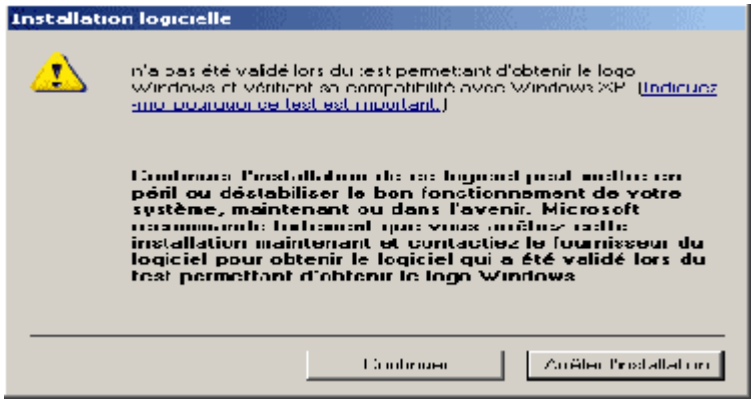


Figure 27 : Installation de pilotes

Virtual Box permet d'émuler complètement un PC. C'est comme si vous aviez un second PC dans une simple fenêtre. C'est utile pour tester d'autres systèmes d'exploitation sans repartitionner et sans risque (par exemple Linux), pour naviguer en toute sécurité ou pour tester un logiciel sans risque de rendre son système d'exploitation instable.

Vous pouvez créer autant de machines virtuelles que vous le souhaitez, et installer tous les systèmes d'exploitation que vous le voulez dedans.
Il est possible de définir - pour chaque machine virtuelle - combien de mémoire elle possède, de disque dur, si elle aura accès aux ports USB, au réseau, à la carte son, etc. Virtual Box contient un gestionnaire de disques qui vous permet de créer des disques virtuels sous forme de fichiers .vdi qui apparaîtront comme de vrais disques dans les machines virtuelles.
Cela vous permet donc de "créer" à volonté des disques, et cela sans jamais avoir à repartitionner votre disque dur.

Vous pouvez également utiliser directement des images ISO de CD et DVD, ce qui permet de tester des distributions Linux sans avoir à les graver. Cerise sur le gâteau, Virtual Box possède un serveur RDP intégré, ce qui permet de démarrer une machine virtuelle sur un ordinateur, et utiliser cette machine virtuelle à partir d'un autre ordinateur. [3.1]

3.2.5.1.1 Création des machines virtuelles

Cliquez sur le menu de démarrage et un clic sur innoTek Virtual Box, la fenêtre suivante 34. Ainsi vous pouvez modifier la configuration de la machine virtuelle a tout moment il suffit d'arrêter la machine et utiliser l'outil **Préférences** montrer dans la figure 35.

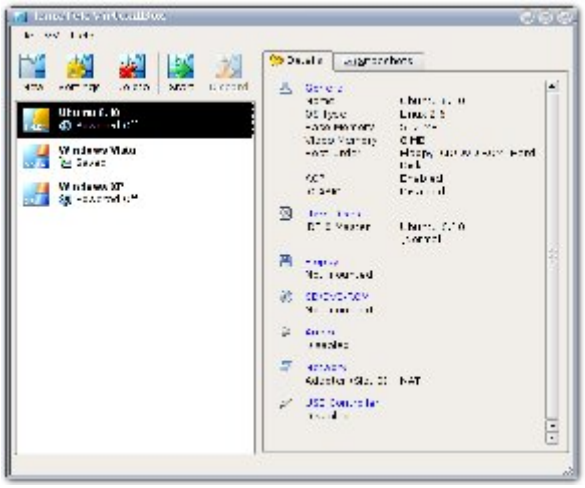


Figure 28: Interface Virtualbox

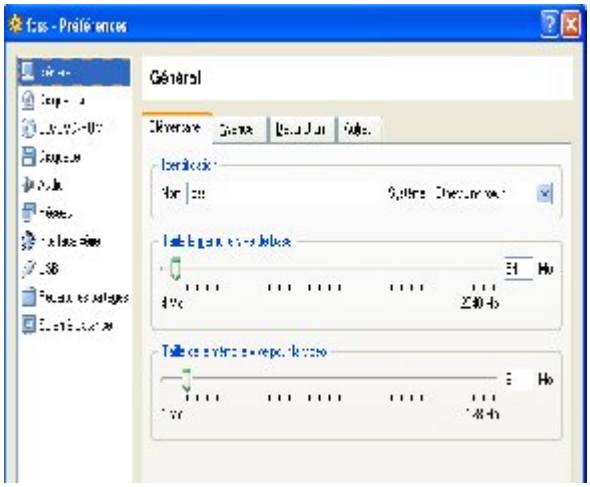


Figure 29 : Configuration des périphériques

Pour d'avantage d'information je vous invite à consulter la référence.

3.2.5.1.2 Configuration du réseau

Normalement, une carte réseau virtuelle est créée à l'installation de votre machine. Vous pouvez en avoir jusqu'à 4 par machine virtuelle (numérotées de 0 à 3). Vous avez la possibilité de créer un réseau entre l'ordinateur virtuel et l'ordinateur hôte, ou entre les machines virtuelles, ou bien d'intégrer votre ordinateur virtuel dans votre réseau (par exemple, comme s'il était relié directement à votre modem / routeur).
Dans tous les cas, ouvrez la fenêtre '**Settings**' dans la partie '**Network**'.[3.1]

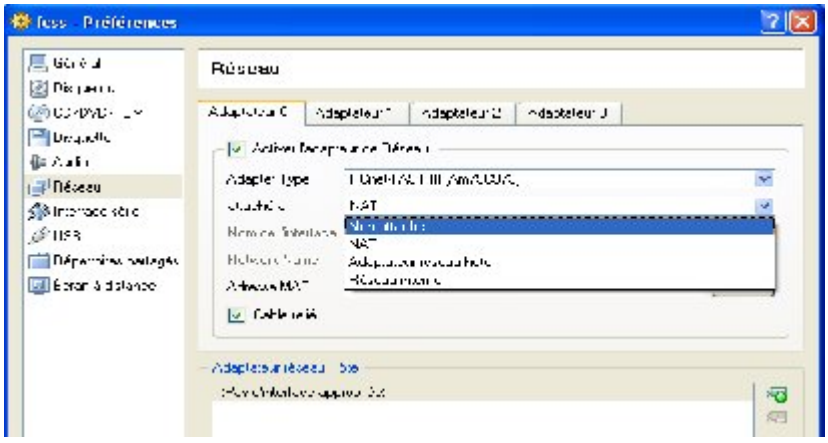


Figure 30 : Configuration réseau

Relier directement votre machine virtuelle à votre réseau

Cochez la case '**Enable Network Adapter**' si elle n'est pas déjà cochée. Dans la partie '**Attached to**', sélectionnez '**NAT**'. Vérifiez que la case '**Cable Connected**' est bien cochée. Maintenant, vous n'avez plus qu'à configurer le réseau d'une manière habituelle. Votre ordinateur virtuel sera connecté à votre réseau de la même manière que votre ordinateur physique.

Créer un réseau entre votre ordinateur virtuel et votre ordinateur réel

Cochez la case '**Enable Network Adapter**' si elle n'est pas déjà cochée. Dans la partie '**Attached to**', sélectionnez cette fois '**Host Interface**'. Là, entrez un nom dans le champ '**Interface name**' de la partie '**Host Interface Settings**', puis cliquez sur le bouton qui représente deux ordinateurs. Si vous utilisez Windows, il va vous demander de valider l'installation du périphérique.

Maintenant si vous voulez connecter les différentes machines virtuelles entre elles il suffit de créer un pont de connexion au niveau de la machine Host.

Création du pont de connexion

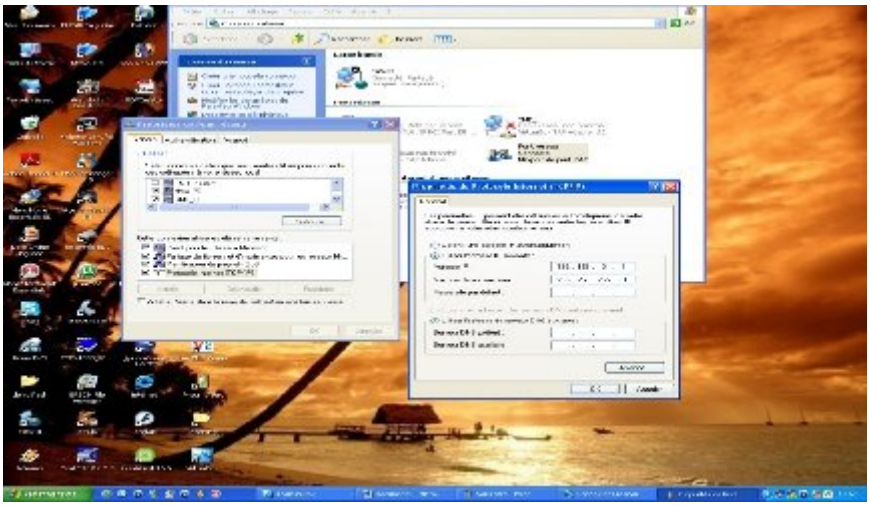


Figure 31 : création du pont

Dans linux on utilise ou bien l'outil terminal ou bien directement a partir de l'interface graphique : La configuration du réseau consiste à affecter a chaque machine l'adresse IP, l'adresse de la passerelle, et le masque plus le DNS, Maintenant pour vérifier l'interconnexion entre les différentes machines il suffit de faire des **PING** a partir de chaque machine vers les autres machines, toute en utilisant les adresses IP.

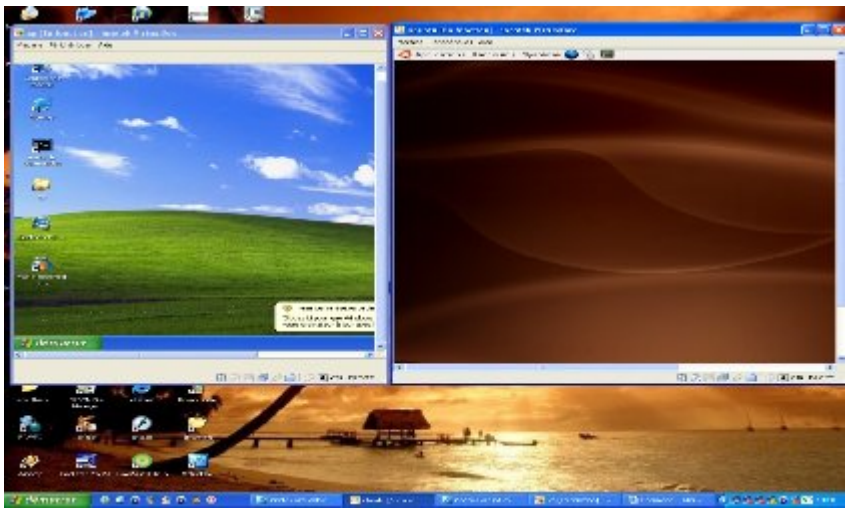


Figure 32 : Winxp et Ubuntu sous Winxp

3.2.5.2 Installation de virtualBox sur ubuntu 7.10

Dans cette application nous avons installé virtualBox non pas sur Windows xp mais sur linux et plus précisément sur la distribution ubuntu 7.10.

La procédure est la suivante : Ouvrir le terminal et taper les commandes suivantes :

```
$sudo apt-get update
$sudo apt-get install virtualbox-ose virtualBox-ose-source
$sudo apt-get install module-assistant
$sudo m-a prepare
$sudo m-a a-i virtualbox-ose
$sudo modprobe vboxdrv
$sudo gedit /etc/modules
$vboxdrv
$sudo adduser [mot de passe] vboxusers
```

pour acceder a virtual box cliquez sur : **Applications → System Tools → InnoTek VirtualBox**

Figure 33 : VirtualBox avec les deux OS invités Ubuntu et Winxp

Mise au point du réseau local

Installation des paquets **bridge-utils**, **uml-utilities**

```
#apt-get install bridge-utils, uml-utilities
```

Création du bridg de connexion

```
$sudo –s
$password :
$ vi /etc/network/interfaces
```

Mettre la configuration suivante :

```
>auto lo
>iface lo inet loopback

>auto eth0
> iface eth0 inet static
>       address 192.168.0.10
>       netmask 255.255.255.0
>       gateway 192.168.0.252
>
> auto tap0
> iface tap0 inet manual
>      tunctl_user nabil
>
> auto tap1
> iface tap1 inet manual
>      tunctl_user nabil
>
> auto bridge0
> iface bridge0 inet static
>    post-up chmod ugo+rw /dev/net/tun
>    address 192.168.0.10
>    netmask 255.255.255.0
>    gateway 192.168.0.252
>    bridge-ports eth0 tap0 tap1
>    bridge-ageing 7200
>    bridge-fd 0
Echap <: x>
```

Ajouter l'utilisateur au groupe **uml-net**

```
$ sudo adduser nabil uml-net
```

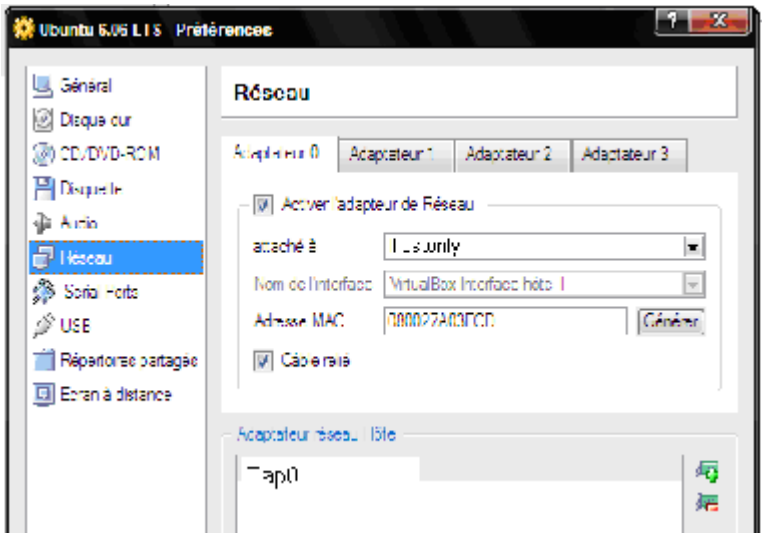


Figure 34 : Configuration du réseau Virtualbox

3.2.5.3 Avantages et inconvénients

Avantages

- Virtual Box est gratuit et open source, et des versions sont disponibles pour Windows et Linux.
- Virtual Box est plus léger que VMware et plus simple à utiliser, et offre la quasi-totalité des fonctions de VMware.
- L'utilisation est assez aisée car Virtual Box possède une interface graphique très simple
- Vous pouvez modifier les caractéristiques de la machine virtuelle après, pendant que le système virtuel n'est pas lancé. Vous pouvez observer la consommation de Virtual Box dans la console grâce au programme top
- Virtual Box propose des drivers adaptés pour l'OS Guest : pour Windows et pour Linux. D'ou :
  - Un redimensionnement automatisé de la zone d'affichage de l'OS Guest, quand on redimensionne la fenêtre d'émulation dans l'OS Host.
  - La possibilité de passer la souris du Host au Guest, et vice versa, plus aisément.
  - Le partage de fichiers entre l'OS Host et l'OS invité sous peine d'obtenir le message d'erreur.

Inconvénients

Cette solution a des inconvénients. Parmi les plus gênants:

- Lenteur: L'émulation par la machine virtuelle fait que Linux est plus lente que ce qu'il serait s'il était lancé seul.
- Pas d'accélération 3D: Linux n'ayant accès qu'à la carte graphique virtuelle de Virtual Box, il ne pourra pas profiter de l'accélération 3D de votre carte graphique réelle.

- Lenteur d'accès disque: Si le fichier contenant le disque virtuel est fragmenté, les accès disque seront d'autant plus lents. (Vous pouvez réduire cette fragmentation en défragmentant vos fichiers .vdi avec des logiciels tels que contig ou JKDefrag).
- Virtual Box consomme beaucoup de RAM car elle est réservée au système d'exploitation lancé. Cela se configure lors de la création du système virtuel.
- La version OSE **Open Source Edition** ne gère pas les périphériques USB.
- Dans la version Open source, il ya un manque des options utiles qui peuvent être disponible dans la version propriétaire telle que les outilles de clonage et d'importation des images machines existante déjà comme celles disponibles dans VMware.
- Il est impossible d'utilisée le même disque dur virtuel pour plusieurs machines contrairement avec VMware ou chez les versions commerciale.

- Virtual box marche Sous Ubuntu 64bits, mais ne permet pas encore l'émulation d'un système d'exploitation 64bits. [3.1]

3.3 VMware

3.3.1 Installation Vmware sous Winxp

VMware Inc. est une société filiale d'EMC Corporation, fondée en 1998, qui propose plusieurs produits propriétaires liés à la virtualisation d'architectures x86. C'est aussi par extension le nom d'une gamme de logiciels de virtualisation.

Bref Historique

- 1998: création de la société
- 1999: VMware Workstation 1.0
- 2000: IBM, Dell et Compaq deviennent partenaires
- 2001: VMware GSX Server 1.0 et ESX Server 1.0
- 2002: HP devient partenaire, Dell revendeur, 1 million d'utilisateurs
- 2003: VirtualCenter 1.0, technologie VMotion
- 2004: Rachat par EMC_Corporation, annonce du support 64 bits
- 2006: VMware ESX 3.0, VirtualCenter 2.0
-

Fonctionnements

VMware crée un environnement clos dans lequel sont disponibles un, deux ou quatre processeurs, des périphériques et un BIOS virtuel.
Selon les concepteurs, le microprocesseur n'est émulé que quand c'est nécessaire, c'est-à-dire quand la VM (machine virtuelle) tourne en mode noyau ou en mode réel, mais pas pour le mode utilisateur (user mode) ou le Mode virtuel 8086. Selon les concepteurs, cela permet à VMware d'être plus rapide que des solutions multi plateformes qui émulent tout.

Lorsqu'une VM exécute dans un mode qui nécessite une émulation, VMware traduit dynamiquement le code privilégié en un code équivalent en mode utilisateur, le place dans un endroit libre de la mémoire, le rend invisible et inaccessible au code d'origine et l'exécute à la place. Lorsqu'une machine virtuelle fait appel à un périphérique, VMware intercepte la demande et la traduit pour qu'elle soit gérée par le système hôte. Bien que les machines virtuelles tournent en mode utilisateur, VMware nécessite d'installer plusieurs pilotes de périphériques privilégiés dans le noyau du système hôte, qui notamment interchangent les tables GDT et IDT chaque fois qu'on passe la main à une VM. [3.4]

VMware assure l'émulation de la carte vidéo, la carte réseau, le lecteur de CD-ROM, le bus USB, les ports séries et parallèle et le disque dur de type SCSI ou IDE. Ce dernier étant un fichier extensible d'une taille voisine de la place occupée sur la machine virtuelle ou fixe pour davantage de performance. Ce fichier contenant le contenu du disque peut être copié sur un autre hôte et exécuté par un ordinateur. Pour l'ordinateur virtuel, tous les périphériques sont identiques, même si le système hôte est totalement différent, car c'est VMware qui caractérise les périphériques.

Produits

Actuellement, les produits suivants sont disponibles :

- VMware *Workstation*, VMware *Player* et VMware *ACE* : logiciels pour stations de travail ;
- VMware *Fusion* : logiciel pour stations de travail Macintosh avec processeurs Intel ;
- VMware *GSX Server*, VMware *Server* et VMware *ESX Server* : logiciels pour serveurs ;
- VMware *Virtual Center* et VMware *Converter* : logiciels de gestion et outils.

La combinaison de ces différents produits crée ce que VMware nomme commercialement une infrastructure virtuelle. [3.6]

VMware ESX est un système d'exploitation ou hyperviseur basé sur la distribution Linux Redhat 7.3. Ce hyperviseur est composé de plusieurs modules :

VMKERNEL

Il permet de gérer et de hiérarchiser l'ensemble des ressources matérielles (mémoire, CPU, disques, réseaux) en fonction de chaque serveur. De plus c'est ce noyau qui est en charge de toute la gestion des ressources physiques pour ESX.

SERVICE CONSOLE

Elle permet la gestion de l'hyperviseur en mode commande. Accessible depuis le port 22 (SSH) elle sert à lancer certaines commandes inaccessibles depuis l'interface graphique ou encore de parcourir les dossiers dans lesquels sont stockées les machines virtuelles. Enfin elle peut permettre de récolter des informations de debug sur les machines virtuelles ou sur le serveur ESX.

Nombres d'options sont disponibles à travers le service console, il est cependant déconseillé de manipuler ESX depuis cette interface.

La gestion des serveurs se fait à l'aide d'un navigateur via une interface web, à l'aide d'une console cliente (Virtual Infrastructure Client) ou d'un outil de gestion centralisé Vmware nommé Virtual Center. Une machine virtuelle est en fait un assemblage de plusieurs fichiers. Ces fichiers sont créés sur un système de fichiers appelé/formaté *vmfs*.

Ce système de fichiers possède plusieurs caractéristiques, la plus signifiante est qu'il est capable de gérer plusieurs connexions concurrentes. Il faut bien sur que cet espace soit commun à tous les serveurs ESX (SAN par exemple). [3.2]

Virtual Center

Cet outil de gestion permet de gérer l'ensemble des machines virtuelles et des hôtes physiques. Il est également possible à travers de cette interface de gérer :

- Les alarmes de supervision (CPU/RAM)
- Les template (enveloppe de systèmes d'exploitation pré-configurés)
- L'utilisation des options (HA; Vmotion; DRS)

VMotion

Cet outil permet de migrer "à chaud" SANS INTERRUPTION DE SERVICE une machine virtuelle d'un serveur ESX vers un autre. Cette opération est possible lorsque les serveurs hôtes utilisent des microprocesseurs identiques et que l'espace de stockage des fichiers des machines virtuelles se trouve partagés sur un SAN. [3.5]

VMware HA (High Availability)

Cette option de Virtual Center consiste en un mécanisme de bascule des machines virtuelles d'un serveur ESX en panne vers un autre serveur ESX.

VMware Consolidated Backup

C'est l'outil de sauvegarde de Virtual Center. Il permet de faire des sauvegardes des machines virtuelles (totale, incrémentale...).

VMware Converter

C'est un outil de migration qui permet de transformer le contenu d'un serveur physique existant vers une machine virtuelle VMware (**P2V**: "**P**hysical to **V**irtual"). Après avoir fait une image du contenu des disques du serveur physique, *Converter* analyse celle-ci et y fait des modifications afin de pouvoir booter ces disques dans une machine virtuelle. Les modifications portent essentiellement sur le remplacement des pilotes dans le système d'exploitation, notamment ceux liés aux contrôleurs de disques. Cela permet d'éviter de réinstaller complètement le système d'exploitation lors d'une migration vers un environnement virtuel.
Note : pour avoir plus d'information sur ces options voir [3.3] - [3.4]

3.3.2 Application Mise en œuvre

3.3.2.1 sous windows

Dans notre application nous avons utilisé le VMware Workstation et le VMware server lesquels ont été installés sur WindowXp (Virtualisation Hébergé)

Figure 35: OS invité1 Fedora Core 8

Figure36: OS invité 2 windows XP

Figure 37: OS invité 3 WindowsXp

3.3.2.2 But de l'application

Le but de cette application est la conception d'un réseau local virtuel liant tous les OS invités entre eux ainsi que la mise au point d'un cyber café virtuel. c a d que chaque OS invité pourra se connecter a Internet via une connexion partagée assurée par le Hote principale.

Procédure

1-Installation de VMware ainsi que les OS invité.

2-Déclaration des cartes réseaux virtuelles pour chaque machine.

Figure 38: installation Vmware et OS invités

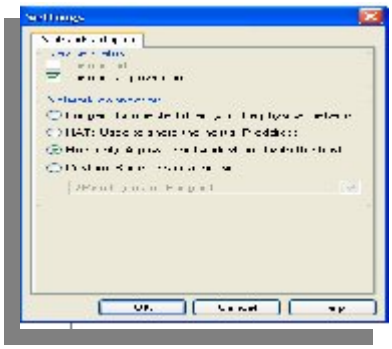


Figure 39: Configuration réseau

3- Configuration des adresses IP

4- Partage de la connexion

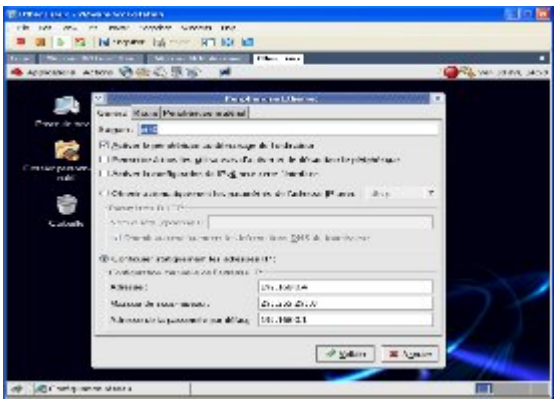


Figure 40 : Configuration des adresses IP

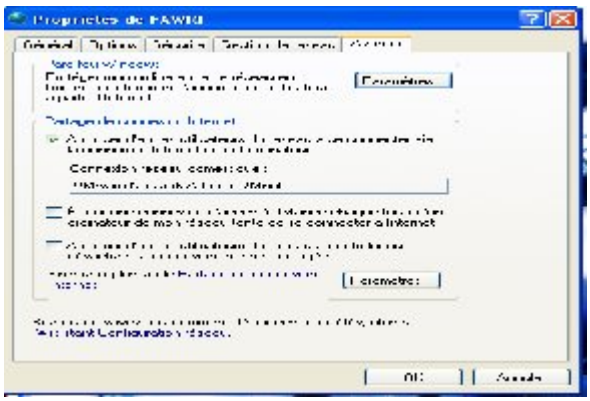


Figure 41 : partage de la connexion

3.3.2.3 Problèmes rencontrés

La mise au point de l'hyperviseur VMware n'a pas posé de sérieux problèmes si ce n'est la configuration du réseau locale, par contre on a remarqué que le VMware était beaucoup plus gourmant que Virtualbox dans l'attribution de la mémoire aux machines virtuelles.

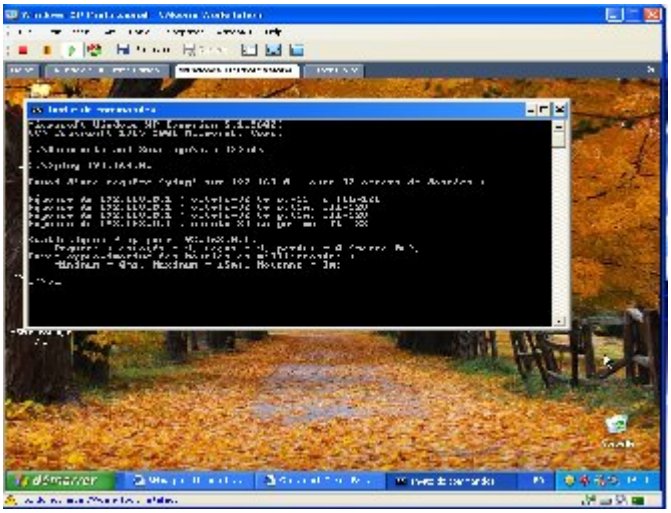


Figure 42 : Ping de l'OS invité XP vers l'OS hôte.

3.3.3 Installation VMware Server sur ubuntu 7.10 :

Pour installer VMware sous ubuntu if suivre la procédure suivante :

```
$ mkdir ~/VMware  (création du dossier pour le stockage des VM)
```

- Installer les paquets nécessaires (7.10 32bits)

```
$ sudo apt-get install ssh openssh-server libx11-6 libx11-dev libxtst6 xinetd wget build-essential

$ sudo apt-get -y install linux-headers-`uname -r` && cd /usr/src && sudo rm -f linux && sudo ln -s linux-headers-`uname -r` linux

$ sudo apt-get install gcc binutils-doc cpp-doc make manpages-dev autoconf automake1.9 libtool flex bison gdb gcc-doc libc6-dev-amd64 lib64gcc1
```

- Télécharger VMware Server 1.04 : VMware-server-1.0.4-56528.tar.gz, puis décompresser l'archive :

```
$ cd /tmp
$ wget http://download3.vmware.com/software/vmserver/VMware-server-1.0.4-56528.tar.gz
$ tar -xvzf VMware-server-1.0.4-56528.tar.gz
```

- Maintenant il suffit de Lancer l'installation :

```
$ cd VMware-server-distrib
$ sudo ./vmware-install.pl
```

Répondre par défaut à toutes les questions lors de l'installation sauf à l'emplacement de stockage des machines virtuelles et lors du renseignent le numéro de série obtenu lors de l'inscription)
- Pour le lancement de VMware server taper

```
$VMware
```

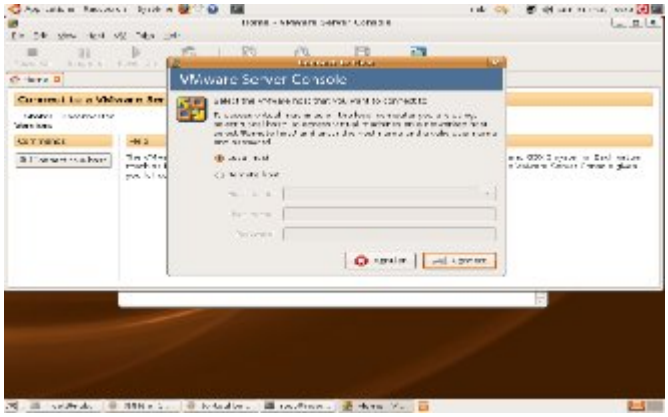


Figure 43 : Lancement Vmware server

3.3.3.1 Création des Machines virtuelles

Pour la création des machines virtuelles la procédure est la même que pour Winxp.

Mise au point du réseau

Dans la menue configuration réseau on choisit l'option bridged comme montrée dans la figure

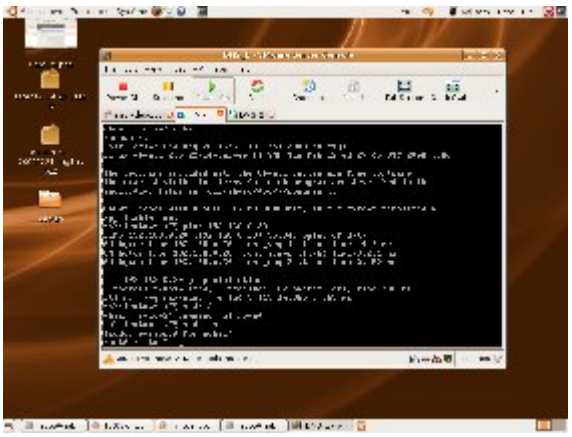


Figure 44 : Lancement de MV (ubuntu server 8.04)

Figure 45 : configuration réseau

3.4 Xen

Historique du projet

Le projet (open source) Xen a été fondé par Ian PRATT en 2003, en se basant sur un projet de recherche de l'université de Cambridge. La première version de Xen, sortie en 2003, était limitée dans ses performances et ses fonctionnalités. Elle a toutefois réussi à attirer des contributeurs et des investisseurs, permettant d'amorcer le développement.

La version 2 du projet a été diffusée fin 2004, et a apporté beaucoup d'améliorations, tant en terme de fonctionnalités (support des noyaux récents, couche réseau plus flexible, etc.) qu'en terme de performances. La version 2 a été maintenue jusqu'au milieu de l'année 2005, date de sortie de la version 2.0.6.

En parallèle, la communauté avait déjà commencé à développer la nouvelle version majeure du projet, la version 3.0. Elle a été diffusée début 2006, et a apporté davantage de fonctionnalités, notamment le support de plusieurs processeurs dans les domaines utilisateurs, améliorant sensiblement les performances. La version 3.0.2 a ajouté le support des instructions de virtualisation pour les processeurs, offrant la possibilité d'exécuter des systèmes invités au dessus de l'hyperviseur, sans avoir à les modifier. (Virtualisation complète).

En mai 2007 est sortie la version 3.1 de Xen, nouvelle version majeure améliorant encore le support des instructions de virtualisation et les performances en général.

3.4.1 Analyse détaillée

Xen est un projet très complexe, constitué de plusieurs composants visant à fournir une solution de virtualisation utilisable très simplement par l'administrateur. Cette simplicité apparente cache toutefois une complexité technique non négligeable, qu'il est souhaitable de maîtriser — ou au moins de connaître — si l'on veut pouvoir exploiter Xen au maximum de ses possibilités.

L'analyse détaillée de Xen portera sur la version 3.0.1 du projet. Les évolutions apportées par cette version n'ayant pas fortement modifié l'architecture globale du logiciel, la plupart des documentations techniques portant sur la version 2 sont applicables à la version 3. [4.1]
Le projet Xen peut être séparé en plusieurs modules :

– l'hyperviseur ;
– les *patches* à appliquer au noyau Linux ;
– les programmes de contrôle en espace utilisateur.

La figure 53 représente l'architecture générale de Xen. Par rapport à l'architecture théorique d'un système à hyperviseur, on peut noter la présence d'un système invité particulier, dénommé domaine 0. Ce système manipule l'hyperviseur Xen à travers une interface standardisée.

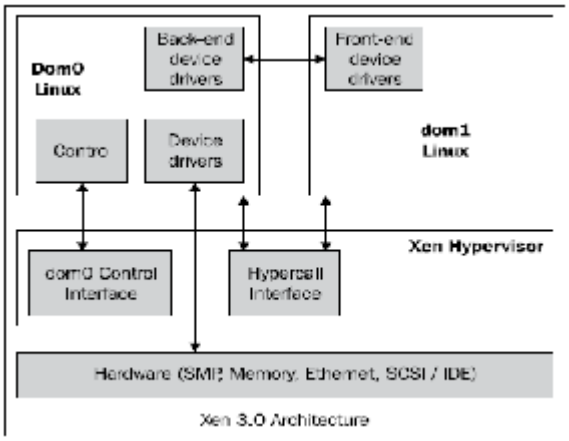


Figure 46 : architecture générale de Xen

Toutes les actions d'administration passent par l'interface de programmation *(API, Application Programming Interface)* de l'hyperviseur Xen. Ainsi, quand l'administrateur souhaite démarrer un domaine invité, il exécute un utilitaire dans l'espace utilisateur, qui fait appel à l'interface de contrôle du domaine 0. Cette dernière communique alors directement avec l'hyperviseur pour instancier le domaine invité.

Seul le domaine 0 a un accès complet au matériel (mais toujours à travers l'hyperviseur). C'est donc dans le domaine 0 que s'effectue toute la configuration des systèmes invités :

– réservation de l'espace disque ;
– création du fichier de configuration du domaine utilisateur ;
– création ou copie des fichiers du système d'exploitation...

Inversement, les domaines utilisateurs n'ont accès qu'à ce que l'administrateur a configuré. L'hyperviseur est démarré directement par le chargeur de démarrage de l'ordinateur, en lieu et place d'un système d'exploitation traditionnel. L'hyperviseur instancie ensuite le domaine 0 et lui passe la main pour le reste de l'initialisation des périphériques, tâche qui n'est pas réalisée par l'hyperviseur Xen.

En pratique, au démarrage de la machine, on voit s'afficher quelques lignes traitant de l'hyperviseur et de sa configuration avant que le noyau Linux ne prenne le relais et entame le processus de démarrage « traditionnel ».[4.2]

Comme pour le noyau Linux, il y a une partie de la configuration de l'hyperviseur qui doit être passée par le chargeur de démarrage, soit parce qu'elle n'est pas modifiable en cours d'exécution (comme par exemple la quantité de RAM à affecter au domaine 0 ou les paramètres de la console série), soit parce que la lecture dans un fichier de configuration est impossible car le système de fichiers n'est pas encore activé au moment où l'information est nécessaire. Dans la configuration du chargeur de démarrage, l'hyperviseur apparaît comme un système d'exploitation de type GNU/Linux, avec simplement des options différentes de celles passées à un noyau Linux.

Techniquement, l'hyperviseur est le point de passage obligatoire pour tout accès au matériel. Il régule et répartit les accès aux ressources entre les systèmes invités (domaine 0 et domaines utilisateurs). À l'image des ordonnanceurs du noyau Linux pour la répartition des accès au processeur et aux périphériques d'Entrées/Sorties entre les différents processus du système d'exploitation, les ordonnanceurs de l'hyperviseur Xen répartissent les Entrées/Sorties et le temps processeur entre les systèmes invités.

À l'inverse, l'accès au processeur n'est pas exclusif, tous les systèmes invités en cours d'exécution ont chacun une part du temps processeur total disponible sur la machine.
Selon le type d'ordonnanceur choisi, l'administrateur peut assigner des priorités aux systèmes, afin d'affecter un quantum de temps plus important à un système par rapport aux autres.
En plus de gérer l'accès aux ressources, l'hyperviseur doit aussi gérer la correspondance entre la représentation de la mémoire des systèmes invités et la disposition effective de la mémoire pour le processeur. En effet, on dit que c'est le processeur qui s'occupe de la gestion physique de la mémoire sur les architectures PC.

C'est l'hyperviseur qui fait la correspondance entre l'adresse virtuelle du système invité et l'adresse réelle manipulée par le processeur.

Le noyau Linux et l'hyperviseur communiquent par le biais d'appels systèmes spécifiques, appelés *hypercalls* (pour *hypervisor calls*, appels à l'hyperviseur). Ces appels permettent de passer des messages à l'hyperviseur, sur les tâches à accomplir (par exemple l'allocation ou la libération de mémoire).

3.4.2 Modifications apportées au noyau Linux

La plupart des modifications apportées au noyau Linux sont nécessaire pour la paravirtualisation réalisé par l'hyperviseur, notamment pour la gestion de la mémoire, qui concentre la majorité des modifications.

Les *patches* ont pour effet de rajouter une architecture matérielle dans les options de compilation du noyau : xen_x86. Cette architecture matérielle, « fictive » car elle ne correspond pas réellement au matériel, permet de regrouper toutes les modifications, au lieu de modifier directement les fichiers concernés pour l'architecture x86.

La communication avec l'hyperviseur s'effectue au moyen d'*hypercalls*. Il y a en tout une trentaine d'*hypercalls* définis, couvrant toutes les opérations courantes de Xen : gestion de la mémoire, gestion des Entrées/Sorties, etc.

Si la communication depuis le système invité vers l'hyperviseur est réalisée par un *hypercall*, le passage d'informations dans le sens inverse est réalisé par un bus d'évènements, au fonctionnement très similaire à celui des interruptions matérielles. Dès qu'un évènement susceptible nécessite une action de la part du système survient (arrivée d'un paquet sur l'interface réseau, fin d'un transfert de données sur le disque, ...), il est signalé au système, qui

prend alors le relais et traite l'évènement. Les bus d'évènements de Xen ont le même rôle, mais pour la transmission d'informations relatives à l'hyperviseur (par exemple domaine utilisateur créé ou arrêté avec succès).[4.1]

3.4.3 Applications en espace utilisateur

Les applications utilisateurs sont des programmes permettant de contrôler l'exécution des domaines utilisateurs. Ils se situent exclusivement dans le domaine 0.
L'application principale de l'espace utilisateur est xend (pour *Xen daemon**). Xend est démarré en tant que service avec le système d'exploitation et sert d'interface entre les *hypercalls* de l'hyperviseur et les outils de contrôle de Xen. Il est aussi chargé de faire passer les informations du bus d'évènement vers l'espace utilisateur.

La figure 54 représente l'architecture de Xen. Les applications utilisateurs communiquent avec xend sous le mode client/serveur, xend assurant le rôle du serveur. La communication avec le noyau s'effectue sous la forme d'appels systèmes traditionnels Unix. Le noyau fait alors un *hypercall* en fonction de l'appel système et des paramètres reçus.

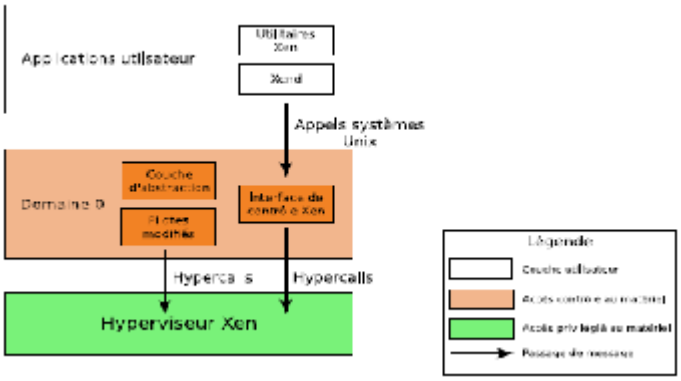


Figure 47 : architecture de Xen/Xend

L'empilement de couches d'abstraction est ici une bonne chose, car cela évite de lier la partie utilisateur à l'interface de programmation de l'hyperviseur. Avec des couches d'abstraction entre eux, les composants de Xen peuvent évoluer séparément, il suffit de maintenir une simple interface entre les couches. [4.1] – [4.2] –[4.3]

3.4.4 Configuration

Xen stocke la configuration des domaines utilisateurs dans des fichiers de configuration, qui sont de simples fichiers textes éditables par l'administrateur.
Le fichier de configuration d'un domaine utilisateur comprend toutes les informations nécessaires à Xen pour pouvoir démarrer le système :

– le noyau à démarrer ;
– l'emplacement des fichiers du système invité (partition dédiée, LVM, répertoire...

– la quantité de mémoire vive à allouer ;
– l'adresse IP de la machine.

```
kernel = "/boot/vmlinuz-2.6.16.38-xenU"
memory = 256
name = "ubuntu_feisty_domU"
disk = ['tap:aio:/home/pchaganti/xen-
images/ubuntu_feisty_domU.img,sda1,w','tap:aio:/home/pchaganti /xen-
images/ubuntu_feisty_boot_domU.img,sda2,w','tap:aio:/home /pchaganti/xen-
images/ubuntu_feisty_swap.img,sda3,w']
vif = [ 'mac=00:16:3e:00:00:10, bridge=xenbr0' ]
  root = "/dev/sda1 ro"
EOF
```

La configuration de l'hyperviseur se fait soit dans le fichier de configuration du chargeur de démarrage, soit dans un fichier de configuration lu par xend au démarrage du domaine 0. En dehors de la configuration initiale lors de l'installation de Xen sur une machine, il n'est pas nécessaire de toucher la configuration de l'hyperviseur.

Cas d'utilisation

Comme énoncé dans le paragraphe consacré aux fonctionnalités de Xen, Xen propose soit de dédier une partition à un système invité, soit de placer l'image du système dans l'arborescence du domaine 0. En pratique, cela implique peu de différences, car même dans le cas d'une partition dédiée, il est facile d'y accéder depuis le domaine 0, de la même façon qu'on accède à une clef USB ou un disque externe. L'attribution à un système invité d'une partition distincte de celle de l'hôte permet de déplacer et de sauvegarder facilement le système invité.

Ce système, couplé à une gestion de haut niveau des partitions avec la technologie LVM (*Logical Volume Manager*, gestionnaire de volumes logiques) permet de s'affranchir des disques physiques pour gérer les partitions accessibles par les systèmes d'exploitation.

En effet, avec une couche d'accès aux disques utilisant LVM, il est possible d'agréger plusieurs disques en un seul espace de stockage, partitionnable librement et indépendamment de la configuration des disques.

On peut alors combiner ces partitions avec du RAID logiciel, pour rajouter de la tolérance aux pannes matérielles. La combinaison LVM plus RAID logiciel est à la fois souple, performante et fiable. C'est d'ailleurs la configuration recommandée par la communauté Xen pour un serveur en production. [4.4]

3.4.5 Avant l'installation de Xen

3.4.5.1 Paravirtualisation et PAE

PAE (Physical Address Extension)

Permet à des processeurs de type x86 de gérer jusqu'à 64 giga-octets de mémoire physique sur des systèmes 32 bits, si le système d'exploitation le supporte. Quel est le rapport entre PAE et Xen en paravirtualisation ?

Ce n'est qu'une question de cohérence. Les possibilités sont indiquées ci-dessous :

	Paravirtualisation et PAE	
Adressages	Dom0 PAE	Dom0 non PAE
DomU PAE	oui	non
DomU non PAE	non	oui

Tableau 7 : Paravirtualisation et PAE

Il faut aussi savoir que les noyaux des Dom0 sont généralement fournis avec le support du PAE. Donc sauf à recompiler, le choix est fait.

Disposez-vous du support PAE ? Exécutez la commande ci-dessous :

```
1.  $ grep "pae" /proc/cpuinfo
2.  flags  : fpu tsc msr pae mce cx8 apic mtrr mca cmov pat pse36 clflush dts acpi mmx
    fxsr sse sse2 ss ht tm syscall nx lm constant_tsc pni monitor ds_cpl vmx est tm2 ssse3 cx16
    xtpr lahf_lm
3.  flags  : fpu tsc msr pae mce cx8 apic mtrr mca cmov pat pse36 clflush dts acpi mmx
    fxsr sse sse2 ss ht tm syscall nx lm constant_tsc up pni monitor ds_cpl vmx est tm2 ssse3
    cx16 xtpr lahf_lm
```

La commande nous retourne 2 lignes (ligne 2 et 3). Une ligne pour chacun de mes processeurs. Ils supportent donc tout les deux le PAE. Si la commande ne retourne rien : pas de support PAE.

3.4.5.2 La virtualisation complète :

Pour voir est ce que la plateforme matérielle accepte la virtualisation complète il suffit : Exécutez la commande ci-dessous :

```
1.  $ egrep "vmx|svm" /proc/cpuinfo
2.  flags  : fpu tsc msr pae mce cx8 apic mtrr mca cmov pat pse36 clflush dts acpi mmx fxsr sse
    sse2 ss ht tm syscall nx lm constant_tsc pni monitor ds_cpl vmx est tm2 ssse3 cx16 xtpr lahf_lm
3.  flags  : fpu tsc msr pae mce cx8 apic mtrr mca cmov pat pse36 clflush dts acpi mmx fxsr sse
    sse2 ss ht tm syscall nx lm constant_tsc up pni monitor ds_cpl vmx est tm2 ssse3 cx16 xtpr lahf_lm
```

VMX c'est pour le support Intel VT/ svm c'est pour le support AMD-V.
La commande me retourne 2 lignes (ligne 2 et 3). Une ligne pour chacun de mes processeurs. Ils supportent tous les deux les instructions vmx. Si la commande ne retourne rien : pas de support des instructions matériel pour la virtualisation complète.

3.4.5.3 Version 32 ou 64 bits ?

Tout d'abord, il faut savoir si on veut installer des DomU 32 et / ou 64 bits.
Ensuite, les tableaux ci-dessous indiquent les possibilités entre Dom0 et DomU suivant le type de virtualisation :

	DomU paravirtualisé	
Adressages	Dom0 32 bits	Dom0 64 bits
DomU 32 bits	oui	non
DomU 64 bits	non	oui

DomU	**complètement virtualisé**	
Adressages	Dom0 32 bits	Dom0 64 bits
DomU 32 bits	oui	Oui
DomU 64 bits	non	Oui

Tableau 8 : version 32 bits 64 bits et la virtualisation

Caractéristiques du processeur :

Pour afficher les caractéristiques du processeur il suffit de taper la commande suivante dans le terminal de système.

```
  # cat /proc/cpuinfo
processor : 0
vendor_id : AuthenticAMD
cpu family : 15
model : 47
model name : AMD Sempron(tm) Processor 3400+
stepping : 2
cpu MHz : 1989.897
cache size : 128 KB
fdiv_bug : no
hlt_bug : no
f00f_bug : no
coma_bug : no
fpu : yes
fpu_exception : yes
cpuid level : 1
```

3.4.6 Matériel requiert et installation de Xen / Domain0

Xen fonctionne actuellement mieux sur l'architecture host x86, et exige une classe de Pentium ou d'autre nouveau processeur. Cette condition inclut mais n'est pas limitée au Pentium pro, Celeron, le Pentium II, le Pentium III, Le Pentium IV, et le Xeon chips d'Intel. En plus, les conditions de processeur peuvent être satisfaites par Advanced Micro Devices (AMD) offres le

produit d'Athlon et de Duron. Ca sera plus performant d'utiliser des processeurs multicore ou hyperthreaded.

Xen soutient les CPUs x86-64. En plus, Xen soutient des invités jusqu'aux 32-way SMP. Le tableau 7 fournit une liste des processeurs x86 appropriés à Xen.

INTEL	
Xeon	71xx, 7041, 7040, 7030, 7020, 5100, 5080, 5063, 5060, 5050.
Pentium	D 920, 930, 940, 950
Pentium	4 662, 672
Core Duo	T2600, T2500, T2400, T2300, L2400, L2300
Core 2 Duo	E6700, E6600, E6400, E6300, T7600, T7400, T7200, T5600
AMD	
Athlon	64 X2 5200+, 5000+, 4800+, 4600+, 4400+, 4200+, 4000+, 3800+
Athlon	64 FX FX-62
Sempron	3800+, 3600+, 3500+, 3400+, 3200+, 3000+
Opteron	Everything starting from Rev. F: 22xx and 88xx (base F), 12xx (base AM2)
Turion 64 X2 dual core	TL-50, TL-52, TL-56, TL-60

Tableau 9: liste de quelques processeurs x86 appropriés à Xen

Remarque

Ces processeurs sont utilisés pour la paravirtualisation, c.à.d. que le Xen ne supporte que des invités OS modifiés. Pour que le Xen applique une virtualisation complète et supporte des invités à OS non modifié il faut que notre plateforme possède des processeurs VT.

Dans notre application on s'est basé plus sur la paravirtualisation à cause de deux choses :

- l'école ne possède pas des plateformes VT.
- Le fait que le Xen soit Open source nous a motivé à le mettre au point malgré les difficultés de mises au point.

3.4.7 Choisissant et obtenant une version de Xen

3.4.7.1 Open Source Distributions

Une manière commune d'obtenir Xen est d'installer une distribution Linux qui contient le support intégré de Xen. Tableau 10 examine certains choix plus populaires des systèmes d'exploitation (Domain0) free/open source.

Operating System	Distribution Support for Xen
Fedora	Xen 3.X packages included since its Fedora 4release.
CentOS	Xen 3.X packages since CentOS 5.
OpenSUSE	Includes Xen 3.X support.

Ubuntu	Includes Xen 3.X packages since 6.10 release (Edgy Eft).
Debian	Includes Xen 3.X packages since Debian 4.0 release.
NetBSD	Host support for Xen 3.X is available in the 4.0 release branch
OpenBSD	Support for OpenBSD self-hosting is near complete.
FreeBSD	Support for using FreeBSD as a Xen host is under development.
Gentoo	A Gentoo package exists for Xen in the Gentoo package Management system, Portage.
OpenSolaris	OpenSolaris xVM available in OpenSolaris developer version, supported by OpenSolaris on Xen community.

Tableau 10: Open Source Distribution

Comme nous avons mentionné, Linux est actuellement le choix le plus commun pour une installation Domain0, mais il n'y a aucune "meilleur" distribution pour Xen. Noter que quelques distributions, telles que Fedora et Ubuntu, fournissent des versions particulièrement des paquets glibc pour s'adapter à des utilisateurs de Xen.

3.4.7.2 Solution commerciale :

Un certain nombre de compagnies fournissent une variété de solutions d'installation, configuration, et services de gestion de Xen.

Commercial Support Solutions	Notes
Citrix XenServer Product Group	Group founded by the original Xen development team. Acquired by Citrix. Leaders of the open source Xen development effort and also offer commercial packages and support.
Virtual Iron	Company predates Xen. Switched to Xen for its hypervisor infrastructure. Offers commercial virtualization management products.
SUSE Linux Enterprise Server 10	The first commercial implementation of Xen in this form is Novell's SUSE Linux Enterprise Server 10 release, which is broadly supported. Red Hat Enterprise Linux 5, released in early2007, will also offer Xen
Red Hat Enterprise Linux 5	RHEL 5 incorporated Xen into the distribution. Though not the first commercial distribution, RHEL 5 provides additional support for tools such as Virtual Machine Manager.

Tableau 11 : Solutions commerciales

3.4.8 Méthodes d'installation du Domain0/Xen :

Le choix de l'environnement pour votre Xen Domain0 est important, car c'est la base sur laquelle tous les invités se reposeront. Historiquement, Linux était le candidat initial pour ce rôle. Ces dernières années, certain système d'exploitation intègre cette solution de virtualisation dans

leur distribution aussi bien qu'Open Solaris. Chacun a ses propres forces et faiblesses comme plateforme pour Xen.

3.4.8.1 GRUB (Le chargeur d'amorçage)

Quelque soit la distribution choisit pour le domaine0, il faut utiliser le Grub, puisque c'est lui le 2ème système qui sera lancer après le bios afin qu'il donne la main au démarrage de Xen.
Xen est lancé par le chargeur d'amorçage. Apres cela Xen chargera le Dom0.
Il faut toujours vérifier le scripte de démarrage de Grub : /boot/grub/grub.conf.
Si les caractéristiques permettant de démarrer le xen sont bien mentionnée c'est bon, si non il faut les introduire. Ces caractéristiques sont :

- ✓ La ligne 1 : titre pour le noyau Xen, montrée dans le menu de Grub, pour le choisir au niveau de démarrage.
- ✓ La ligne 2 : définit la partition root qui sera employée.
- ✓ La ligne 3 : indique le noyau qui doit être employé pour le booting. Dans Xen ceci indique le hypervisor de Xen et pas le noyau dom0.
- ✓ La ligne 4 : indique le noyau qui est employé pour démarrer le dom0 privilégié,
- ✓ La ligne 5 : définit le dossier qui contient l'image initiale de disque virtuel qui est d'abord chargée par le noyau sur l'initialisation [4.5]

En prend comme exemple celle de Fedora Grub :

```
1. title Fedora Core (2.6.19-1.2911.6.5.fc6xen)
2. root (hd0,1)
3. kernel /xen.gz-2.6.19-1.2911.6.5.fc6
4. module /vmlinuz-2.6.19-1.2911.6.5.fc6xen ro root=/dev/VolGroup00/ LogVol00 rhgb quiet
5. module /initrd-2.6.19-1.2911.6.5.fc6xen.img
```

3.4.8.2 Open SUSE

L'installation de Xen pendant une installation fraîche d'Open SUSE est simple. En exécutant l'installateur d'Open SUSE, au cours du choix des options de l'installation exactement la ou on doit personnaliser les logiciels choisis, il faut cocher la case correspondant au virtualisation.

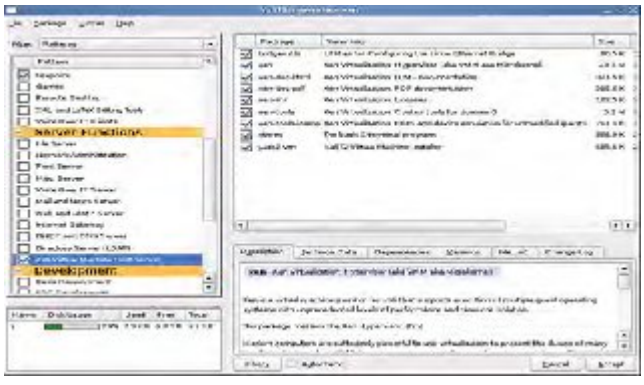


Figure 49 Installation Open SUSE – Xen

La communauté Open SUSE a intégré le support de service Xen et leurs modes d'installation. [4.4] - [4.5]

De même vous pouvez installer xen après l'installation du système en utilisant l'outil de mise à jour.

3.4.8.3 CentOS

Exactement comme Open SUSE, on peut installer les paquets de Xen au niveau de l'installation du CentOS (domain0), il suffit de faire le paramétrage convenable.

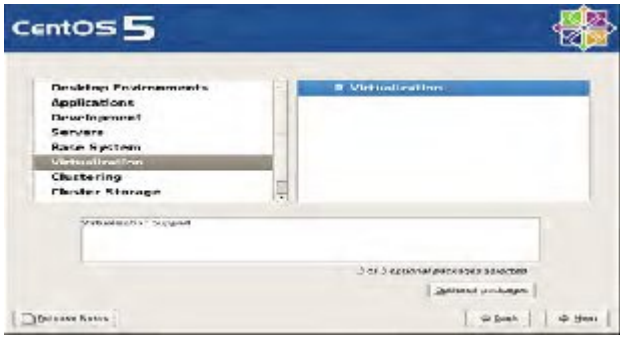


Figure 50 : Installation Centos Xen

Le choix de la virtualisation à partir des menus latéraux gauches et droits est la clef à ajouter le support de Xen.

3.4.9 Installation de Xen sur un système existant :

3.4.9.1 CentOS

Code View:

```
[root@centos]# yum install Xen kernel-Xen virât-manager
```

Les paquets qui seront installé sont :

```
================================================================================
Package Arch Version Repository Size
================================================================================
Installing:
kernel-xen i686 2.6.18-1.2747.el5 base 14 M
virt-manager i386 0.2.5-1.el5 base 357 k
xen i386 3.0.3-8.el5 base 1.7 M
Installing for dependencies:
bridge-utils i386 1.1-2 base 27 k gnome-python2-gnomekeyring i386 2.16.0-1.fc6 libvirt i386
0.1.8-1.el5 base 119 k
libvirt-python i386 0.1.8-1.el5 base 43 k
python-virtinst noarch 0.96.0-2.el5 base 28 k
xen-libs i386 3.0.3-8.el5 base 83 k
```

Transaction Summary

Après avoir vérifié que le fichier de configuration de Grub est bien configuré c.à.d. les paramètres de démarrage de Xen et de domain0 sont bien édités sinon il faut les rééditer.
Pour cela on utilise le Code View:

```
[root@centos]# cat /boot/grub/menu.lst
```

3.4.9.2 Ubuntu :

Les paquets de Xen Debian sont disponibles dans le Debian et l'Ubuntu. Dans notre application on a utilisé Ubuntu 7.10 et 8.04

3.4.9.2.1 Utiliser apt-get pour installer les paquets de Xen d'Ubuntu

Code View:

```
[root@ubuntu]# apt-get install ubuntu-xen-desktop
```

Après que les paquets soient installés avec succès, on redémarre le système et on s'assure qu'on a choisi le kernel approprié de Xen, et pour vérifier que vous êtes sur Xen et que le domain0 fonctionne on utilise les commandes uname -r et xm list.

3.4.9.2 .2 Xen Binary Packages

Tout d'abord Les utiles **iproute** et bridge sont exigés puis il faut télécharger le paquet compressé et faire la décompression ensuite on l'Install. Pour cela il faut procéder comme suit :

Code:

```
[root@ubuntu]# apt-get update
[root@ubuntu]# apt-get upgrade
[root@ubuntu]# apt-get install iproute python python-twisted bridge-utils

[root@ubuntu]# wget http://www.xensource.com/dowland/xen-3.1.0-install-x86_32.tgz
[root@ubuntu]# tar xzpf xen-3.1.0-install-x86_32.tgz
[root@ubuntu]# cd dist
[root@ubuntu/dist]# ls install/lib/modules/2.6.18-xen
[root@ubuntu/dist]#./install.sh

[root@ubuntu]# depmod -a <xen kernel version>
[root@ubuntu]#echo –e "loop max_loop=64" >> /etc/mkinitramfs/modules
[root@ubuntu]# mkinitramfs -o /boot/initrd.img-<Xen kernel version> <Xen kernel version>
```

Finalement il faut vérifier le Grub :

Editer et rajouter dans le fichier /boot/Grub/menu.lst
title Xen 3.0 (Dom0 Based on Linux 2.6 Kernel)
kernel /boot/xen-3.gz
module /boot/vmlinuz-2.6-xen root=<root device> ro
module /boot/initrd.img-<Xen kernel version>

Start les services de Xen Automatiquement :

[root@ubuntu]# update-rc.d xend defaults
[root@ubuntu]# update-rc.d xendomains defaults

Vérifiant que le noyau de Xen fonctionne :

root@ubuntu_Domain0# xm list
Name ID Mem VCPUs State Time(s)
Domain-0 0 2270 2 r----- 85554.2

3.4.9.3 Fedora

Si on utilise Fedora on peut trouver le paquet Xen directement. Il suffit de les choisir au niveau de l'installation du système (exactement comme CentOS). Comme suit

Au niveau du type de l'installation, décochez "Suite bureautique et productivité" et cochez "Personnaliser maintenant".

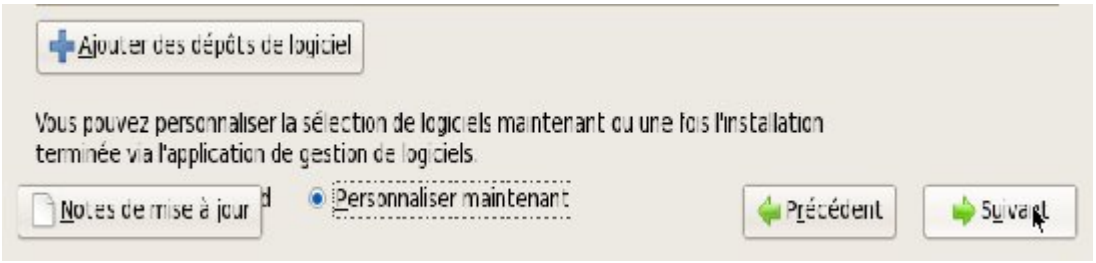


Figure 51 : Installation Fedora Xen

Dans le choix des groupes de paquetages, sélectionnez dans "Système de base" le groupe "Virtualisation" et accédez aux "paquetages optionnels".

Figure 52 : Sélection des paquets Xen

3.4.9.3.1 Installing Xen with yum

Pour l'installation via les lignes de commande il suffit de procéder exactement comme CentOS montré en page 83. Ou bien on peut utiliser le support graphique qui se trouve dans : application>ajouter/supprimer>recherche « Xen »

Figure 53 : Installation Xen via Package Manager

Pour vérifier si vous êtes bien dans le Xen il suffit de taper :
$ uname –r
2.6.24-16.2911.Xen

Maintenant on doit vérifier l'état du service Xend, et l'état des différentes machines fonctionnant au dessus de Xen comme suit :

```
# chkconfi g --list | grep 3:on | grep xen
xend 0:off 1:off 2:on 3:on 4:on 5:on 6:off
xendomains 0:off 1:off 2:off 3:on 4:on 5:on 6:off

# xm list
Name ID Mem(MiB) VCPUs State Time(s)
Domain-0 0 1059 1 r————— 38.
```

Pour plus de détaille http://fedoraproject.org/wiki/Tools/Xen

3.4.10 Comparaison entre les plateformes étudiées

Type	Advantages	Disadvantages
OpenSUSE	Well integrated into the Platform. Easy to install and get going.	Xen code is not quite as current. Software isn't necessarily optimized for your hardware and environment.
CentOS Fedora	Easy installation with relatively strong integration into the distribution	Distribution tracks another upstream development process, which causes a short delay in receiving current code, and the support community is not as large as some of the other end

		user focused distributions.
Ubuntu	A solid performer. Good choice for both package and binary installations. A balance of ease of use and latest release.	Graphical Xen management relies on third party support

Tableau 12: Comparaison entre les plateformes étudiées

3.4.11 Création des machines virtuelles :

3.4.11.1 Virt-manager :

Allez encore un peu de ligne de commande :

```
1. $ su - -c "virt-manager"
2. Mot de passe :
```

Nous sommes connectés au localhost, notre Dom0 est apparu et les boutons « *Nouveau* » et « *Détails* » sont accessibles.

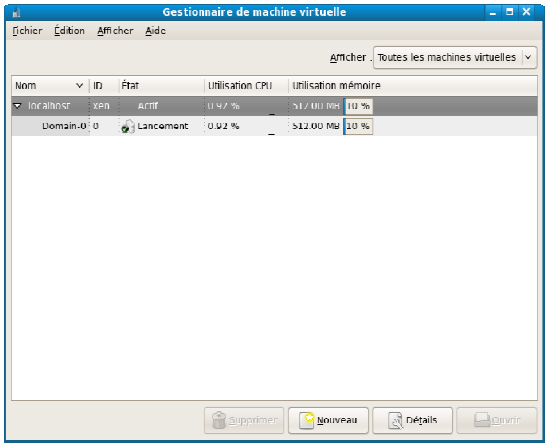


Figure 54 : Les Interfaces de Création des MV

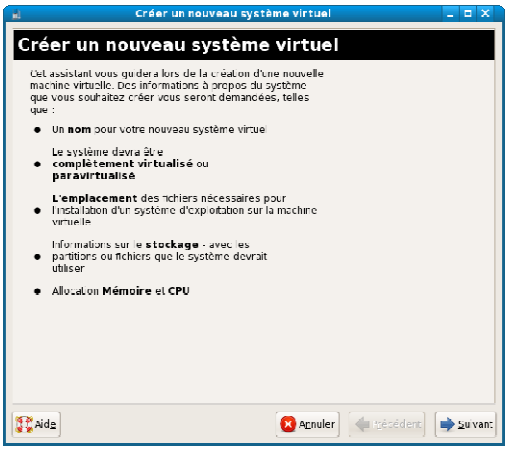


Figure 55 : Virt-Manager

Localiser le médium d'installation

Pour l'installation on peut utiliser un serveur ftp/http locale ou non par exemple le serveur http://192.168.0.10/fedora. Comment accéder au lecteur DVD via un réseau locale c'est simple il suffit de voir le lien suivant [11].
Ou bien un miroir http ou ftp. Par exemple :
http://fr2.rpmfind.net/linux/fedora/releases/8/Fedora/i386/os/

Affectation de l'espace de stockage

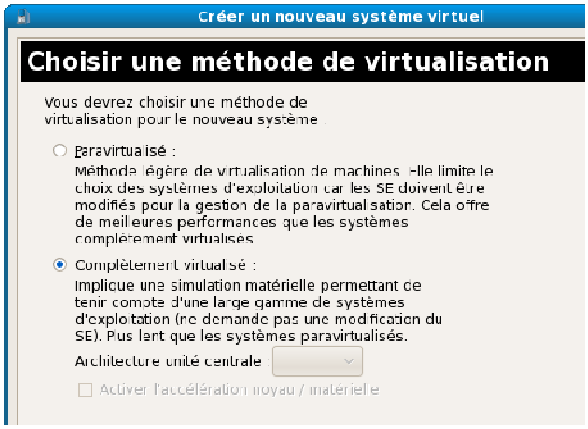
Vous avez le choix d'utiliser un « *Simple fichier* », l'emplacement par défaut /var/lib/Xen/ images/Mon_premier_DomU.img est proposé.

Vous pouvez adapter la taille. Allouer tout de suite la taille demandé ou l'allouer (automatiquement) au fur et à mesure que cela sera nécessaire (c'est transparent, le système s'en occupe). Ou bien on peut choisir « *Partition de disque normale* », vous allez pouvoir indiquer une partition.

C'est selon votre configuration matérielle et ce que vous voulez faire avec vos DomU. « *Suivant* ». Finalement il suffit d'allouer l'espace mémoire pour terminer la création de cette machine virtuelle.
Pour utiliser la souris, il faut cliquer sur la fenêtre du DomU. Pour en sortir, il faut faire « *<Ctrl>+<Alt>* » simultanément.

Un DomU en virtualisation complète, en 3 clics ou presque

Avec la virtualisation complète c'est simple il suffit de cocher la virtualisation complète et pour le média d'installation en choisissant le lecteur DVD, puis il suffit de suivre les procédures de toutes à l'heure pour pouvoir créer le domaine U de Windows



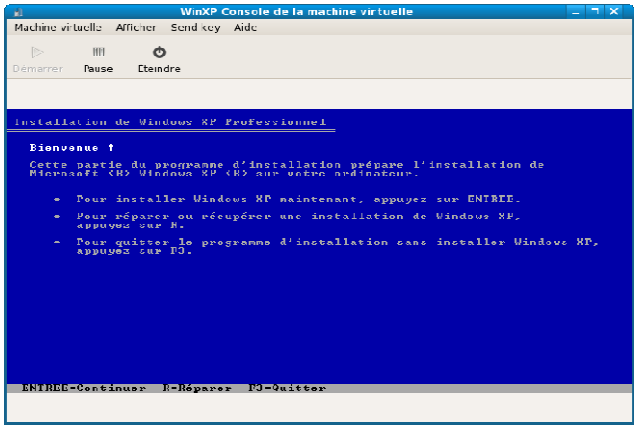


Figure 56 : Virtualisation complète *Figure 57 : Démarrage de console Windows de la MV*

La console suivante nous permet de personnaliser les caractéristiques matérielles de notre domaine invité.

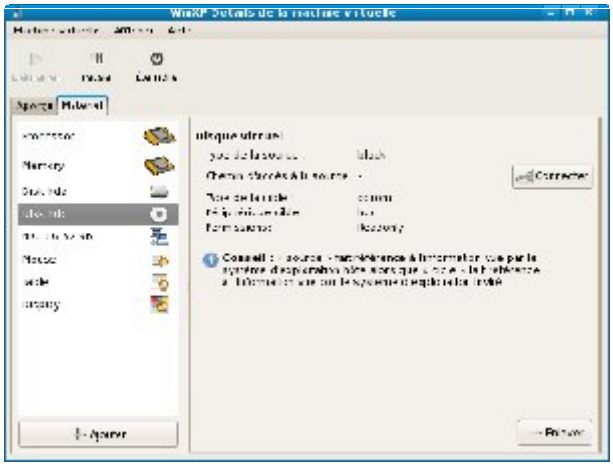


Figure 58 : Interface de la configuration matérielle des MV

Remarque

Au moment du démarrage du Xen, testez la plateforme matérielle. Si votre processeur ne possède pas la VT dans ce cas on ne peut pas faire la virtualisation complète en plus la case de la virtualisation complète dans la console de virt-manager ne sera pas disponible.

3.4.11.2 Téléchargement Des Images Invité compressé :

Pour télécharger les images système voir les sites : **jailtime.org - rpath.com**

Dans ces sites on peut télécharger des images systèmes compressés qu'on utilisera pour créer les systèmes invités comme suit :

Code :

```
# uname –r          /*le noyau employé*/
2.6.24.16-xen
#  wget http://jailtime.org/lib/exe/ download/image_guest.img.tar.bz2    /* telecharger l'image .tar.bz2*/
# tar –C/home/pchaganti/Xen-images –jxvf image_ guest.ig.tar.bz2        /*décompresser l'image*/
# mount -o loop /home/pchaganti/Xen-images/image_guest.img /mnt/img    /*mount le fichier .img*/
# cp –r  /lib/modules/`uname –r`  /mnt/img/lib/modules/     /*copier le module Xen dans le répertoire...*/
#vi  /mnt/etc/fstab                                                    /*vérifier le scripte fstab*/
# vi /mnt/etc/inittab                                                 /*ici on remplace tty0 par console*/

# umount /mnt/img                                                     /*démonter l'image*/
```

Modifier le dossier de configuration de domU pour démarrer l'installation :

```
kernel = "/boot/vmlinuz-2.6.16.38-xenU"
memory = 128
name = "slackware.11-0"
vif = [ 'mac=00:16:3e:00:00:14, bridge=xenbr0' ]
disk = ['tap:aio:/home/pchaganti/xen-images/image_guest.img,sda1,w', 'tap:aio:/home/pchaganti/xen-
images/image_guest.swap,sda2,w']
root = "/dev/sda1 ro"
```

```
# xm create image_guest.cfg  -c                    /* démarrer le systeme*/
```

3.4.11.3 Méthode de création de l'image système :

Il suffit de suivre le script suivant pour créer des images systèmes :

```
[root@dom0]# dd if=/dev/zero of=/xen/images/sid-example.img bs=1024k  seek=4000 count=1
[root@dom0]# mkfs.ext3 -F /xen/images/sid-example.img
[root@dom0]# mkdir -p /mnt/guest_image
[root@dom0]# mount -o loop /xen/images/sid-example.img  /mnt/guest_image
[root@dom0-debian-based]# debootstrap gutsy /mnt/guest_image file:///media/cdrom/
[root@dom0]# dd if=/dev/zero of=/mnt/domU/swap bs=1M count=256
[output omitted]
[root@dom0]# mkswap /mnt/domU/swap
```

3.4.11.4 Creating Virtual Machines (DomU) sous Ubuntu 8.04/7.10:

Sous Ubuntu on trouve aussi l'outil de ligne de commande `Xen-create-image`

```
#   xen-create-image --hostname=xen1.example.com --size=2Gb --swap=256Mb --ide \
   --ip=192.168.0.101 --netmask=255.255.255.0 --gateway=192.168.0.1 --force \
    --dir=/home/xen --memory=64Mb --arch=i386 --kernel=/boot/vmlinuz-2.6.22-14-xen \
   --initrd=/boot/initrd.img-2.6.22-14-xen --debootstrap --dist=gutsy \
   --mirror=http://de.archive.ubuntu.com/ubuntu/ --passwd

#  cat /etc/Xen/xen1.example.com.cfg          /* vérifie le fichier de configuration */
# xm create /etc/Xen/xen1.example.com.cfg   /* start la machine virtuelle*/
# xm console xen1.example.com               /* démarrer la console de cette machine */
# xm list                                   /* voir la liste et l'état des machines virtuelles*/
# xm shutdown xen1.example.com              /* shutdown*/
```

3.4.12 Configuration du Réseau

La configuration peut être assignés manuellement (static) au domain0 et invités exactement comme le cas des machines physiques, ou bien automatiquement en utilisant un serveur DHCP, ce dernier peut être un serveur de réseau physique ou bien une machine virtuelle qui joue le rôle d'un serveur DHCP. Les configurations de réseau et les topologies virtuelles peuvent avoir un grand impact sur la sécurité de système.
Ici on parle de Xen, mais c'est valable pour VirtualBox et VMware

3.4.12.1 Concevoir une topologie de réseau virtuelle :

Puisque Xen peut être employé pour créer une variété de topologies de réseau virtuelles, il est important de concevoir la topologie désirée de réseau avant de commencer la configuration de chaque invité. Pour cela il faut suivre les étapes suivantes :

- ✓ La première étape est de déterminer le numéro et le type d'interfaces réseau physiques

- ✓ Décider pour chaque NIC physique s'il sera partagé avec des domaines d'invité comme bridge, router, or NA Gateway
- ✓ le nombre de sous réseau à créer.
- ✓ Pour chaque invité, déterminer le nombre d'interfaces réseau virtuelles et si chacun se relie à un sous réseau physique ou virtuel.
- ✓ Déterminer comment les adresses MAC (Media Access Control) et les adresses IP seront assignés à chaque interface.
- ✓ Enregistrer la topologie virtuelle de réseau.

3.4.12.2 Bridging, Routing, and NAT:

Xen fournit trois modes de réseau virtuels pour les domaines invité pour accéder au réseau par Bridging, Routing, ou NAT. dans le mode bridging, l'interface de réseau virtuelle (vif) des domaines d'invité est visible à la couche Ethernet externe. En mode routing, le vif n'est pas visible à la couche Ethernet externe, mais l'adresse IP est extérieurement visible. En mode NAT, le vif et les adresses IP ne sont plus visibles à l'extérieure.

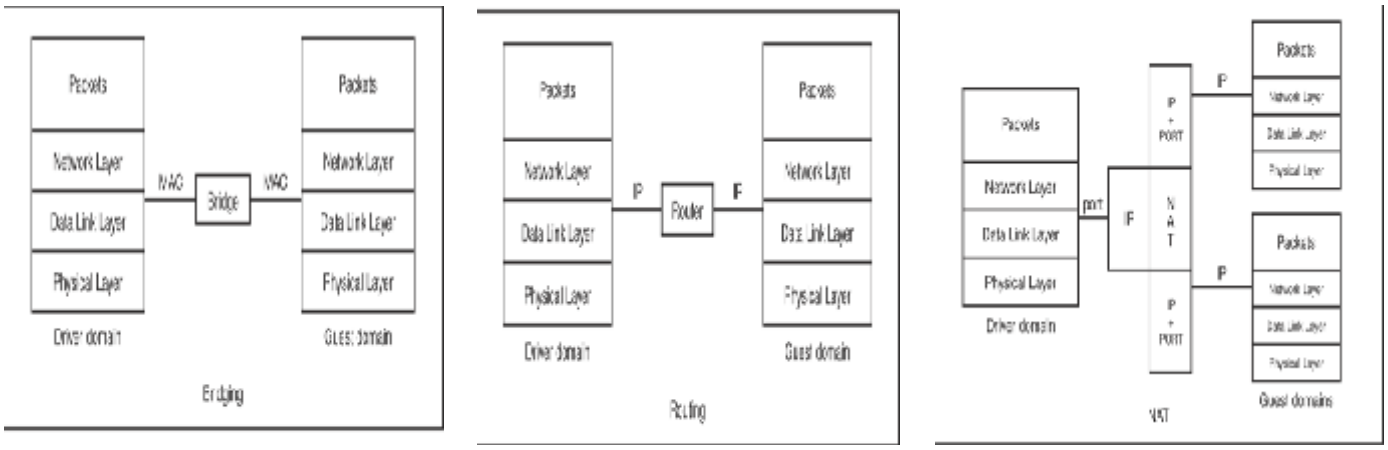


Figure 59 : les différents modes de connexion

En mode bridging, l'outil brctl est utilisé pour créer une interface de pont dans le domaine driver. Une interface physique de réseau est alors fixée au pont. Les vifs principaux des domaines invités de Xen peuvent être attachés à l'interface de pont à l'heure de la création de domaine en utilisant le dossier de la configuration du domaine d'invité. Quand l'interface de pont reçoit des paquets de l'interface physique, elle les transmet par relais aux différents domaines par leurs adresses MAC, ici les domaines d'invité semblent "transparents" à l'Ethernet.

Quand le domaine driver est configuré pour utiliser le routing en utilisant le mécanisme d'iptables de Linux, tous les paquets reçus par la carte d'interface de réseau physique (NIC) sont traités par la couche du réseau du domaine driver. Le domaine driver recherche ses entrées de table IP et expédie alors les paquets à ses domaines invitées selon les adresses d'IP de destination. En mode routing, le domaine driver relie deux segments différents de réseau : le segment interne employé par les domaines invitées et le segment externe se reliant à l'Internet global.

Comparison Aspect	The Bridging Mode	The Routing Mode
ISO Layer	Data Link Layer	Network Layer
Identifier	MAC address	IP address
IP segments	Within a network segment	Connecting two different segments
IP assignment	Dynamic or Static	Static
Setup tools	Bridge utils	Linux iptables
Data Filter	Bridge filter	IP filter
Isolation Properties	Well isolated from the driver domain	Hard to isolate from the driver domain
Configuration	Lasy	A little complicated
Perceived Network Access	Ethernet	Internet

Tableau 13 comparaison bridge et routing modele

Tandis que Linux firewall fournit le filtre de paquet iptables, Linux package bridge-utils fournit aussi le filtre Ethernet, permettant le filtrage a base des adresses MAC sur le pont.

En mode routing, les paquets des domaines invités sont simplement conduits par le domaine driver et envoyés sur l'Ethernet. Mais mode bridging, les interfaces du domaine driver sont données des interfaces physiques de réseau. Le pont fonctionne comme commutateur, reliant les interfaces physiques aux interfaces virtuelles des domaines. Des ponts multiples peuvent être installés dans une machine, avec chaque interface physique fixée à un pont individuel. [4.6] - [4.4]

3.4.12.3 Configuration de réseau sur Xen :

La première étape est d'assigner chaque dispositif physique à un domaine driver pour le contrôler. Ici toutes les fois que nous mentionnons un domaine driver en ce chapitre, nous nous référons au domaine qui possède le dispositif physique de réseau. La configuration la plus simple et mieux examinée emploie Domain0 comme domaine driver.

La deuxième étape est de mettre en application le modèle de partage choisi.
Pour mettre en application le modèle de partage choisi, les mesures de configuration générales suivantes devraient être prises :

- ✓ configurer Xend pour installer le réseau virtuel de mode partage et pour courir le xend dans Domain0
- ✓ configurer les paramètres de réseau dans le dossier de configuration d'invité et **boot** le domaine invité.
- ✓ configurer les interfaces de réseau à l'intérieur du domaine invité
- ✓ Examiner les domaines invités et le réseau virtuel pour assurer qu'ils sont installés correctement.

Fichier de configuration xend

Xen fourni deux types de scripts par défaut : les scripts de réseau et les scripts de l'interface réseau virtuel (vif). Les deux scripts par défaut sont entreposé dans /etc/Xen/scripts. Correspondant à trois modes virtuels différents du réseau de Xen, les scripts par défaut sont network-bridge, vif-bridge, network-route, vif-route, network-nat, and vif-nat.

Il faut toujours s'assurer que le service Xend est start.
Chaque script peut prendre un certain nombre d'arguments, comme suite :

(Network-script 'network-name arg_derivative_1=value argument_derivative_2=value')
Ex: (network-script 'network-bridge netdev=eth1 vifnum=1')

L'argument netdev indique le dispositif matériel de réseau dans Domain0.L'argument vifnum indique le nombre de vif.

3.4.13 Test de performance :

Nous avons utilisé pour les tests de performance le logiciel Geekbench 2.0, et cela n'est pas un pur hasard mais c'est grâce à sa simplicité et surtout du fait qu'il supporte les différents systèmes d'exploitation (Win, linux, Mac …) en donnant la même structure du rapport. Tout cela nous a permis de comparer facilement les résultats.

Geekbench 2.0

Est un benchmark de type « cross-platform ». En d'autres mots, il fonctionne sous différents OS (Windows, MacOS, Solaris et Linux) et différents types de processeurs. Cette nouvelle version est capable d'exploiter plusieurs core et/ou plusieurs processeurs. Geekbench 2.0 teste les performances du processeur ainsi que de la mémoire. Il est possible de soumettre ses résultats en ligne et de les comparer très facilement.

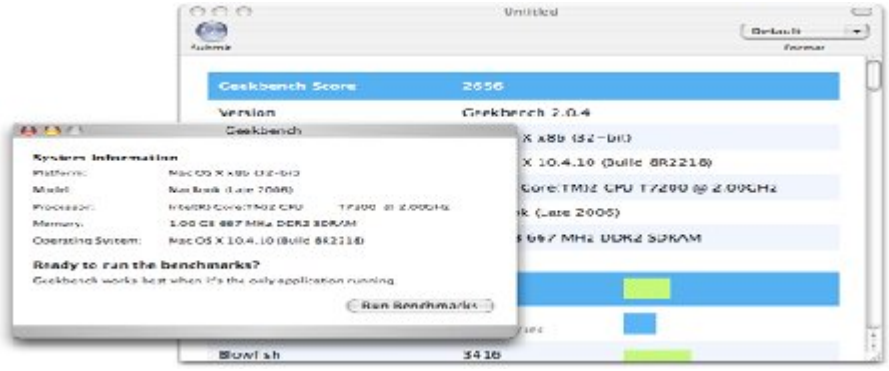


Figure 60 : Geekbench

logiciel de test geekbench

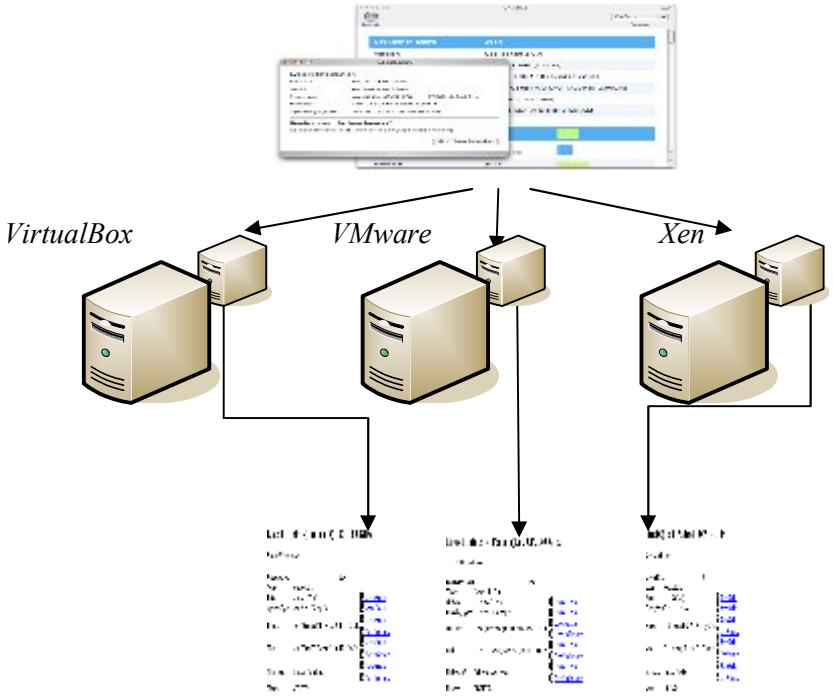


Figure 61: 1ère test de performance avec geekbench

Note : vous trouverez les résultats du test dans l'annexe.

Cette figure résume les résultats des tests de performances appliqués aux différentes solutions.

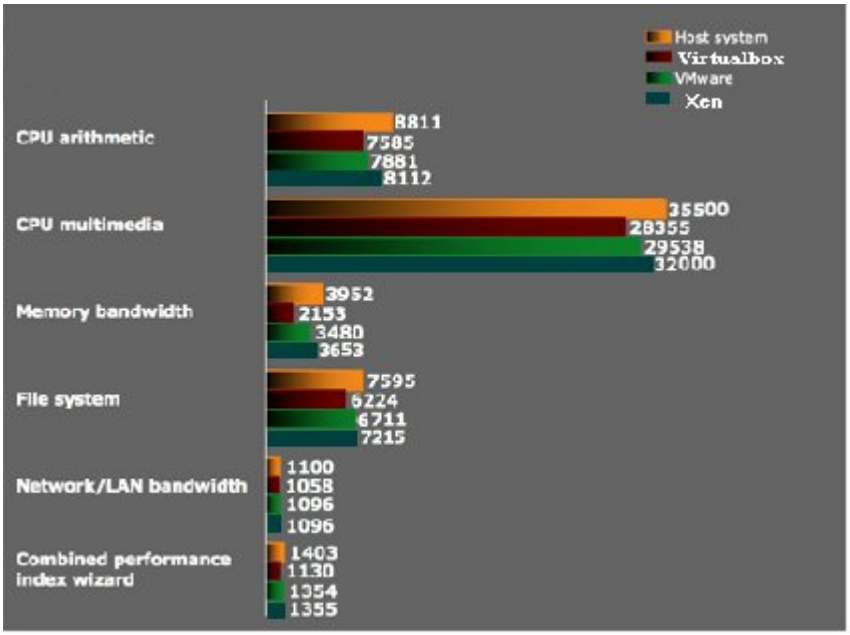


Figure 62 : Résultat du test de performances

3.4.14.1 VMware Workstation 5.5

Sur le plan des performances, VMware est globalement mieux performant que Virtualbox, en particulier grâce à de bonnes performances graphiques. Globalement, en mode bi-processeur, VMware atteint 78% des performances de la machine hôte, et 89% en excluant les tests graphiques.

En mode monoprocesseur, les performances sont bien évidement moins bon, mais restent légèrement au dessus de la barre des 60% et frôlent les 70% en excluant les tests graphiques. Dans ce mode, la mesure du temps était perturbée par l'utilisation d'un processeur dual core sur la machine hôte. Les mesures ont donc été effectuées en isolant VMware sur un seul core. Les tests chronométrables à la main (SuperPi et WinRAR) ont également été effectués sans isolation. Nous avons alors constaté un léger ralentissement (2%).

3.4.14.2 VirtualBox 1.5.6

Les performances sont un peu moindre que celles de VMware (60% avec les tests graphiques, 68% sans), ce qui est plutôt intéressant compte tenu de la gratuité de VirtualBox. Les mesures ont pu être effectuées sans isoler l'application sur un seul core (à l'inverse, avec isolation, la base de temps était incorrecte). Les tests chronométrables à la main (SuperPi et WinRAR) ont également été effectués avec isolation, et, comme avec VMware, l'isolation a apporté un léger gain de performances (1.5%). VirtualBox est également le plus rapide à démarrer Windows XP : à peine 13 secondes, contre une trentaine pour VMware.

3.4.14.3 Xen

Lorsqu'on lui applique les tests de performance. On constate que l'on s'approche 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. Le test donne une performance de 90% par rapport a la machine hôte ce qu'il le place largement en tête.

Finalement en comparant les résultats obtenus avec ce test dans les 3 solutions on trouve bien les résultats connus en théorie c.à.d. que la performance de Xen et meilleure que celle de VirtualBox et VMware .

3.4.15 Conclusion

Indépendamment du coût, VMware est plus performant que Virtualbox. La possibilité d'émuler une machine multiprocesseur lui confère en effet des performances de tout premier ordre et les fonctionnalités offertes sont parmi les plus complètes.

Pour les budgets plus modestes, nous recommandons VirtualBox qui n'a pas à rougir face à VMware sur le plan des performances (en monoprocesseur) malgré sa gratuité et dont les fonctionnalités devraient largement suffire aux utilisateurs grand public. De plus, sa publication sous licence GPL devrait permettre de le voir évoluer plus rapidement que ses concurrents.

Pour la solution native (Xen) on a remarqué que les résultats des tests de performances sont très concluants et dépassent de loin ceux des deux autres solutions, mais ce test n'est pas suffisant on verra dans le chapitre suivant qu'avec le test queryperf ou l'on va éclater la partie Network test.

CHAPITRE 4

Déploiement de Serveurs Internet sur des plateformes virtualisées

5.1 Introduction

Notre objectif dans ce chapitre est le déploiement de serveurs Internet sur une plateforme virtualisée. Apres avoir choisi la solution adaptée au couple DNS &WEB.

La courbe de test de performances précédente nous a montré que le Xen présente de meilleures performances comparées aux autres solutions dans les domaines de la CPU, Mémoire…cependant les résultats concernant les bandes passantes du réseau sont presque identiques et pas du tout significatives.

Ce test reste insuffisant pour le choix de la plateforme de virtualisation ce qui nous oblige à appliquer un test global qui vise à mieux comparer les capacités réseaux de ces dernières.

5.2 Architecture de la plateforme étudié

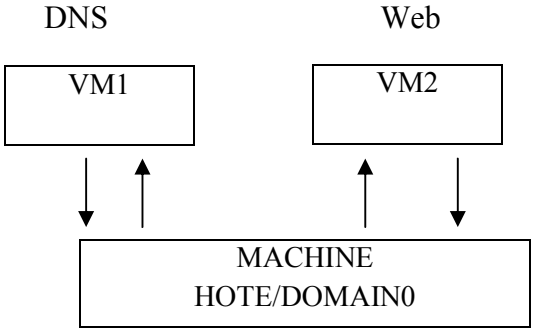


Figure 63 : Architecture de la plateforme étudié

5.2.1 Serveur DNS

Présentation du DNS

Le service **DNS** (*Domain Name System*) est un système hiérarchique distribué permettant la résolution des noms de machines en adresses IP et inversement, comme montrer dans la figure suivante.

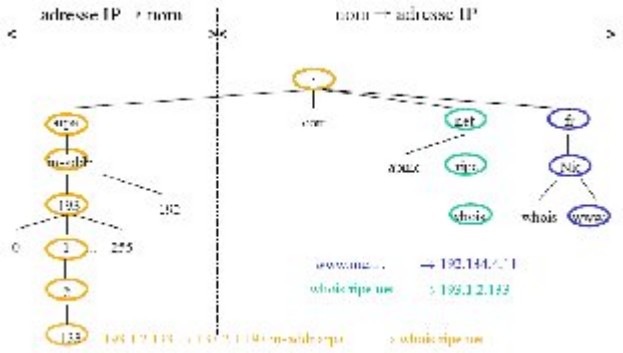


Figure 64 : Exemple de fonctionnement de DNS

Le nom de domaine est séparé en parties appelées zones. Chaque zone part d'un nœud et va vers un nœud où commence une nouvelle zone. Les données pour chaque zone sont stockées dans le serveur de noms, qui répondra aux requêtes d'une zone en utilisant le protocole DNS. Les données associées à chaque nom de domaine sont enregistrées sous forme de Resource Record (RR). [5.1] - [5.2]

5.2.2 Serveur Web :

Un serveur HTTP ou démon HTTP ou HTTPd (HTTP daemon) ou (moins précisément) serveur Web, est un logiciel servant des requêtes respectant le protocole de communication client-serveur HyperText Transfer Protocol (HTTP), qui a été développé pour le World Wide Web.

Un ordinateur sur lequel fonctionne un serveur HTTP est appelé serveur Web. Le terme

« Serveur Web » peut aussi désigner le serveur HTTP (le logiciel) lui-même. Les deux termes sont utilisés pour le logiciel car le protocole HTTP a été développé pour le Web et les pages Web sont en pratique toujours servies avec ce protocole. D'autres ressources du Web comme les fichiers à télécharger où les flux audio ou vidéo sont en revanche fréquemment servis avec d'autres protocoles. [5.2]

5.3 Déploiement d'un serveur DNS

Pour bien commencer, il faut mentionner que l'on va appliquer la configuration DNS dans les 3 solutions de virtualisation VirtualBox, VMware et Xen.

Apres la création des machines virtuelles on procède a la configuration des serveurs DNS comme le montre la figure ci-dessous :

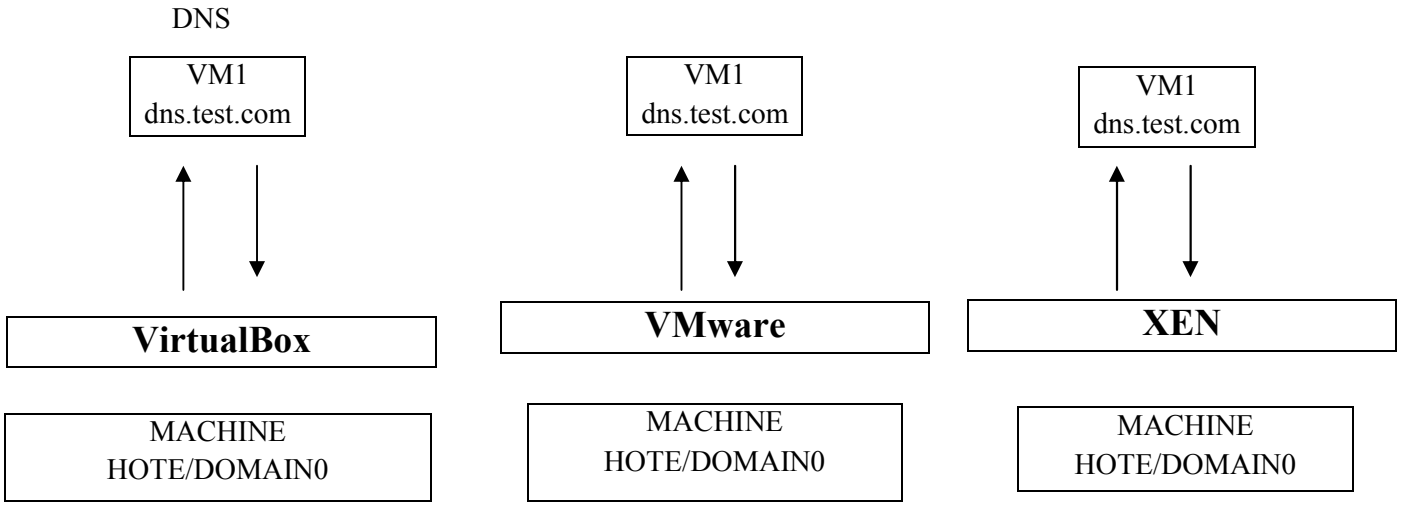


Figure 65 : Schéma de 3 configurations

Configuration bind

Tous d'abord il faut installer le serveur en utilisant la ligne de commande comme suit :
Sous Ubuntu ou Debian : # apt-get install bind9

Après il suffit d'éditer les fichiers de configuration trouvés dans le répertoire /etc/bind/
Ces fichiers sont : named.conf, named.conf.options et db.local
En fin il suffit de taper # /etc/init.d/bind9 restart pour relancer le serveur bind.
La configuration sera simple puisque les DNS seront utilisés seulement dans le test
Note : vous trouverez les fichiers de configuration du DNS dans le CD ci-joint.

5.4 Test de performance :

Maintenant après avoir installé le DNS dans les 3 solutions il nous reste qu'à appliquer le test de performance afin de choisir la meilleure solution pour notre application.
Le test consiste à bombarder notre DNS via un client par des requêtes, pour que ce dernier soit fiable on utilise le logiciel de test **queryperf**

Benchmark avec queryperf

Il est possible de mesurer la charge que votre serveur DNS peut encaisser grâce à un outil inclut dans les sources de Bind9. Celui-ci s'appel "queryperf". Ce test doit être effectué à partir d'une machine cliente UNIX et surtout pas sur le serveur DNS au risque de fausser les résultats.

Il vous faudra donc télécharger les sources de Bind9 :
`ftp://ftp.isc.org/isc/bind9/9.2.3/bind-9.2.3.tar.gz`
Après les avoir extraites allez dans le répertoire des sources de bind puis dans "contrib", vous trouverez un répertoire "queryperf", allez dedans puis faites :

```
#./configure
#make
```

Lisez le README inclut avec queryperf pour comprendre le mode de fonctionnement. Il lui faut un dictionnaire « queryperf-in.rnd » pour qu'il puisse faire des tests, de plus votre serveur devra pouvoir résoudre les noms sur Internet.

On peut éditer un simple fichier pour concevoir le dictionnaire sinon on peut utiliser une autre méthode a consulter dans la référence [5.3]

Le fichier de sortie est "queryperf-in.rnd", vous n'avez plus qu'à lancer queryperf :

```
# queryperf –d queryperf-in.rnd -s ip_de_votre_serveur_dns
```

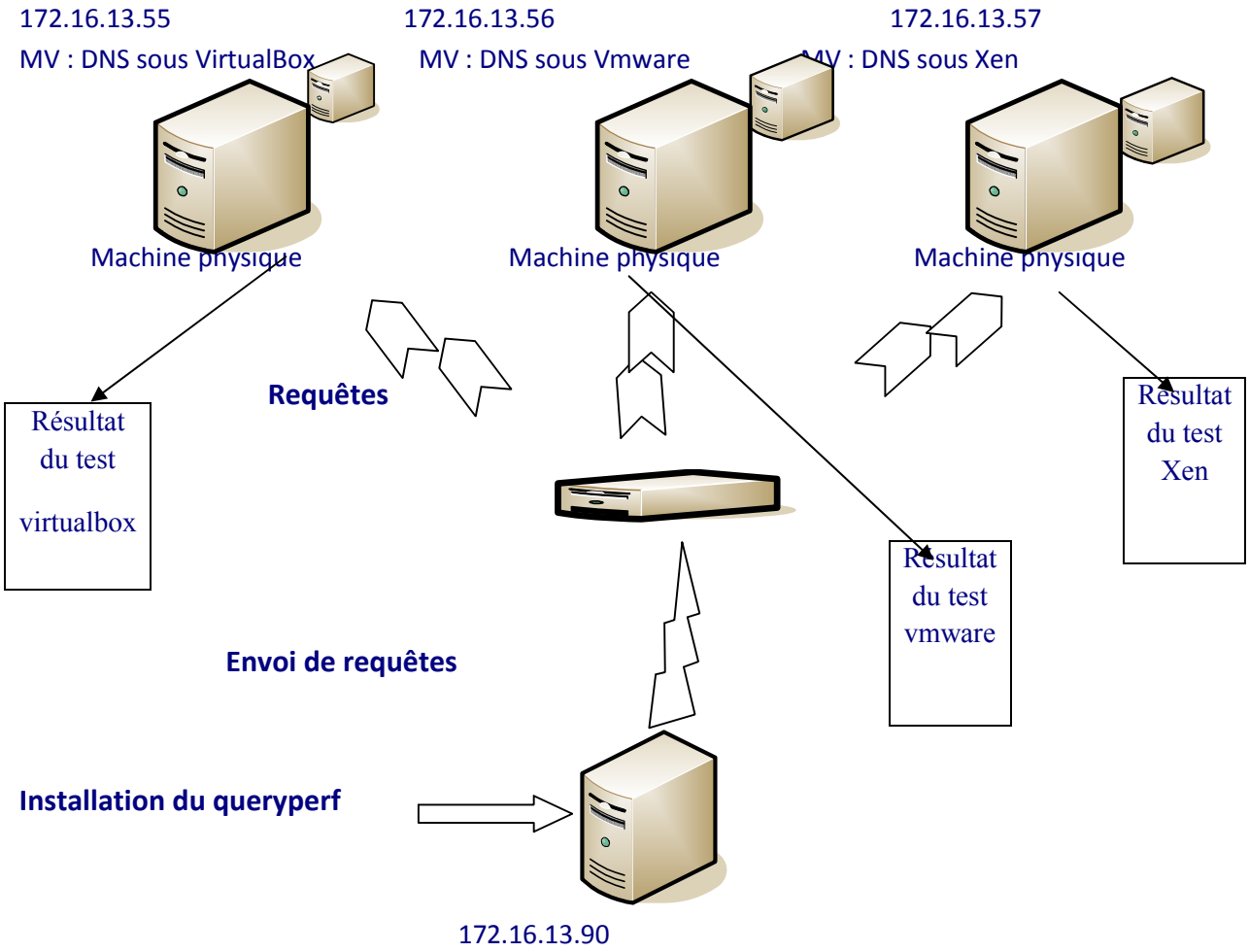


Figure66 (a) : Schématisation du test de performance

Note : vous trouvez les résultats du test dans l'annexe.

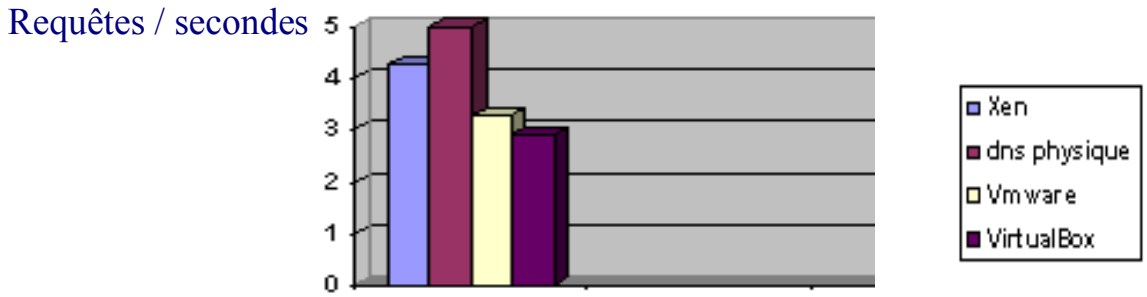


Figure 66 (b) : comparaison du test DNS

Interprétation

Le premier test ne sera pas révélateur car le serveur n'aura pas en cache tous les noms, d'ailleurs ce test prendra un certain temps suivant la bande passante de votre connexion Internet. Il faudra relancer le test 3 à 4 fois afin d'avoir un bon résultat.

D'après la figure on peut dire que :

Les résultats obtenus avec VMware sont meilleurs que ceux de VirtualBox donc il est clair que si on veut choisir l'un des deux on prendra certainement vmware.

Maintenant on les compare avec ceux du Xen on trouve des résultats prévisibles.
C.à.d. que les résultats du Xen sont meilleurs que ceux de Vmware et virtualbox. On voit bien qu' avec les deux tests exercés sur les 3 solutions le Xen donne de meilleurs résultats comparé à ceux de VMware et de virtualbox donc la solution la mieux adaptée a notre application c'est bien le Xen.

Une autre remarque importante c'est que durant ce test on a comparé les résultats du Xen avec un DNS physique les résultat sont très satisfaisants. On a trouvé une ressemblance de 90% entre notre serveur DNS de la machine virtuelle Xen et celle de la machine réelle.

On peut dire maintenant que le Xen est la meilleure solution pour le déploiement de serveur DNS et l'on peut généraliser pour n'importe quel type de serveur internet.

5.5 Applications

Première manipulation

Tous d'abord on va préparer le milieu c.à.d. la plateforme Xen et les machines virtuelles, comme on a fait dans le chapitre 3

Concernant la configuration des DNS on fait exactement comme tout à l'heure sauf pour les fichiers de configuration on utilise ceux de l'école.

Cette manipe a pour but de se familiariser avec les serveurs Internet, nous allons adopter le schéma suivant le mettre au point et le faire fonctionner dans un réseau internet.
Le schéma de configuration sera comme suit :

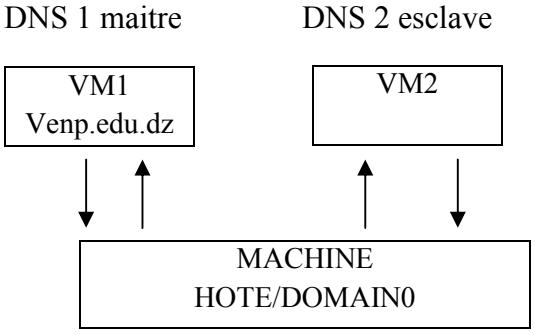


Figure 67 : Schéma de configuration DNS1 maitre+DNS2 esclave

La configuration du DNS esclave n'a pas été faite faute de temps mais elle peut se faire très facilement.

Deuxième manipulation
Configuration du Web

Il faut installer le serveur en utilisant la ligne de commande comme suit :
Sous Ubuntu ou Debian # apt-get install apache
Maintenant il faut éditer les fichiers de configuration trouvés dans le répertoire /etc/apache/
Ces fichiers sont : httpd.conf, modules.conf et access.conf
En suite il faut taper # /etc/init.d/apache restart pour relancer le serveur Web.

Cette manipulation ressemble fortement aux deux premières sauf que pour celle-ci on va installer un serveur web sur la troisième machine virtuelle.
Pour les DNSs on gardera celui de la 1ère manipulation.

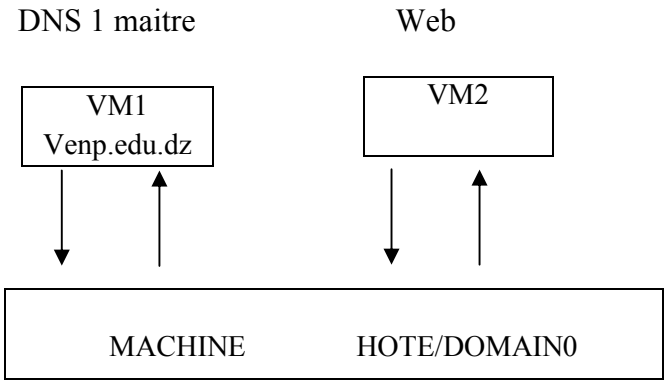


Figure 68: Schéma de configuration DNS1 maitre+DNS2 esclave+Web

La manipulation fonctionne exactement comme prévu.

La conclusion tirée de ces manipulations c'est que l'on peut remplacer n'importe quel serveur par des machines virtuelles faisant le même travail avec autant de performance.

5.6 Conclusion

Nous avons vu en théorie que les machines virtuelles bâties sur un hyperviseur natif sont plus performantes que celles bâties sur un hyperviseur hébergé et comme Xen appartient à la famille native contrairement VirtualBox et VMware, il n'est pas étonnant que les machines virtuelles exécutées au dessus de Xen soient plus performantes et plus rapides que celle de VirtualBox et VMware, exactement comme l'indiquent les résultats obtenus avec les tests de performance.

Finalement ce qu'on peut dire c'est que la meilleure solution de virtualisation pour le déploiement des serveurs internet de point de vue performance c'est le Xen, donc une fois cette

solution maîtrisée. On peut jouer avec les machines virtuelles et les serveurs. Et les résultats de performances de ces serveurs s'approchent de ceux obtenus avec des serveurs physiques. Sans oublier que VirtualBox et VMware restent des solutions très importantes, surtout dans le développement là où la performance n'est pas exigée, car le but est de faire développer un mécanisme ou un protocole de connexion, on insiste toujours sur ces deux derniers pour une simple raison qui est la simplicité de la mise en œuvre comme on a déjà vu dans les chapitres précédents.

Dans notre application on n'a pas eu la chance de comparer notre solution de paravirtualisation Xen avec une mise en œuvre basée sur une plateforme VT. Par contre nos tests se sont basés sur des comparaisons avec des machines réelles.

Conclusion générale

Apres trois mois d'effort on a pu atteindre notre objectif, qui est la mise en œuvre de Xen et mettre en évidence ces points forts face aux autres solutions étudiées tout en appliquant une méthode basée sur trois principes fondamentaux qui sont la récolte d'informations, l'analyse et la mise en œuvre.

La récolte d'informations consiste à chercher les concepts, les techniques, les intérêts ainsi que l'état de l'art de la virtualisation, qui ont comme objectif de fournir une idée claire et nette sur la virtualisation pour bien maitriser cette solution malgré les difficultés rencontrées pendant et après cette étape.

L'analyse consiste à démarrer sur la base des informations récoltées dans la première étape pour examiner les solutions de virtualisation VirtualBox, VMware et Xen étudiés pour la première fois toutes en essayant de résoudre les difficultés de mise au point.

La mise en œuvre consiste à faire fonctionner des serveurs Internet avec les machines virtuelles créées, leurs faire appliqués les tests de performances et traiter les résultats de sorte à choisir la meilleure solution.

L'objectif de cette étude est de montrer les vertus de la virtualisation et ses performances dans le domaine des serveurs Internet. Ainsi que les critères de choix d'une solution de virtualisation.

Nous espérons que cette modeste expérience partagée dans ce rapport vous aidera à vous approfondir d'avantage dans le domaine de virtualisation.

Webgraphies

Note : vous trouvez toutes ces documents dans le DVD ci-joint.

[1.4] Article : Xen – using virtualisation techniques in a Grid environment
Dr.Rüdiger Berlich, Marcus Hardt, Dr. Marcel Kunze
www.eu-egee.org

[1.5] http://fr.wikipedia.org/wiki

[2.1] *Les Livres Blancs LINAGORA*
Publié sous licence Creative Commons « CC BY-NC-SA »1 :
http://creativecommons.org/licenses/by-nc-sa/2.0/fr/legalcode

[2.2] Article: Xen – using virtualisation techniques in a Grid environment
Dr.Rüdiger Berlich, Marcus Hardt, Dr. Marcel Kunze
www.eu-egee.org

[2.3] Article : Architectures d'ordinateurs
Dans l'IT Compendium en pages 144–161
www.transtec.fr , www.ttec.nl , www.ttec.be
http://en.wikipedia.org/wiki/X86

[2.4] Article: Vanderpool Technology for the Intel® Itanium® Architecture (VT-i)
Preliminary Specification - Revision 1.0 January 2005

[2.5] Article: Intel® Virtualization Technology and Intel® Active Management Technology in Retail Infrastructure, White Paper - December - 2006 Revision 1.0

[2.6] www.intel.com /technology /virtualization.pdf

[2.7] Article: Intel® Virtualization Technology for Directed I/O Architecture Specification
September 2007- Revision: 1.1 - Order Number: D51397-003
www.intel.com /technology/virtualization.pdf

[2.8] Article: Intel® Virtualization Technology for Directed I/O Architecture Specification
September 2007/ Revision: 1.1 - Order Number: D51397-003

[2.9] AMD64: Survol des améliorations
http://www.xiti.com/xiti.asp?s=139437

[2.10] http://www.x86-secret.com/articles/cpu/k8-2/a64-3.htm

[3.1] innotek VirtualBox R User Manual
Version 1.5.6 - innotek GmbH -Werkstrasse 24
http://www.innotek.de

[3.2] http://www.vmware.com/solutions/
[3.3] http://www.vmware.com/vinfrastructure/
[3.4] http://www.vmware.com/virtualization

[3.5] Systèmes d'exploitation supportés par VMWare (ESX Server).
http://www.vmware.com/pdf/GuestOS_guide.pdf

[3.6] Version 3.5 sera déployé dans le premier semestre 2008.
http://www.vmware.com/pdf/vi3_35/esx_3/r35/vi3_35_25_config_max.pdf

[4.1] Site officiel du support de Xen : http://www.xensource.com/
Site à l'University of Cambridge: http://www.cl.cam.ac.uk/research/srg/netos/xen/
[4.2] Running Xen: A Hands-On Guide to the Art of Virtualization

[5.1] Documentation Apache en français : http://httpd.apache.org/docs/2.0/

[5.2] http://fr.wikipedia.org/wiki/Apache_HTTP_Server
[5.4] http://www.apachefrance.com (La version française du site)
[5.5] livre : Configurer les composants d'un réseau
www.linuxenrezo.org

Bibliographies

[1.1] Livre blanc : La consolidation et la virtualisation d'infrastructures
Eric LIEURE, directeur du marketing et de la communication d'Overlap

[1.2] Livre blanc : État de l'art des solutions libres de virtualisation pour une petite entreprise
Lucas Bonnet — Bearstech / lbonnet@bearstech.com — http://bearstech.com

[1.3] Livre blanc : *Virtualisation*
Equipe Administration Système Smile

[4.3] Professional Xen® Virtualization
William von Hagen

[4.4] Xen Virtualization
A fast and practical guide to supporting multiple operating systems with the Xen hypervisor,
Prabhakar Chaganti

[4.5] the Best Damn Server Virtualization Book Period
PUBLISHED BY Syngress Publishing, Inc.
[4.6] Virtualization with Xen™
David E. Williams Technical Editor - Juan Garcia

الملخص

ال virtualisation تتحقق مهمة جديدة تتمثل في تشغيل عدة أنظمة في نفس الوقت و على نفس الآلة المادية, ذي يؤدي إلى استغلال أمثل لـ لآلات الـserver ذات القدرات في التطور المتواصل.
هذا العمل يأتي لدراسة أنواع الvirtualisation لاستعمالها في أجهزة الserver بدأ بالحلول المستبيّنة Virtualbox ,VMware مرورا بالحل الغير المستّتبيت ألا و هو الXen.
كذلك قمنا بقياس قدرات الأجهزة التخيلية عن طريق برنامج الBenchmark.
بالاضافة إلى ذلك قمنا ببرمجة الDNS,WEB لمدرسة الوطنية المتعددة التقنيات في الأجهزة التخيلية بطريقة الXen.

Résumé

La virtualisation vient apporter une nouvelle fonctionnalité qui est de faire fonctionner plusieurs systèmes d'exploitation en même temps et sur une même machine physique. Ce qui va exploiter de façon meilleure la capacité des serveurs physique qui ne cesse d'augmenter.
Ce travail consiste à étudier les différents types de virtualisation pour le déploiement des serveurs internet. En commençant par les solutions hébergées VirtualBox et Vmware et passant à la solution native qui est le Xen. Toute en leurs appliquant les tests de performances. Ajouter a cela nous avons configuré les serveurs Web et DNS de l'ecole polytechnique dans deux systèmes d'exploitation invités virtualisés sous Xen fedora.

Abstract

The virtualization comes to bring a new functionality, which is to work several operating systems at the same time and on the same physical machine. What will exploit the best physical capacity of the server, which increases every year. This work consists in studying the various types of virtualization for the deployment of Internet server. Starting with the lodged solutions VirtualBox and Vmware and passing to the native solution, which is Xen. Applying to them the benchmarks.
In addition, we have configured Web and DNS server of Polytechnic school in two guest operating systems virtualized under Xen fedora.

MOTS CLEFS

virtualisation -vmware – virtualbox - xen – serveur internet – machine virtuelle