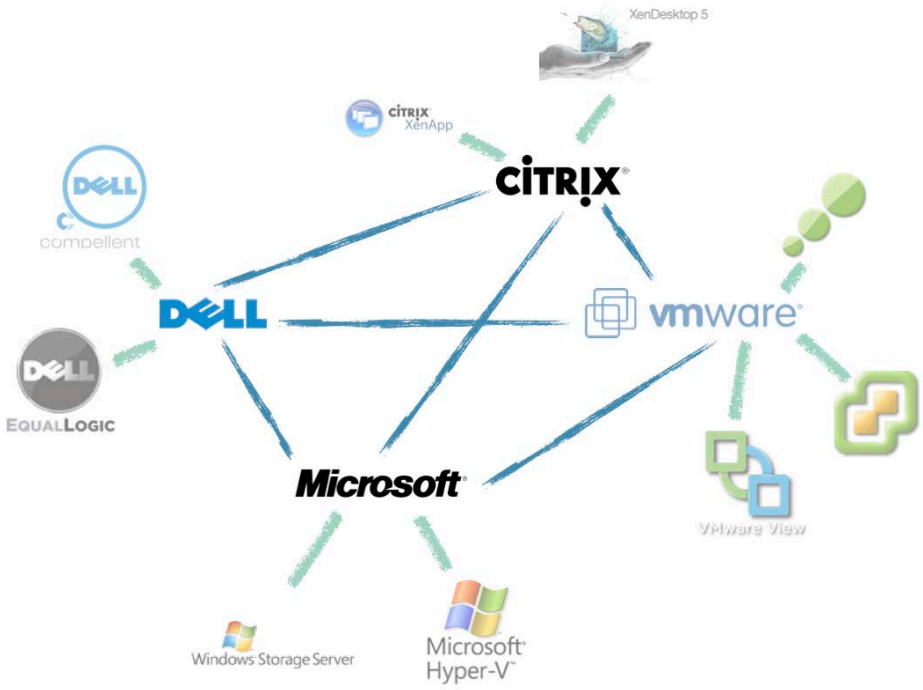
Haute Ecole de Gestion de Genève (HEG)

La virtualisation des systèmes d'information

Mémoire de Bachelor



Réalisé par : Lionel Berger
Directeur de mémoire : Peter Daehne, professeur HES

À Genève, le vendredi 28 septembre 2012

Déclaration

Ce mémoire de Bachelor est réalisé dans le cadre de l'examen final de la Haute Ecole de Gestion (HEG) de Genève, en vue de l'obtention du titre de Bachelor of Science HES-SO en informatique de gestion. L'étudiant accepte, le cas échéant, la clause de confidentialité. L'utilisation des conclusions et recommandations formulées dans le travail de Bachelor, sans préjuger de leur valeur, n'engage ni la responsabilité de l'auteur, ni celle du conseiller au travail de Bachelor, du juré et de la HEG.

« J'atteste avoir réalisé seul le présent travail, sans avoir utilisé d'autres sources que celles citées dans la bibliographie. »

Fait à Genève, le vendredi 28 septembre 2012

Lionel Berger

Remerciements

En premier lieu, je souhaiterais remercier ma compagne, Arielle Moro, qui m'a soutenu inconditionnellement durant les huit semaines que j'ai consacrées à ce mémoire de Bachelor. Son aide a été fort précieuse, tant sur le plan moral qu'au niveau de l'appui qu'elle n'a pas hésité à me conférer pour élaborer ce document.

J'adresse également de vifs remerciements à M. Peter Daehne, professeur à la Haute École de Gestion de Genève, qui a cru en ce projet et qui a décidé de l'encadrer. Sa disponibilité sans égale et ses conseils particulièrement précieux m'ont été d'un grand soutien.

Je souhaite, par ailleurs, remercier mes collègues à la Clinique Générale-Beaulieu, MM. Jean-Marc Barone, Cyril Ackermann et Jacques Grillet, pour les conseils qu'ils ont pu me promulguer et pour avoir toujours fait preuve de compréhension lorsque j'ai eu à consacrer du temps à ce mémoire.

Mes remerciements s'adressent aussi à M. Paul Santamaria, de Dell® Genève, qui a eu la gentillesse de consacrer une partie de son temps à répondre à mes questions.

Enfin, je tiens également à remercier ma famille et mes amis auprès desquels j'ai été moins présent durant cette période et qui cependant n'ont pas hésité à m'apporter leur appui.

Résumé

En 1999, VMware® démocratisait la virtualisation des environnements x86 avec la sortie de son premier produit, WMware® Workstation. Deux ans plus tard, la même société entrait sur le marché des serveurs. L'engouement pour cette technologie ne faiblira dès lors plus, révolutionnant l'infrastructure du centre de données. Aujourd'hui, la virtualisation est au cœur d'une réflexion d'ores et déjà engagée par les professionnels de l'informatique, portant sur les modèles potentiels d'architecture du système d'information au 21e siècle.

Ce mémoire de Bachelor est destiné en premier lieu aux administrateurs systèmes et aux directeurs des systèmes d'information qui, conscients des enjeux auxquels ils seront prochainement confrontés, désirent approfondir leur connaissances en matières de virtualisation. Ces derniers trouveront au sein de ce document, qui s'inscrit dans une approche généraliste de la virtualisation, les bases nécessaires à leurs réflexions ultérieures.

L'historique de la virtualisation est abordée de manière concise, et ce, dans l'unique but de conforter le lecteur quant au fait que cette technologie n'est pas récente au point d'être immature.

Après un bref état des lieux des principales spécificités d'un environnement virtuel, nous dressons la liste des principaux bénéfices offerts par la virtualisation. Nous évoquons ensuite les différentes techniques de virtualisation et leur mise en œuvre au sein d'un certain nombre d'applications commerciales.

Enfin, nous décrivons les différents domaines au sein desquels la virtualisation peut être mise en place, avant de faire référence à un certain nombre de critères et de bonnes pratiques auxquels il convient d'être attentif lors de la construction d'une infrastructure virtualisée. Le document s'achève sur des considérations relatives à la sécurisation de l'environnement, à sa récupération en cas de sinistre et à sa pérennisation.

Table des matières

Liste des figures

Liste des tableaux

Guide de lecture

Afin que ce mémoire de Bachelor puisse être lu en tout aisance, nous vous invitons à prendre bonne note des indications suivantes :

- Les mots en **gras** s'avèrent particulièrement importants et sont mis en évidence par ce biais ;
- Les guillemets ne sont pas utilisés pour mettre en relief des citations (ce document n'en contient pas) mais ont plutôt vocation à souligner un usage particulier d'un terme ou d'une expression. Toutefois, les noms des articles qui sont cités dans le corps du texte sont entre guillemets ;
- Certains mots ou expressions sont mis en *italique* pour souligner leur appartenance à une langue étrangère. Toutefois les solutions ou technologies, référencées au sein de ce document, qui sont commercialisées par des éditeurs ou autres fabricants sous des appellations anglaises, ne sont pas mis en italique. Les noms des ouvrages cités dans le corps du texte sont, quant à eux, en italique. Quant aux articles cités dans le corps du texte, dont le nom est en anglais, ils sont uniquement mis entre guillemets ;
- Les mots en bleu sont définis dans le glossaire à la fin du document.

1 Introduction

Le présent document s'adresse aux **professionnels de l'informatique**, tels que les administrateurs systèmes, mais également aux membres de la Direction des Systèmes d'Information (DSI), seuls habilités à se prononcer sur les budgets relatifs à l'infrastructure informatique.

Ce document vise à combler les connaissances lacunaires en matière de virtualisation des systèmes d'information dont pourraient faire preuve ces acteurs prépondérants. Il vise à les aider à comprendre l'impact de la virtualisation de l'infrastructure informatique dont ils ont la charge. Pour l'administrateur systèmes, il s'agit plus précisément de la diminution des ressources dévolues à la maintenance, au profit de l'efficacité et de l'innovation. Pour la DSI, il convient d'optimiser les moyens financiers, de réorienter les ressources humaines et de préserver la planète en diminuant la consommation énergétique.

Les DSI se trouvent en effet confrontés au dilemme consistant à devoir fournir chaque année plus de services avec des budgets se réduisant d'autant. L'informatique d'entreprise se doit d'être suffisamment flexible et réactive pour être à même de pouvoir répondre avec efficacité aux exigences des métiers qui sont la source des revenus de l'entreprise.

La quantité de données générées croît par ailleurs sans cesse, alors que les contraintes légales, s'agissant des plans de continuité d'activité et de protection des données viennent alourdir la charge de travail des équipes informatiques. De nombreux centres de données (*Datacenters*) se voient affectés par un manque chronique de place débouchant sur de nouvelles difficultés lorsque des machines supplémentaires sont exigées par les métiers qui font face à de nouveaux besoins. L'énergie constitue à l'heure actuelle un autre sujet de préoccupation, les entreprises ne pouvant parfois pas obtenir de nouvelles augmentations de puissance électrique ou ne le désirant pas pour des raisons écologiques.

Dans ce contexte, nous comprenons que les méthodes habituelles en vigueur aujourd'hui au sein des services informatiques doivent évoluer et qu'une transformation du système d'information est dès lors nécessaire. La virtualisation des serveurs est souvent présentée comme étant la solution permettant de mettre en œuvre un socle technique efficace pour accompagner les besoins du métier cités précédemment, et ce, tout en réduisant les coûts, aussi bien au niveau de l'acquisition de dispositifs qu'au niveau énergétique. Cette technologie, associée à la virtualisation des réseaux et du stockage, permet de disposer d'un *datacenter* virtuel de nouvelle génération.

Le présent document fait donc office de guide destiné à la prise de décision liée à l'adoption d'une telle technologie. Il ne s'agit pas d'un manuel technique s'y rapportant, le degré de granularité y relatif étant limité à la compréhension générale des concepts abordés.

Au sein du chapitre 1, nous aborderons rapidement l'**historique de la virtualisation**, s'agissant tout d'abord des travaux d'IBM® liés à la mise au point d'une solution offrant des capacités optimales de temps partagé. Cette démarche aboutira finalement à l'élaboration du concept de virtualisation.

Cet historique n'a pas pour vocation d'être absolument exhaustif. Les travaux d'IBM® sont abordés car essentiels à la gestation du concept. Nous traiterons ensuite de l'architecture x86 et du « portage » de la technologie sur cette plateforme, et ce, parce que cette architecture occupe une place prépondérante dans le modèle d'infrastructure informatique que nous connaissons aujourd'hui.

Enfin, nous pensons qu'**évoquer les dispositifs candidats à la virtualisation** fait office de passage obligé dans une introduction. Nous jugeons en effet que le lecteur doit savoir d'emblée si la lecture d'un tel document correspond à ses attentes.

Une **feuille de route** clôturera finalement cette introduction, en offrant au lecteur une vue d'ensemble du contenu disponible dans ce document, par souci de clarté et au vu d'éviter toute perte de temps inutile au lecteur.

1.1 Concept en quelques mots

En informatique, la virtualisation consiste à créer une version virtuelle d'un dispositif ou d'une ressource, comme un système d'exploitation, un serveur, un dispositif de stockage ou une ressource réseau. Nous pouvons donc considérer la virtualisation comme l'**abstraction physique des ressources informatiques**. En d'autres termes, les ressources physiques allouées à une machine virtuelle sont abstraites à partir de leurs équivalents physiques.

Chaque dispositif virtuel, qu'il s'agisse d'un disque, d'une interface réseau, d'un réseau local, d'un commutateur, d'un processeur ou de mémoire vive, correspond à une ressource physique sur un système informatique physique. Les machines virtuelles hébergées par l'ordinateur hôte sont donc perçues par ce dernier comme des applications auxquelles il est nécessaire de dédier ou distribuer ses ressources.

Il existe de nombreux domaines d'application à la virtualisation, s'agissant généralement de la virtualisation de **serveur**, de **poste de travail**, d'**application**, de **stockage** et du **réseau**.

Lorsque nous évoquons la virtualisation, il est généralement fait référence à la virtualisation des serveurs, et ce, pour des raisons historiques. Il s'agit en effet du premier domaine d'application à avoir été touché par cette technologie. La virtualisation des serveurs consiste à allouer, à l'aide d'un logiciel *ad hoc*, une partie du matériel composant un serveur à chacune des machines virtuelles que nous souhaitons y voir hébergées simultanément. Chaque machine virtuelle est un environnement virtuel cohérent piloté par un système d'exploitation et fonctionnant indépendamment des autres machines.

L'ordinateur hôte dispose bien évidemment de suffisamment de ressources matérielles pour garantir à ses invités une puissance de calcul optimale et un espace disque adéquat. Un système hôte se compose généralement de plusieurs processeurs multi-cœurs, de plusieurs giga-octets (Go) de RAM (*Random Access Memory*), de plusieurs téraoctets (To) d'espace disque ou d'un accès à un stockage en réseau (NAS), voire à un réseau de stockage (SAN).

1.2 Historique de la virtualisation

La virtualisation voit conceptuellement le jour dans les années soixante avec pour but le partitionnement de la vaste gamme de *mainframes* alors disponibles, au vu d'optimiser

l'utilisation de ces derniers. Les *mainframes* étaient en effet confrontés à des problèmes de rigidité et de sous-utilisation. De nos jours, les ordinateurs basés sur l'architecture x86 font face au même paradigme.

1.2.1 À l'origine, la virtualisation des *mainframes*

La virtualisation n'est pas un concept particulièrement récent puisque sa première implémentation a été mise en œuvre pour la première fois il y a plus de 40 ans par IBM®. Il s'agissait alors de partitionner logiquement les *mainframes* précédemment évoqués, en plusieurs machines virtuelles distinctes, rendant possible un **traitement multitâche**[1]. Ces systèmes représentaient à l'époque de coûteuses ressources. Il était dès lors impératif d'utiliser ce partitionnement pour tirer pleinement parti de l'investissement matériel réalisé.

IBM® cherchait en effet à l'époque à maintenir sa domination sur l'informatique scientifique alors que des projets tels que le *Project MAC* ou le *Compatible Time-Sharing System*, axés sur le **temps partagé** (*time-sharing*), étaient menés par le MIT et suscitaient énormément d'enthousiasme. L'énorme projet lié au System/360[2] avait alors eu pour conséquence d'éloigner IBM® des notions de temps partagé.

La société prendra finalement conscience de son retard en la matière, en grande partie grâce à la déception exprimée alors par la communauté au sujet du fait que le System/360 n'avait jamais été pensé comme un environnement temps partagé. C'est alors que le *Cambridge Scientific Center* (CSC) d'IBM®, lié au MIT et initialement dévolu au soutien du *Projet MAC*, fut mis à contribution, sous l'égide de Robert Creasy, dans le but de rehausser la crédibilité d'IBM® à ce niveau en développant un système d'exploitation adéquat pour le System/360.

Parmi les systèmes proposés, l'IBM System/360-67 offrait des fonctionnalités de temps partagé, tandis que le TSS/360 était un système d'exploitation mettant pleinement en œuvre ladite notion. Le second n'a cependant jamais véritablement été commercialisé. L'alternative au System/360-67 existait néanmoins et visait à utiliser la virtualisation pour parvenir à atteindre les objectifs en matière de temps partagé. Il s'agissait du **CP/CMS** (*Control Program/Console Monitor System* ou initialement *Cambridge Monitor System*).

Ce dernier, mis au point en 1967, différait grandement tant des systèmes d'exploitation existant à cette époque que des travaux issus des autres grands projets d'IBM®. Il s'agissait d'un système dont le code source était libre et auquel tous les clients d'IBM® avaient gratuitement accès.

CP faisait office de **programme de contrôle** (*Control Program*) en créant l'environnement de la machine virtuelle (il s'agit donc d'un VMM), chaque utilisateur disposant dès lors de la simulation d'un ordinateur de la gamme System/360 autonome, faisant office d'ordinateur personnel. La version du CP la plus largement utilisée a été la CP-67.

[1] Exécution simultanée de plusieurs applications et processus.
[2] Le System/360 d'IBM® était un ordinateur de la famille des systèmes de type *mainframe*, mis sur le marché par IBM® entre 1965 et 1978.

CMS (*Console Monitor System*) était un **système d'exploitation mono-utilisateur léger**, conçu pour une utilisation interactive partagée. Eu égard aux caractéristiques précitées, une grande quantité de copies de ce système d'exploitation pouvaient être utilisées simultanément sur chaque machine virtuelle de type CP sans prétériter pour autant la performance de chacune d'entre elles.

Le concept de machine virtuelle CP/CMS peut être considéré comme une importante progression dans la conception de système d'exploitation, et ce, à différents titres :

- En isolant les utilisateurs les uns des autres, CP/CMS a grandement amélioré la fiabilité du système en question, ainsi que la sécurité y relative ;
- En simulant un ordinateur autonome pour chaque utilisateur, CP/CMS était en mesure de faire fonctionner chaque application prévue pour le System/360 dans un environnement en temps partagé et non uniquement les applications conçues à l'origine pour un tel usage ;
- En utilisant CMS comme interface utilisateur primaire, CP/CMS parvenait à atteindre des performances sans précédent en termes d'environnement en temps partagé.

De plus, il était parfaitement possible d'offrir plusieurs systèmes d'exploitation différents, simultanément sur une unique machine.

Les différentes versions de CP/CMS ayant vu le jour sont les suivantes :

- CP-40/CMS qui permettra d'établir l'architecture de la machine virtuelle de CP/CMS ;
- CP-67/CMS, nouvelle implantation du CP-40/CMS prévue pour les IBM System/360-67 ;
- CP-370/CMS, nouvelle implantation du CP-67/CMS prévue pour l'IBM System/370. Cette version n'a jamais été commercialisée en tant que telle mais servira de base à la mise sur le marché du système d'exploitation VM/370 en 1972, le System/370 ayant à l'époque été doté de mémoire virtuelle (cf. section 4.1.1.2, Principe de la mémoire virtuelle).

VM peut donc être considéré comme le premier **VMM** ou **hyperviseur** (cf. section 4.4.1.1, Hyperviseur) jamais créé. Sa version actuelle, z/VM, est largement utilisée comme solution de virtualisation pour le marché des *mainframes*.

1.2.2 Nécessité d'une virtualisation x86

Durant les années 1980 et 1990, les architectures x86[3] prennent peu à peu la main sur les *mainframes*. La technologie de virtualisation est dès lors de moins en moins utilisée, à l'exception de quelques solutions destinées principalement aux ordinateurs personnels.

Vers 1988, IGC International[4], une société sise à San Jose en Californie, met au point un système d'exploitation multi-utilisateurs, connu sous le nom de **VM/386**. Un PC pouvait ainsi héberger plusieurs stations de travail virtuelles, chacune d'entre elles étant capable d'exécuter plusieurs programmes DOS ou Windows® simultanément. Le VM/386 peut donc

[3] La famille x86 regroupe des microprocesseurs compatibles avec le jeu d'instructions de l'Intel 8086. Cette série est nommée IA-32 (pour Intel Architecture 32 bits) par Intel® pour ses processeurs à partir du Pentium.
[4] http://www.igcinc.com/company.htm.

être considéré comme le premier VMM capable de créer des machines virtuelles pour un processeur de la famille x86, s'agissant en l'occurrence du premier d'entre eux, soit le 80386.

À la fin des années 1990, à la Stanford University, l'équipe de recherche de Mendel Rosenblum cherche à comprendre plus précisément les spécificités des architectures x86 afin de pouvoir contourner les difficultés qui leur sont liées, ces dernières présentant, malgré leur succès, de nombreuses limites. Pour se faire, ladite équipe se servait du simulateur de système SimOS[5] couplé à une sonde qui permettait d'analyser les appels processeurs, les appels mémoires et les entrées/sorties. Un tel simulateur étant capable de modéliser un système complet, cette technologie s'apparente de très près au concept de la virtualisation et positionne d'emblée l'équipe de Mendel Rosenblum en bonne place dans domaine de la virtualisation.

L'équipe de Mendel Rosenblum parvient finalement à mettre au point un procédé en 1998, permettant de transposer aux environnements x86, le concept de virtualisation des serveurs précédemment mis au point par IBM® (cf. section 1.2.1, À l'origine, la virtualisation des *mainframes*) qui était conforme aux **critères de Popek et Goldberg** (cf. section 4.2.2, Exigences liées à la virtualisation), notamment en ce qui concernait le System/370, puisque dans son cas, toutes les instructions sensibles étaient considérées comme privilégiées (cf. section 4.2.2.2, Classification des instructions processeur).

Ce procédé consiste, en résumé, en la découverte d'un moyen de gérer un certain nombre d'instructions processeur critiques susceptibles de causer des plantages du système. La famille x86 étant omniprésente sur le marché des processeurs, être capable de virtualiser cette architecture constituait véritablement une avancée capitale.

Mendel Rosenblum, sa compagne Diane Green, Edouard Wang, Edouard Bugnion et Scott Devine s'unissent d'ailleurs par la suite pour fonder la société **VMware®**. Créée l'année même, cette société réalisa un coup de force en mettant la virtualisation à portée de tous. C'est en effet à partie des années 2000 que l'ensemble des acteurs du marché finit par s'intéresser à la virtualisation, soit un an après le lancement du premier produit de VMware®, VMware Workstation™ 1.0. Cette société est donc indissociable de la virtualisation, en grande partie pour avoir mis au point une solution optimale de virtualisation des environnements x86.

1.2.3 VMware®

L'entreprise se présente au marché en **1999** avec la sortie de **VMware Workstation™ 1.0** pour Linux™ et Windows®. De gros acteurs s'impliquent rapidement dans le projet VMware®. Compaq®, IBM®, Dell®, Intel® aident la jeune entreprise sous différentes formes.

En **2001**, les trois premiers précités et HP rejoignent le programme partenaire de VMware®. La même année, le produit aujourd'hui phare de l'éditeur, l'**hyperviseur ESX™**, voit le jour.

[5] Simulateur de système développé à la Stanford University à la fin des années 1990 au sein du groupe de recherche de Mendel Rosenblum. Il permettait de simuler une architecture informatique à un tel niveau de détail que la pile logicielle complète d'un système réel pouvait fonctionner sur ce dernier sans aucune modification.

La notoriété de l'entreprise s'étend et l'année suivante elle compte plus d'un million d'utilisateurs.

2003 est un tournant pour VMware® avec l'extension de son portefeuille de produits et l'arrivée de **vMotion™**, un logiciel qui permet de transférer les machines virtuelles d'une machine physique à une autre. La même année, **EMC®**[6] achète VMware® qui préserve toutefois une certaine indépendance organisationnelle et une division R&D qui le demeure également.

2007 représente une étape importante pour l'entreprise avec son **introduction en bourse**. Cisco®[7] et Intel®[8] prennent des parts dans VMware®, liens capitalistiques qui renforceront les partenariats technologiques par la suite.

En **2010**, VMware® rachète **Zimbra**®[9] à Yahoo® et s'engage ainsi résolument dans le monde de l'informatique en nuage.

1.3 Candidats à la virtualisation

Lorsque nous abordons le sujet de la virtualisation, les intéressés se demandent souvent ce qui peut être virtualisé et ce qui ne peut pas l'être. Nous pouvons considérer que **tout ce qui sous-utilise la charge matérielle disponible peut être virtualisé avec succès**. Il en va ainsi, notamment, des serveurs web, des serveurs de messagerie, des différents serveurs réseau (DNS, DHCP, NTP), des serveurs d'applications (Tomcat™, etc.), sans oublier les serveurs de base de données. Les serveurs constituent ainsi le cœur de l'infrastructure virtuelle mais ne sont pas les uniques dispositifs à pouvoir être virtualisés.

Il n'existe par ailleurs aucune restriction quant aux systèmes d'exploitation que nous pourrions utiliser sur les machines invitées. Tant les systèmes Windows® que les Linux™, Solaris™ ou autres font parfaitement l'affaire.

Nous précisons qui si les dispositifs susmentionnés font de parfaits candidats, les fondations de l'infrastructure doivent être sciemment planifiées. En effet, les besoins doivent être clairement identifiés et le périmètre établi avec circonspection. Pour se faire, un **audit préliminaire**, tant au plan technique qu'opérationnel doit être envisagé.

Ce dernier nous aidera à faire le bon choix en matière de dimensionnement des serveurs et du stockage, mais également en termes de réseau de stockage, de paramètres de performances ou de sécurisation des données.

[6] EMC® est une entreprise américaine de logiciels et de systèmes de stockage fondée en 1979 à Newton (Massachusetts). L'entreprise est leader mondial du stockage.
[7] Cisco® est une entreprise informatique américaine spécialisée, à l'origine, dans le matériel réseau (routeur et commutateur Ethernet).
[8] Intel® Corporation est une entreprise américaine produisant des microprocesseurs - elle est à l'origine du premier microprocesseur x86 -, des cartes mères, des mémoires flash et des processeurs graphiques notamment.
[9] Zimbra® est une solution Open Source pour messagerie et partage de calendrier.

Nous aborderons ces différents éléments en détails à la section 6.1.1, Capacités nécessaires.

1.4 Feuille de route

Le présent document est structuré de la manière suivante :

- Le chapitre 2, **Spécificités d'un environnement virtualisé**, met en évidence de manière succincte les aspects fondamentaux d'une infrastructure virtualisée, et ce, afin que le lecteur puisse se faire une rapide idée de ce à quoi il serait confronté s'il faisait le choix de la virtualisation ;
- Le chapitre 3, **Bénéfices de la virtualisation**, évoque les différents bénéfices qui peuvent être obtenus par le biais de la virtualisation de tout ou partie du système d'information. Ces dits bénéfices sont grandement liés à la virtualisation des serveurs ou des stations de travail. À ce titre, la liste figurant au chapitre 2 en référence un nombre substantiel. Cette dernière n'est toutefois pas consacrée qu'à ce domaine d'application en particulier ;
- Le chapitre 4, **Fondamentaux technologiques de la virtualisation**, a pour objectif initial de permettre au lecteur de se réapproprier certaines notions relatives à des éléments clés intervenant dans la composition d'un ordinateur, ainsi que des mécanismes ou principes qui régissent ces derniers. Ainsi, le lecteur disposera de toutes les bases nécessaires à la bonne compréhension des techniques de virtualisation. Ces dernières seront également abordées dans ce chapitre, ainsi que certaines applications commerciales qui en découlent ;
- Le chapitre 5, **Domaines d'application**, constitue une vue d'ensemble des domaines d'une infrastructure physique qui contiennent des périphériques susceptibles d'être virtualisés, au-delà de la virtualisation des serveurs abondamment évoquées au cours du chapitre 4 en particulier ;
- Le chapitre 6, **Construction de l'infrastructure virtuelle**, est destiné à accompagner le lecteur tant dans les choix qu'il aura à faire en termes d'achat de serveurs et d'éléments de stockage que dans le choix du réseau de stockage qu'il devra mettre en place. Il s'agit principalement d'une succession de bonnes pratiques et de quelques notions techniques de base, destinées avant tout à faire en sorte que le lecteur ne néglige rien dans son évaluation de l'ensemble des éléments nécessaires à la construction de l'infrastructure virtuelle. Enfin, ces informations, destinées à la construction de l'infrastructure virtuelle portent principalement sur le cœur de cette dernière, à savoir la virtualisation des serveurs et le stockage partagé (les éléments propres à la virtualisation du stockage ne sont pas abordés dans ce chapitre) ;
- Nous terminerons ce document par le chapitre 7, **Gestion de l'infrastructure virtuelle**, qui met l'accent sur certains points essentiels à prendre en compte une fois l'infrastructure virtuelle en place. Les grands principes de la sécurité des systèmes d'information sont rappelés et mis en relation avec les particularités de ce type d'infrastructure. Certains mécanismes permettant d'assurer la disponibilité de l'infrastructure et l'intégrité des données y sont décrits. Certaines notions destinées à accompagner le lecteur dans la mise en place d'un plan de reprise d'activité qui tienne compte des particularités propres à la virtualisation y sont également abordées. Nous y évoquerons également la planification budgétaire relative à l'usage de l'infrastructure virtuelle et les moyens d'évaluer les coûts réels des machines

virtuelles. Enfin, certaines perspectives liées à l'évolution de la virtualisation au sein des centres de données, et leurs impacts sur le système d'information sont évoquées.

2 Spécificités d'un environnement virtualisé

Ce chapitre a été conçu pour faire en sorte d'aider le lecteur à prendre conscience des **aspects fondamentaux d'une infrastructure virtuelle**. Le niveau de détails choisi pour mettre en exergue les éléments contenus dans ce chapitre est volontairement faible, de telle sorte que le lecteur puisse bénéficier rapidement d'une vue d'ensemble. Chacun des éléments figurant dans ce chapitre sera toutefois abordé en détails ultérieurement.

Après avoir attiré l'attention du lecteur sur les candidats à la virtualisation, nous voulions nous assurer que ce dernier puisse d'emblée se faire une idée de l'infrastructure qu'il aurait à gérer, une fois prise la décision d'évoluer vers la virtualisation.

2.1 Évolution du modèle de centre de données

Le remplacement des serveurs physiques par des instances virtuelles encapsulées dans des fichiers modifie notablement le modèle habituel du centre de données. Avec la virtualisation, le modèle distribué comprenant un nombre important de petits serveurs physiques fait place à un modèle centralisé et consolidé sur un même site. **Le stockage, puisqu'il héberge les machines virtuelles, devient la pièce maîtresse de l'infrastructure** et doit donc être capable de hautes performances et fournir des solutions à même de sécuriser les données.

Ce changement pousse les entreprises à redéfinir complètement leurs infrastructures existantes.

2.2 Machines virtuelles

Il va de soi que dans un environnement virtuel, l'administrateur gère des machines virtuelles. Une machine virtuelle représente l'ensemble de l'état d'une machine physique (i.e. le système d'exploitation, appelé système d'exploitation invité, avec les applications et les données).

Dans ce type d'infrastructure, aucun problème de portabilité n'est à prévoir, une machine virtuelle étant absolument identique à une machine physique. Un administrateur dispose d'une telle granularité de configuration qu'il lui est permis de fournir à une machine virtuelle les ressources dont elle a besoin de façon très fine.

Le fait que les machines virtuelles soient totalement isolées les unes des autres (systèmes d'exploitation, registre, application et données), les immunise dans le cas d'une infection d'une d'entre elles par un virus. Il en va de même avec le crash d'un système d'exploitation. Cette protection entre machines virtuelles n'a jamais été mise en doute à ce jour.

L'état complet d'une machine virtuelle est contenu dans des fichiers. C'est ce que nous entendons par encapsulation. Cette dernière autorise une grande souplesse d'utilisation en simplifiant les sauvegardes, les copies et les plans de reprises d'activité.

Il convient également de préciser qu'une machine virtuelle est totalement indépendante de la machine qui l'héberge. Il n'existe dès lors plus aucune contrainte matérielle, facilitant d'autant la migration vers de nouvelles plateformes de virtualisation. Ainsi, lors du renouvellement du serveur hôte de machines virtuelles, il suffit de déplacer la machine

virtuelle sur ce dernier, en toute simplicité. Il n'est aucunement nécessaire de procéder à une nouvelle installation sur le nouveau dispositif.

2.3 *Provisioning* instantané

Grâce à la virtualisation, les méthodes habituelles de gestion des serveurs est révolutionnée. Par *provisioning* instantané, nous entendons la possibilité de mettre en service un nouveau serveur facilement, en quelques minutes, alors qu'il fallait parfois plusieurs semaines pour le faire en environnement physique.

L'échelle du temps en est durablement modifiée, ce qui a pour effet de permettre aux entreprises de s'adapter très rapidement aux changements et évolutions liés aux affaires : fusion/acquisition, création de nouveaux services, mise en place de nouveaux projets, etc. Le service fourni aux utilisateurs est dès lors grandement amélioré, les besoins spécifiques étant rapidement traités.

2.4 Regroupement des ressources en clusters

Les serveurs hôtes de machines virtuelles peuvent être regroupés dans une entité appelée *cluster*, autorisant une gestion globale et non unitaire de l'infrastructure. Des fonctionnalités évoluées de haute disponibilité sont ainsi fournies, au même titre qu'une répartition de la charge sur l'ensemble des serveurs du *cluster* en cas de forte activité. La tâche des administrateurs est simplifiée par ce fonctionnement qui, par ailleurs, garantit des niveaux de services pour les applications.

2.5 Qualité de service (QoS)

La qualité de service peut être mise en place dans les infrastructures virtuelles, et ce, pour garantir que chaque machine virtuelle dispose des ressources dont elle a besoin en fonction de la criticité du service qu'elle héberge. La QoS peut être paramétrée tant au niveau de la machine virtuelle elle-même qu'au niveau de l'hyperviseur ou du *cluster*.

3 Bénéfices de la virtualisation

Le multi-cœur, le 64 bits, tout comme la gestion d'une quantité très importante de mémoire, sont autant d'évolutions dans la technologie des serveurs ces dernières années. Installer un seul système d'exploitation sur un serveur qui serait susceptible d'en supporter plusieurs dizaines est devenu aujourd'hui totalement injustifiable.

La virtualisation s'avère être une des technologies la mieux à même de tirer parti des processeurs multi-cœurs, puisqu'elle offre un **haut niveau de consolidation**, soit la possibilité d'héberger un nombre important de machines invitées sur un serveur hôte.

Un serveur est actuellement considéré comme étant dix à douze fois plus puissant en termes de performances qu'un de ses homologues d'il y a quatre ans. Dès lors, il devient même possible d'envisager la virtualisation de serveurs hébergeant des applications considérées comme stratégiques, telles qu'Oracle® ou SAP®, moyennant toutefois quelques configurations spécifiques.

Les fabricants ont embarqués certaines technologies au niveau du matériel qui permettent de gérer nativement la virtualisation, s'agissant notamment de l'adaptation des processeurs (cf. section 4.3.3, Assistance matérielle).

Les baies de stockage sont capables de s'interfacer avec les API (*Application Programming Interface* ou interface de programmation) fournies par les constructeurs afin de décharger le serveur hôte de machines virtuelles de certaines tâches liées au stockage.

Certains commutateurs permettent de simplifier la gestion réseau en environnement virtualisé, à l'instar du Nexus 1000v™[10] de Cisco® (cf. section 5.5.2, Commutateur).

Au vu de ces éléments, la virtualisation peut être considérée comme une alternative non seulement viable mais particulièrement pertinente aux architectures classiques en vigueur au sein de bon nombre de sociétés à l'heure actuelle. Le présent chapitre vise à mettre en exergue les bénéfices octroyés par cette technologie et leurs impacts sur l'entreprise.

Nous avons choisi l'option d'aborder les bénéfices octroyés par cette technologie avant d'évoquer les aspects techniques y relatifs, la mise en œuvre, les domaines d'application ou même la construction d'une infrastructure virtuelle.

Nous jugeons en effet indispensable que le lecteur soit convaincu de la pertinence de la technologie en question avant de poursuivre sur les aspects susmentionnés.

Il convient de préciser que les avantages listés dans le présent chapitre **ne constituent pas une liste exhaustive**. Pour des raisons pratiques, certains d'entre eux sont explicités au sein d'autres chapitres, afin d'en conserver la cohérence. Il en va ainsi de certains avantages liés à la virtualisation du stockage ou des applications, abordés au chapitre 5.

[10] http://www.cisco.com/en/US/products/ps9902/index.html.

3.1 Réduction des coûts CAPEX/OPEX

Quoiqu'essentiel à la bonne marche d'une société, le système d'information, ou plus précisément l'infrastructure y relative, est bien souvent perçu comme un centre de coût par les dirigeants. Ces derniers exigent des responsables informatiques qu'ils garantissent un certain niveau de service malgré le fait que les demandes évoluent constamment à la hausse. Les budgets doivent par contre rester identiques, voire évoluer à la baisse.

Or, au sein d'une infrastructure physique, le taux d'utilisation moyen des serveurs d'un centre de données est estimé à moins de 10% pour 80% d'entre eux. Quant au centre de données eux-mêmes, ils arrivent à la limite de leur capacités en matière d'emprise au sol, d'alimentation et de climatisation.

La raison fondamentale de ce gaspillage résulte du fait que les sociétés ont massivement investi dans des serveurs x86 destinés pour chacun d'entre eux à **n'héberger qu'une seule application**, et ce, pour limiter les interruptions au niveau de la production en cas de panne. Cette prolifération de serveurs physiques augmente très fortement, pour ne pas dire de manière astronomique, le coût d'exploitation.

Nous estimons d'ailleurs à 70% le temps consacré par les administrateurs systèmes à des opérations de support ou de maintenance, ces dernières n'apportant aucune valeur ajoutée à l'entreprise.

Les coûts indirects, tels que les coûts de gestion, d'administration et de consommation électrique atteignent des montants éminemment supérieurs à ceux liés à l'acquisition des serveurs eux-mêmes, à tel point qu'ils peuvent représenter jusqu'à trois fois le coût initial du matériel composant l'infrastructure.

Les entreprises perdent des sommes substantielles en raison du coût élevé du maintien en condition opérationnelle (MCO) de l'infrastructure. Une inefficacité opérationnelle est engendrée, pesant sur l'innovation et la gestion de nouveaux projets, pourtant capitaux pour l'évolution de la société.

Dans un tel contexte, la virtualisation permet aux responsables informatiques de consentir à la modernisation du système d'information de l'entreprise, tout en réduisant les coûts et en satisfaisant aux exigences mentionnées plus haut.

Les entreprises distinguent en principe deux types de coûts :

- CAPEX (*Capital Expenditures*), s'agissant des dépenses liées aux investissements et immobilisations (i.e. matériel divers, logiciels, etc.) ;
- OPEX (*Operational Expenditures*), s'agissant des dépenses liées au fonctionnement de la société (i.e. expertise, prestation, conseil, gestion de projets, etc.).

La virtualisation fait résolument partie des leviers permettant de réduire les coûts CAPEX/OPEX.

3.1.1 Diminution du nombre de serveurs

La virtualisation permet en tout premier lieu d'éviter d'avoir à acheter du matériel à chaque fois qu'un nouveau système doit être déployé. Cette affirmation se vérifie aisément lorsqu'il s'agit de mettre en place un nouveau serveur. Il est en effet possible de créer un certain nombre de machines virtuelles par serveur, le taux de consolidation y relatif dépendant des ressources qui doivent être effectivement allouées à chacune des machines en question.

En tenant compte du fait que, conformément aux bonnes pratiques, chaque serveur doit être dédié à une application (afin d'éviter un arrêt complet de la production en cas d'intervention sur la machine concernée par une panne éventuelle), nous pouvons facilement prendre conscience de l'avantage dont il est question. Ceci est d'autant plus vrai selon que la société concernée fera le choix d'un mode de licence adapté au contexte de la virtualisation[11].

Pour s'en convaincre, étudions le scénario suivant :

Considérons un serveur rack doté d'un processeur double-cœur, comportant 2 Go de RAM et un disque dur d'une taille de 80 Go, ce système valant environ CHF 1'500.-. Partons du principe que ce serveur est doté d'une technologie RAID[12] (*Redundant Arrays of Inexpensive Disks*) dont le coût s'élève à plus ou moins CHF 200.- à CHF 300.-, le prix de notre serveur s'élevant dès lors à CHF 1'800.- environ.

Considérons ensuite un serveur lame[13] destiné à la virtualisation, doté de deux processeurs quadruple-cœur, comportant 32 Go de RAM, ainsi que trois disques de 400 Go montés en RAID 5, pour un prix d'environ CHF 14'500.-. Le tableau 3-1 ci-dessous met en exergue tant la consommation électrique du serveur en question que la place utilisée au sein du rack ou les interfaces nécessaires.

[11] Il est par exemple possible, en faisant le choix d'une licence Windows Server™ 2008 R2 ou Windows Server™ 2012, d'installer autant de machines virtuelles que désirées, la licence n'étant nécessaire que pour le hardware qui est, dans ce cas précis, l'hyperviseur (http://www.google.ch/url?sa=t&rct=j&q=2008%20r2%20licence%20%2B%20virtualisation&source=web&cd=1&ved=0CCUQFjAA&url=http%3A%2F%2Fdownload.microsoft.com%2Fdocuments%2FFrance%2FServeur%2F2011%2Fwindows-server-2008-r2%2FWindows_Server_2008_R2_virtualisation_mode_de_licence.pdf&ei=0AVBUKvnM-nP4QT8v4GgCg&usg=AFQjCNGDvhkX8hWT16cxT_G7hdAjgld8zw).

[12] Techniques permettant de répartir des données sur plusieurs disques durs afin d'améliorer soit la tolérance aux pannes, soit la sécurité, soit les performances de l'ensemble, ou une répartition de tout cela.

[13] Qui peut être monté dans un châssis prévu à cet effet (un peu à la manière des serveurs rack) mais qui est plus compact qu'un serveur rack. Il s'agit du serveur permettant d'occasionner le plus faible encombrement possible. Les châssis *ad hoc* fournissent généralement l'alimentation électrique, le refroidissement, l'accès au réseau, la connectique pour l'écran, le clavier et la souris. Plusieurs serveurs lame peuvent ainsi être mutualisés dans le même châssis.

Spécification	Serveur destiné à la virtualisation (physique)	Serveur (physique)	Serveur virtuel
Coût	14'500.- CHF	1'800 CHF	0 CHF
Unité de rack	4U	1U	0
Puissance (watts)	1570	670	0
Interfaces réseau	2*	2	0**

Tableau 3-1 : Comparaison de coûts de machines physiques et virtuelles (1)

* Minimum pour ce type de serveur
** En utilisant des connexions partagées sur la machine hôte

Considérons maintenant les mêmes données mais avec une infrastructure comprenant dix serveurs :

Spécification	Serveur destiné à la virtualisation (physique)	Serveur (physique)	Serveur virtuel
Coût	14'500.- CHF	18'000 CHF	0 CHF
Unité de rack	4U	8U	0
Puissance (watts)	1570	6700	0
Interfaces réseau	2 + 10*	20	10**

Tableau 3-2 : Comparaison de coûts de machines physiques et virtuelles (2)

* Deux pour le serveur hôte et une par serveur virtuel
** Les mêmes dix interfaces physiques que sur le serveur hôte

La réduction du nombre de machines au sein de l'infrastructure, rendue possible par la virtualisation, est donc à l'origine d'économies substantielles, comme nous pouvons le constater dans le tableau 3-2.

3.1.2 Diminution du matériel réseau

La virtualisation permet également de limiter l'achat de composants réseau puisque les machines virtuelles sont capables de communiquer au sein d'un même hôte physique, elles n'ont nul besoin de dispositifs tels que routeur ou commutateur.

De plus, les machines virtuelles peuvent être configurées pour être présentes sur la même carte réseau ou regroupées au sein d'une même interface, ce qui permet là aussi de diminuer le nombre de composants réseau.

3.1.3 Réduction de la consommation électrique

Si la virtualisation permet de diminuer le nombre de machines présentes au sein de l'infrastructure, elle permet également de réduire drastiquement la consommation d'énergie électrique, s'agissant probablement de l'économie la plus frappante.

Il est avéré qu'un petit nombre de serveurs physiques consomment plus d'énergie qu'un unique gros système (i.e. le serveur hôte de machines virtuelles ou VMM), comme illustré au tableau 3-2 plus haut. La virtualisation réduit de facto le nombre d'alimentations, de processeurs et de disques. Or, une quantité importante de chaleur est générée et dissipée par ces éléments, étant précisé que la consommation électrique est étroitement liée au

refroidissement et à la circulation d'air. En diminuant le nombre des éléments précités, nous diminuons d'autant la puissance dédiée au refroidissement les lieux.

La souplesse offerte par la virtualisation du stockage permet d'affiner au mieux ses besoins en capacités de stockage, en agissant ainsi sur la quantité de baies nécessaire. Une réduction du nombre de baies équivaut également à une diminution de la consommation électrique.

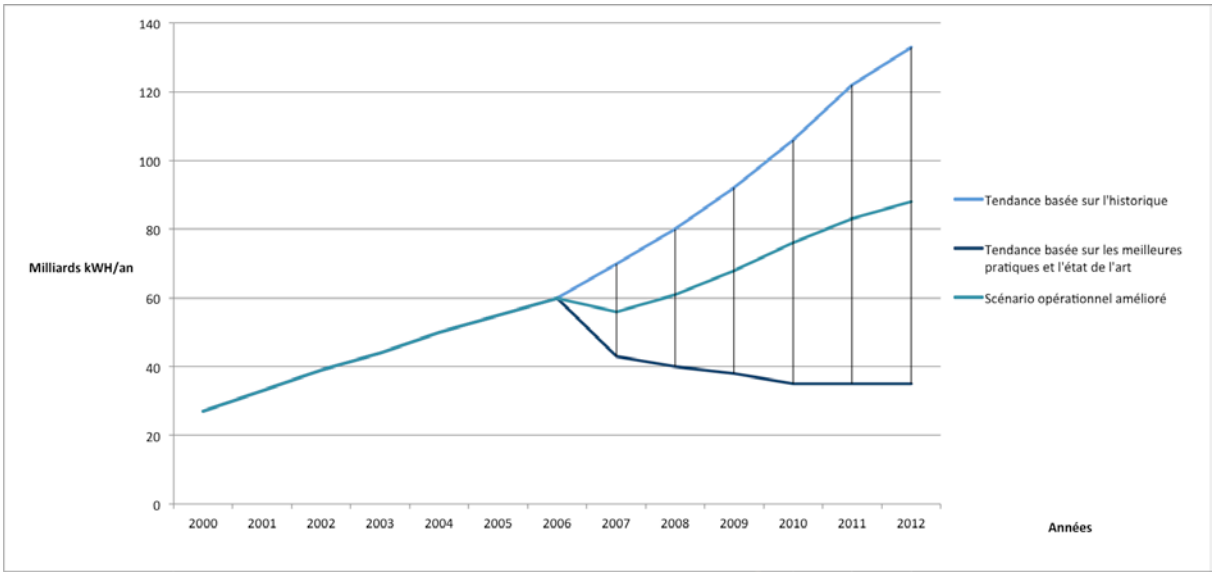


Figure 3-1 : Évolution de la consommation électrique au sein des centres de données aux USA (Source : Étude de Jonathan G. Koomey, Stanford University, août 2011 http://www.analyticspress.com/datacenters.html)

Comme nous pouvons le voir dans la figure 3-1, la courbe correspondant à la croissance de la consommation d'électricité (Scénario opérationnel amélioré) utilisée par les centres de données entre 2000 et 2010 a été infléchie à partir de 2006-2007, étant précisé que si cette consommation a doublé entre 2000 et 2005 (de 30 à 60 milliards de kWh/an), elle n'a progressé « que » de 56% au lieu du double entre 2005 et 2010. Trois facteurs expliquent cette décroissance :

- La crise économique avec pour corollaire la réduction des investissements ;
- Une meilleure gestion de la consommation électrique au sein des centres de données ;
- Une adoption massive de la virtualisation avec pour conséquence une réduction drastique du nombre de serveurs.

Il convient toutefois de tenir compte du fait que lorsqu'une machine virtuelle est en fonction, elle consomme de la mémoire, elle utilise le processeur et le réseau et elle sollicite les disques, augmentant dès lors la consommation électrique totale de l'hôte, cette consommation étant toutefois sans commune mesure avec celle d'une machine physique indépendante.

Ainsi, à l'heure où les considérations écologiques font également parties des priorités, de telles économies d'énergie s'avèrent être un argument supplémentaire lorsqu'il s'agit de la défense du budget octroyé à la direction du système d'information.

3.1.4 Diminution des besoins en climatisation

Si le nombre de machines (serveurs comme baies de stockage) présentes dans le *datacenter* est réduit grâce à la virtualisation, limitant dès lors la consommation électrique, il en ira de même pour les besoins en climatisation, ces derniers étant liés à ladite consommation. De nouvelles économies d'énergie seront *de facto* générées.

3.1.5 Diminution de la consommation d'espace

Les *datacenters* ne sont pas extensibles à l'infini et leur entretien représentent une somme substantielle. Les entreprises doivent faire la part des choses entre les locaux dévolus aux employés et ceux prévus pour l'informatique, les premiers au détriment des seconds.

La virtualisation, en autorisant une réduction du nombre de machines – serveurs comme baies de stockage –, permet par conséquent une diminution conséquente de l'espace qui d'ordinaire aurait été destiné aux serveurs.

Il convient toutefois de tenir compte du fait que la virtualisation augmente la criticité des serveurs, au sens où ces derniers vont être concentrés dans un même lieu, avec pour corollaire le risque de tout perdre lors d'un incident majeur. Cette problématique sera évoquée en détails dans la section 7.1, Sécurisation, relative à la sécurité.

3.1.6 Agrégation des charges d'inactivité

La virtualisation offre la possibilité de consolider les serveurs, avec pour conséquence une réduction du nombre des machines physiques, les charges étant combinées sur du matériel plus récent et par conséquent plus fiable. Le nombre de machines virtuelles nécessaires à la gestion de ces charges peut également être optimisé. En résumé, le matériel est utilisé de manière plus efficace, la consommation d'énergie réduite et les services plus simples à gérer. Les coûts de maintenance peuvent être réduits d'autant puisque le nombre de machines physiques est moindre.

Les opérations de consolidation ont le mérite d'éviter aux administrateurs systèmes de consacrer un temps considérable à la gestion des serveurs, tout en négligeant la veille technologique pourtant indispensable à la pérennisation du système d'information. En effet, créer de nouvelles machines virtuelles basées sur des modèles ne nécessite que quelques clics et limitent drastiquement la manutention. Il n'est nul besoin de commander et de réceptionner du matériel, de le mettre en rack et de procéder par la suite au remplacement de certains composants, de calculer de nouveaux besoins en puissance électrique ou en refroidissement.

3.1.7 Optimisation de la restauration

La virtualisation permet *de facto* de diminuer le temps nécessaire pour la restauration. Par temps de restauration, nous entendons le temps moyen nécessaire pour faire en sorte qu'un système rendu indisponible par une panne quelconque puisse être à nouveau exploité par les utilisateurs. Nous faisons habituellement référence à la notion de RTO dans ce cas (cf. section 7.1, Sécurisation).

La virtualisation réduit considérablement ce temps en faisant bénéficier l'administrateur systèmes d'instantanés (*snapshots*) ou de sauvegardes de machine virtuelle complète pour

procéder efficacement à la restauration. En effet, une restauration par copie directe de système s'avère nettement plus rapide que l'installation d'un nouveau système qui, de surcroit, obligera l'administrateur systèmes à fouiller dans un lot de sauvegardes incrémentales pour que le système restauré soit à jour.

Cette technologie est particulièrement efficace pour peu que les données exploitées par la solution présente sur le serveur ne soient pas stockées sur le même disque logique que l'environnement restauré, sans quoi ces dernières ne seraient plus à jour (en particulier si la sauvegarde de la VM en question date de plusieurs jours). Il convient donc de fournir au serveur virtuel un emplacement logique supplémentaire sur le SAN qui contiendrait les données et qui pourrait être exploité sans problème par la machine restaurée, une fois cette dernière opérationnelle.

Il convient également de préciser qu'une machine virtuelle peut être considérée comme très fiable car elle ne repose sur aucun matériel physique susceptible de tomber en panne. En conséquence, une sauvegarde de machine virtuelle sera toujours un point de restauration stable et fiable pour le matériel physique sous-jacent.

Certains produits, tels que *VMware vCenter Converter*^{TM}[14], permettent la récupération de copies de machines physiques pour une conversion en machines virtuelles (cf. figure 3-2). Cette méthode, généralement qualifiée de sauvegarde P2V (physique vers virtuelle, cf. section 6.1.2.6, Conversion P2V), permet à l'administrateur systèmes de mettre en production une machine virtuelle en remplacement d'un serveur physique défectueux, et ce, rapidement, en toute fiabilité, pour un coût insignifiant et en limitant considérablement les interruptions de services.

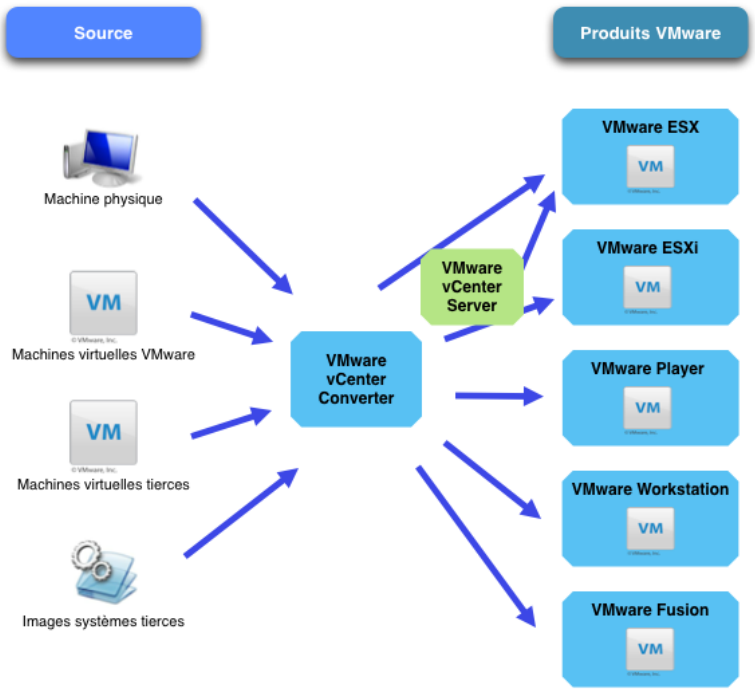


Figure 3-2 : VMware vCenter Converter^{TM} (Source : www.vmware.com)

[14] http://www.vmware.com/fr/products/datacenter-virtualization/converter/overview.html.

3.1.8 Optimisation de la sauvegarde

Les experts considèrent généralement la sauvegarde comme un projet complexe, dépendant fortement des systèmes d'exploitation présents sur les machines, des applications hébergées et de l'outil de sauvegarde sélectionné. Des licences supplémentaires sont généralement vendues aux entreprises par les commerciaux officiant pour le compte des éditeurs. Une fois l'outil de sauvegarde choisi, il devient très difficile d'opter pour une autre solution, l'entreprise étant contrainte d'acquérir de nouvelles licences nécessaires à l'utilisation de certains agents (par exemple pour Exchange™, Microsoft SQL Server™, Oracle®, SharePoint™, etc.).

Les projets de sauvegarde sont donc de forts consommateurs de ressources financières qui sont très difficilement justifiables auprès des décideurs comme un directeur financier notamment.

Étant donné qu'un système d'exploitation est perçu comme un simple fichier par une application de virtualisation, sa sauvegarde consistera également à ne traiter qu'un fichier. Toute licence supplémentaire, toute contrainte système ou toute mise à jour d'agents deviennent dès lors caduques. Une telle opération se résume en quelque sorte à un copier-coller.

Il convient cependant de tenir compte du fait que si, au sein d'une machine virtuelle, les données figurent sur le même disque virtuel que le système d'exploitation, il est impossible de les distinguer. L'usage de produits tels que ***WMware vSphere Data Protection™*** permet toutefois de restaurer une image complète ou uniquement un fichier.

VMware vSphere Data Protection™

VMware vSphere Data Protection™[15] (VDP) est un bon exemple des solutions de sauvegarde et de restauration de machines virtuelles actuellement disponibles sur le marché. C'est à ce titre que nous l'évoquons brièvement à ce stade du document.

VDP n'est autre qu'une *appliance* virtuelle intégrée à VMware vSphere™, l'hyperviseur de la société éponyme. Cette solution a été spécialement développée pour éviter toute perte de données au sein de l'environnement virtuel, en assurant des sauvegardes rapides sur disque et en permettant une restauration complète et tout aussi rapide des données. VDP s'appuie sur la déduplication (cf. section 7.1.4, Déduplication), le traitement parallèle et l'utilisation du mode CBT[16] (*Changed Block Tracking* ou suivi des blocs modifiés).

Le recours à un algorithme de déduplication à bloc de taille variable (cf. section 7.1.4.2, Bloc variable) limite considérablement l'espace disque nécessaire, limitant ainsi la croissance continue du volume consacré aux sauvegardes. La déduplication des données est effectuées sur l'ensemble des machines virtuelles associées à l'*appliance* virtuelle VDP.

[15] http://www.vmware.com/solutions/datacenter/business-continuity/data-protection.html.
[16] Méthode mise au point par VMware® permettant de faciliter les sauvegardes incrémentales.

En outre, VDP réduit la charge des hôtes et la bande passante nécessaire sur le réseau. En effet, en s'intégrant étroitement à vStorage™ APIs for Data Protection™[17] (VADP), VDP exploite pleinement le mode CBT, en limitant les transmissions sur le réseau aux modifications effectuée durant la journée en cours uniquement. Le processus est totalement transparent, entièrement automatisé et peut assurer la sauvegarde de huit machines virtuelles simultanément. VDP étant hébergé dans une *appliance* virtuelle, les processus de sauvegarde ne sont plus situés au sein des VM en production.

Les stratégies de sauvegarde peuvent être définies de manière centralisée, depuis le client vSphere™, soit l'interface d'administration de l'hyperviseur, sans avoir à utiliser une autre console. En effet, VDP est étroitement intégré à vCenter Server™[18].

3.1.9 Amélioration de la sécurité

Au sein d'une infrastructure virtuelle, la sécurité peut être efficacement renforcée, pour peu qu'un certain nombre de bonnes pratiques soient respectées. Dans le cas contraire, la sécurité peut rapidement s'en trouver compromise. Eu égard à l'importante de l'enjeu y relatif, nous y avons consacré une section à part entière, s'agissant de la section 7.1, Sécurisation.

3.2 Simplification du déploiement

Lorsque de nouveaux serveurs ou de nouvelles applications doivent être déployés au sein des moyennes ou grandes structures, les différents acteurs concernés contribuent bien souvent (sans forcément le vouloir) à ralentir le processus d'intégration.

En effet les prérogatives des uns ne correspondent que très rarement à celles des autres. L'utilisateur est soucieux de pouvoir utiliser le logiciel dont il a besoin au plus vite. Le chef de projet se voit forcé de passer par les experts systèmes et réseaux pour l'intégration. Ces derniers, débordés, fournissent les spécifications nécessaires concernant l'achat du serveur prévu pour l'hébergement de ladite application plusieurs semaines après la demande du chef de projet. Le gestionnaire du centre de données souhaite connaitre les caractéristiques du nouveau serveur pour l'intégrer au sein de ses locaux. Il prendra également plusieurs semaines pour donner son accord, le temps de procéder à l'examen de l'espace disponible et aux calculs relatifs à l'alimentation énergétique et à la climatisation. Le directeur financier, quant à lui, décide de faire patienter le fournisseur durant quelques semaines, le temps que ce dernier révise ses prix à la baisse.

Même si ce scénario est quelque peu exagéré, il résume parfaitement le processus d'intégration précédemment évoqué.

La virtualisation permet de faciliter grandement ce processus. En premier lieu, il n'est nullement nécessaire d'acquérir du matériel supplémentaire puisqu'il suffit de créer une

[17] API de stockage permettant aux clients et aux éditeurs de logiciels indépendants d'optimiser et d'étendre les fonctions de l'hyperviseur vSphere™ dans les domaines de la détection de stockage, de l'intégration des baies, du *Multipathing* et de la protection des données.
[18] VMware vCenter™ Server, soit une plate-forme utilisée pour la gestion de la virtualisation (anciennement appelée VMware VirtualCenter™).

nouvelle machine virtuelle. Il n'est nul besoin de mettre en rack, de câbler, de brancher, ni de se préoccuper des capacités de refroidissement pour le matériel supplémentaire.

Ensuite, ce procédé peut être mis en œuvre rapidement et de manière automatisée. Dans le cas où plus d'une machine virtuelle était nécessaire, il suffirait d'en créer une, puis de la cloner autant de fois que désiré.

De plus, l'espace disque dévolu à tout nouveau serveur peut être facilement mis à disposition à partie du SAN, par le biais de la virtualisation du stockage. Un disque virtuel (donc logique) peut être créé en quelques clics de souris à partir d'un LUN quelconque du pool de stockage.

Une intégration peut dès lors se résumer à quelques heures en lieu et place des quelques semaines, voire des quelques mois qu'elle aurait exigé auparavant.

3.3 Simplification de l'administration

D'une manière générale, les produits disponibles sur le marché de la virtualisation disposent tous d'une console de gestion des machines virtuelles accessible à partir d'une interface unique. Cette interface de gestion centralisée prend tout son sens lorsqu'il s'agit de prendre en charge une infrastructure composée de systèmes d'exploitation hétérogènes.

L'interface en question permet d'interagir avec la console de ce dernier comme si nous nous trouvions devant le système physique.

De plus, un certain nombre d'outils de monitoring y sont disponibles, permettant aux administrateurs systèmes de suivre en temps réel la consommation ou les performances des différentes machines virtuelles ou la capacité d'espace de stockage disponible dans le pool y relatif, parmi bien d'autres fonctionnalités. La figure 3-3 montre un échantillon des outils disponibles par le biais du client VMware vSphere™ pour effectuer le *monitoring* des serveurs (ici, un certain nombre de diagrammes de performances).

Figure 3-3 : Console d'administration de la plateforme VMware vSphere™

Les opérations de maintenance programmées, les phases de migration ou de mise à jour des logiciels, qui sont généralement considérées comme des opérations délicates au sein d'un environnement physique, sont grandement simplifiées par les fonctionnalités inhérentes à la virtualisation, avec pour corollaire une augmentation de l'efficacité opérationnelle.

3.4 Optimisation de la gestion de l'obsolescence matérielle

La constante évolution technique du matériel et des applications que nous connaissons dans le domaine des systèmes d'information, nous oblige, en tant qu'administrateur systèmes, à veiller constamment à mettre à jour notre infrastructure.

Nous estimons l'espérance de vie d'une infrastructure matérielle à trois, voire cinq ans au maximum. La garantie offerte par le constructeur sur le produit en question est généralement d'une telle durée, ce qui ne trompe pas sur les délais précités.

Une application, quant à elle, peut avoir été développée pour fonctionner sur une plateforme plus récente que celle disponible sur le serveur sensé l'héberger. *A contrario*, ce dernier doit éventuellement être doté d'un système d'exploitation correspondant aux standards en vigueur plusieurs années auparavant, pour que l'application puisse y fonctionner.

La migration d'une telle application sur un environnement virtuel adéquat permet de résoudre de telles problématiques. Bien que le système d'exploitation en question ne soit plus supporté par son éditeur, l'application peut continuer à y fonctionner. Les problèmes de mises à jour de l'environnement ou de support de ce dernier par l'éditeur ne sont pas résolus mais l'entreprise peut ainsi donner un sursis à l'application concernée et éviter d'avoir à se précipiter dans de nouveaux développements d'applications ou dans l'achat de ces dernières, lui épargnant ainsi une fortune.

De plus, la virtualisation des applications elles-mêmes permet de faire fonctionner ces dernières dans des « bulles » applicatives qui ne dépendent plus du système d'exploitation sous-jacent, offrant les mêmes bénéfices que décrits au paragraphe précédent, et ce, malgré le fait que le système d'exploitation ait été mis à jour (cf. section 5.3, Applications).

La virtualisation des serveurs permet de gérer au mieux la problématique de l'espérance de vie du matériel puisque si le système hôte devient obsolète, ce ne sera pas le cas des machines virtuelles. Nous pourrons ajouter de la RAM, des processeurs, de l'espace disque, des interfaces réseau ou tout autre périphérique à ces dernières indépendamment du matériel physique sous-jacent. Leur système d'exploitation pourra même être mis à jour une fois que le matériel physique qui les héberge l'aura également été.

De plus, remplacer un serveur obsolescent par du nouveau matériel peut se faire de manière tout à fait transparente pour les clients dudit serveur. Les machines virtuelles clientes sont simplement migrées sur un nouveau système physique, et ce, en court de production, tel qu'illustré à la figure 3-4.

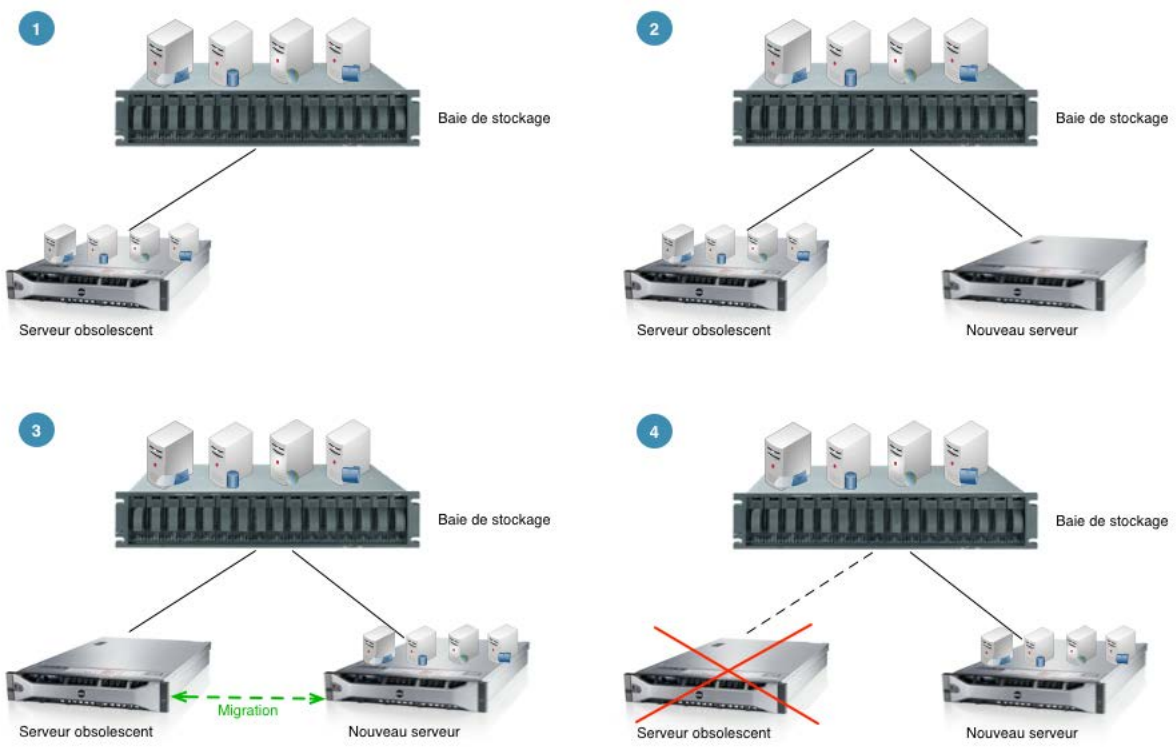


Figure 3-4 : Migration d'un serveur obsolescent vers un nouveau serveur (Source : Virtualisation des systèmes d'information avec VMware® de P. Gillet)

3.5 Amélioration de la gestion du changement

La virtualisation permet une application plus souple de la gestion du changement. Nous rappelons que la gestion du changement est un des six processus de la partie **Soutien des services** des bonnes pratiques ITIL *(Information Technology Infrastructure Library)*.

Un changement consiste à modifier ou à supprimer un des composants de l'infrastructure du système d'information (logiciel, application, équipement, matériel, configuration, documentation, procédure, etc.). Il peut également s'agir de la création d'un nouveau composant.

Le cycle de vie d'une machine virtuelle est beaucoup plus facilement contrôlable. Ce type de « matériel » peut également être éliminé de manière totalement transparente, ce qui contribue du même coup à optimiser la gestion du cycle de vie du matériel.

3.6 Optimisation de l'évolution des capacités physiques

Les capacités d'un système physique sont limitées, étant donné qu'elles ne peuvent être modifiées. Un système monoprocesseur restera toujours un système monoprocesseur. Si la limite de prise en charge de la mémoire vive par notre système se situait à 4 Go, nous ne pourrions en ajouter plus. Par opposition, les machines virtuelles ne sont pas affectées par de telles limites. Pour autant que le système hôte dispose de capacités matérielles suffisantes et que le logiciel de virtualisation le permette, de nouvelles ressources peuvent être allouées aux machines virtuelles sans autre forme de procès.

3.7 Amélioration de l'équilibre des charges (*load balancing*)

Au sein d'une configuration visant à l'équilibrage des charges (par exemple au sein d'un *cluster* de serveurs Web), les machines virtuelles contribuent à la mise en place d'une méthode peu onéreuse mais cependant efficace de répartition du trafic réseau sur plusieurs systèmes. En effet, le trafic réseau peut être facilement réparti entre plusieurs systèmes, tant virtuels que physiques, et ce, grâce à un répartiteur de charge réseau[19].

Supposons dès lors que nous désirions virtualiser nos services Web afin de supprimer notre dépendance aux systèmes physiques. Considérons que notre trafic Web (port 80) est redirigé vers une unique adresse en **.ch**, servie par trois serveurs physiques (cf. figure 3-5).

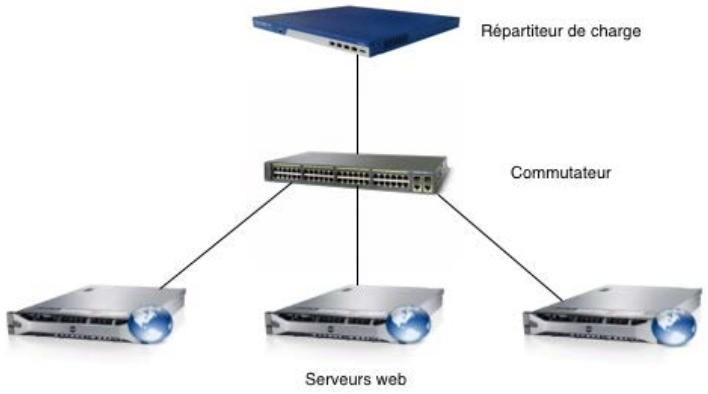


Figure 3-5 : Charge des serveurs, scénario 1 (Source : Virtualisation en pratique de Kenneth Hess et Amy Newman)

La figure 3-6 illustre une configuration similaire mais comportant des machines virtuelles à la place des machines physiques. Dans ce cas, le nombre de machines physiques n'a pas changé, eu égard au fait que les charges ainsi équilibrées doivent être isolées à un certain degré. Les trois machines virtuelles pourraient cependant cohabiter sur le même serveur physique car chacune d'entre elles possède sa propre adresse IP. Il est également possible d'allouer à chaque machine virtuelle sa propre interface réseau associée à son interface virtuelle.

L'inconvénient d'un tel scénario réside dans le fait qu'il contrevient à sa raison d'être initiale, à savoir l'équilibrage des charges. En effet, un trafic Web important dégraderait les performances de cet unique hôte.

Afin d'atténuer l'altération de performances liée aux entrées/sorties sur un disque unique de l'hôte partagé, nous allons utiliser du stockage réseau auquel toutes les machines virtuelles sont susceptibles de pouvoir se connecter (cf. section 5.4, Stockage) afin d'en obtenir le contenu désiré.

[19] Cette répartition de charge réseau peut se faire par un logiciel éventuellement contenu au sein d'une image virtuelle prévue pour les solutions de virtualisation des principaux éditeurs du marché (par exemple ALOHA Load Balancer Virtual Appliance), elle-même disponible à partir d'une *appliance* d'1U pouvant être mise en rack.

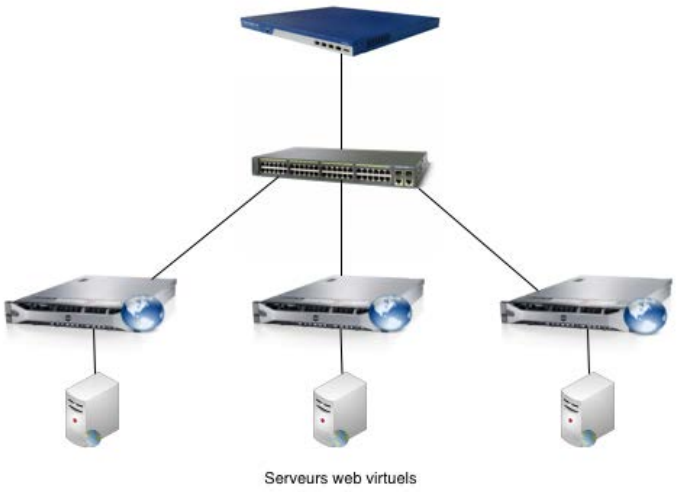


Figure 3-6 : Charge des serveurs, scénario 2 (Source : Virtualisation en pratique de Kenneth Hess et Amy Newman)

La figure 3-7 illustre le cas d'un hôte unique hébergeant trois serveurs Web virtuels. Si les trois machines virtuelles sont équilibrées en charge, tant les capacités de traitement que la sécurité de l'infrastructure sont, quant à elles, nullement garanties.

Figure 3-7 : Charge des serveurs, scénario 3 (Source : Virtualisation en pratique de Kenneth Hess et Amy Newman)

La figure 3-8 illustre par contre le même service Web correctement équilibré, faisant appel au stockage réseau mentionné plus haut.

Figure 3-8 : Charge des serveurs, scénario optimal (Source : Virtualisation en pratique de Kenneth Hess et Amy Newman)

Notons que dans un centre de données réel, chacun des hôtes présents sur la figure 3-8 hébergerait plus d'une machine virtuelle.

3.8 Simplification des tests de logiciels

Utiliser une machine virtuelle pour effectuer des tests de logiciels fut l'une des premières raison pour laquelle la virtualisation des environnements x86 a été mise en œuvre.

Une fois la copie de travail créée (création de la machine virtuelle, démarrage, application de correctifs, attribution d'un nom et d'une adresse IP), une sauvegarde de cette dernière est réalisée (*snapshot*). Cette copie de travail est par la suite utilisée pour l'installation, la modification ou la suppression de paquetages logiciels. Cette méthode permet ainsi d'éviter tout conflit ou problème potentiel (plantage, écrans bleus, vidages de mémoire, etc.) qui pourraient survenir au moment de la mise en production desdits logiciels.

Si certains de ces problèmes venaient à apparaitre, la copie de travail pourrait servir de plateforme de débogage. Si cette dernière venait à être irrémédiablement corrompue, il suffirait de la supprimer et d'en créer une nouvelle à partir de la machine originale. Cette manière de faire évite d'avoir à réinstaller à chaque fois le système d'exploitation, les applications et les correctifs, avec pour corollaire un gain substantiel de temps.

Lorsque le système est parfaitement fonctionnel, la copie de travail de la machine virtuelle peut être mise en production en remplacement de l'ancienne mouture, en la copiant dans le *cluster* des machines de production présent sur le VMM. Le processus de test, le débogage et le déploiement sont donc clairement rationnalisés.

4 Fondamentaux technologiques de la virtualisation

Nous l'avons vu au chapitre 1, la virtualisation consiste en une abstraction physique des ressources informatiques. Les serveurs et, par extension, les systèmes d'exploitation, ne sont dès lors pas les seuls éléments du système d'information à pouvoir être virtuels.

Ces derniers constituent toutefois la pierre angulaire d'une infrastructure virtuelle, les serveurs étant généralement les premiers dispositifs à être virtualisés lorsqu'une telle décision stratégique est prise. C'est une des raisons pour lesquelles nous avons décidé d'aborder l'aspect technologique de la virtualisation en nous intéressant tout particulièrement à ce domaine d'application.

Nous précisons également que, si décision était prise de virtualiser les stations de travail, un hyperviseur serait nécessaire, tout comme dans le cas de la virtualisation des serveurs. Les éléments techniques relatifs à ces deux domaines d'application s'avèrent donc similaires. Ceci constitue une raison supplémentaire d'aborder les aspects techniques relatifs à la virtualisation sous l'angle de la virtualisation des serveurs.

Ainsi, même si évoquer la notion d'hyperviseur, nous oblige à sous-entendre également l'existence d'un ou plusieurs serveur(s) physique(s) dédié(s) à cette tâche, ainsi que la présence d'un réseau de stockage, nous n'aborderons pas les aspects techniques liés à la virtualisation du stockage et du réseau dans le présent chapitre, et ce, dans le but de rester le plus synthétique possible. Il en ira d'ailleurs de même pour la virtualisation des applications.

Ce chapitre est en effet dédié aux avancées techniques significatives ayant historiquement permis l'avènement de la virtualisation. Or, il se trouve que les premières recherches menées dans ce domaine ont eu pour objectif de virtualiser des serveurs d'applications. Aborder les fondements techniques de la virtualisation du stockage, des réseaux ou des applications dans ce chapitre ou, *a fortiori*, dans ce document, aurait pour conséquence d'alourdir passablement ce dernier, la méthode d'abstraction de ce type de ressources informatiques étant passablement différente de celle utilisée pour la virtualisation des serveurs ou des stations de travail.

Aussi, nous nous contenterons d'évoquer ces sujets sous un angle plutôt pratique au sein du chapitre 5, dévolu aux domaines d'applications et sous un aspect plutôt stratégique au sein du chapitre 6, consacré à la construction de l'infrastructure virtuelle.

Avant d'entrer dans le vif du sujet, soit la virtualisation, nous reviendrons, à la section 4.1, Notions élémentaires, sur un certain nombre de notions relatives aux éléments matériels composant un ordinateur, ainsi qu'aux mécanismes ou principes qui les régissent. Nous souhaiterions en effet nous assurer que le lecteur maitrise les bases nécessaires à la compréhension des notions abordées plus avant dans ce chapitre. En effet, la virtualisation peut avoir un impact sur ces éléments ou dépendre de ces derniers.

À la section 4.2, La genèse de la virtualisation, nous aborderons la genèse de la virtualisation en évoquant les principes élaborés pour l'encadrer et la première méthode qui fut mise au point.

Nous procéderons à l'examen des différentes techniques de virtualisation à la section 4.3, Techniques de virtualisation. Cependant, même si certaines méthodes s'appliquent à des architectures de processeur diverses, nous insisterons sur la virtualisation des environnements x86. En effet, l'adoption généralisée de Windows® et l'émergence de Linux™ comme systèmes d'exploitation serveurs dans les années 1990 ont fait de cette architecture la norme de l'industrie. Il est dès lors impensable de ne pas lui réserver une place de choix dans le présent chapitre.

Nous terminerons, à la section 4.4, Mise en œuvre, par un aperçu des mises en œuvre possibles des techniques évoquées précédemment ou, en d'autres termes, des applications commerciales y relatives.

4.1 Notions élémentaires

4.1.1 Composants et principes de fonctionnement d'un ordinateur

4.1.1.1 Processeur

Le processeur, ou l'unité centrale de traitement, est un microprocesseur dont le rôle fondamental consiste à exécuter une série d'instructions stockées appelées **programmes**. Les instructions et les données transmises au processeur sont exprimées en langage machine (binaire) et sont généralement stockées dans la mémoire. Le séquenceur (ou unité de contrôle de commande), dont le rôle consiste à commander le chemin de données et à réguler les interactions de ce dernier avec la mémoire vive, ordonne la lecture du contenu de cette dernière et se charge de la constitution des mots présentés par la suite à l'unité arithmétique et logique, soit l'organe chargé d'effectuer les calculs.

L'ensemble des instructions et des données forme un programme.

Ces instructions sont définies dans le jeu d'instructions que possède chaque type de processeur et représentent les opérations que ce dernier est capable d'exécuter. Le processeur possède également ses propres registres, soit des emplacements de mémoire au meilleur temps d'accès qui sont en quelque sorte disponibles pour son jeu d'instructions.

Le processeur gère également les interruptions, soit des arrêts temporaires de l'exécution d'un programme, et définit les privilèges (cf. section 4.1.1.6, Mode d'adressage protégé (*protected mode*) et section 4.1.1.7, Niveaux de privilèges).

4.1.1.2 Principe de la mémoire virtuelle

Ce mécanisme, reposant sur l'utilisation d'une mémoire de masse (i.e. disque dur), permet en premier lieu de partager la mémoire en processus ou, en d'autres termes, augmenter le taux de multiprogrammation. La mémoire virtuelle permet également la mise en place des mécanismes de protection de la mémoire, puisqu'un programme ne peut plus accéder à la mémoire au travers des adresses physiques (comme en mode réel) mais uniquement au moyen d'adresses virtuelles.

Avant de considérer le modèle théorique général de la mémoire virtuelle, ainsi que les manières de le mettre en pratique, examinons les différents problèmes qu'il doit résoudre.

4.1.1.2.1 Différents problèmes à résoudre

Relocation

Un programme contient généralement un certain nombre de références à des variables, des fonctions ou des zones du code. Ces références sont parfois relatives mais souvent absolues. Dans un contexte de multiprogrammation, les références absolues ne sont plus valables lorsque le programme y relatif est chargé à un endroit différent de la mémoire que lesdites références.

La relocation dynamique au démarrage du programme est une des solutions disponibles pour pallier ce problème. Chaque adresse se voit réécrite suivant l'endroit précis où le programme a été chargé. Il est possible d'envisager également la relocation dynamique lors de chaque accès, moyennent un coût non négligeable.

SWAP

Un certain nombre de zones mémoires doivent pouvoir être créées sur le disque dur, une fois que la mémoire centrale est pleine.

Protection de la mémoire

Il est nécessaire de mettre en œuvre un mécanisme qui empêche un processus d'accéder (en lecture comme en écriture) à l'emplacement de la mémoire dévolu à un autre processus, voire au système d'exploitation lui-même.

Mémoire partagée

Il est nécessaire, afin que la communication soit suffisamment rapide entre deux processus, de prévoir un mécanisme qui leur permette de partager de la mémoire, sans toutefois compromettre la protection de la mémoire évoquée ci-dessus.

Fichiers « mmapés »

Il est utile de pouvoir fournir aux programmes la capacité de projeter un fichier en mémoire, en l'utilisant directement, comme si ce dernier faisait office de données en mémoire. Ce mécanisme, appelé **mmap** (pour *Memory Map* en anglais), offre au programme la flexibilité et la rapidité de données en mémoire, tout en bénéficiant de la persistance.

4.1.1.2.2 Hiérarchie des mémoires

Il se trouve que la mémoire est soit rapide et très chère, soit bon marché mais particulièrement lente. De plus, la mémoire rapide n'est jamais persistante.

Afin qu'un ordinateur soit à même de nous offrir des performances satisfaisantes, nous sommes dans l'obligation d'utiliser différentes mémoires, certaines étant très rapides mais limitées en quantité (par exemple, les registres), d'autres étant lentes mais disponibles en grande quantité (en particulier les disques durs). Les différents types de mémoire en question, ainsi que leur taille courante et la vitesse d'accès qu'elles offrent habituellement sont mentionnés dans le tableau 4-1, ci-dessous.

Type de mémoire	Taille typique	Vitesse typique
1. Registres	256 octets	0.5 ns
2. Cache L1	16 Ko	1 ns
3. Cache L2	2 Mo	2 ns
4. Mémoire centrale	2 Go	10 ns
5. Disque dur	300 Go	10 ms

Tableau 4-1 : Hiérarchie des mémoires (Source : Systèmes d'exploitation – Gestion de la mémoire de Pilot Systems 2007 par Gaël Le Mignot)

Cette hiérarchie de mémoires peut être considérée comme une hiérarchie de caches, les registres servant de cache au cache L1, le cache L1 faisant de même pour le cache L2, qui lui-même sert de cache à la mémoire centrale, cette dernière n'étant qu'un cache pour le disque dur.

Si le système d'exploitation n'a que peu de contrôle sur les caches gérés automatiquement par le matériel (L1 et L2) ou sur les registres (qui sont gérés par le compilateur), il intervient sur la gestion de la mémoire centrale. C'est en effet à lui qu'incombe de décider ce qui doit être gardé en mémoire et ce qui doit l'être uniquement sur le disque dur.

4.1.1.2.3 Backing Store

Pour tout système de gestion de la mémoire un tant soit peu évolué, les notions de *swap* et de cache disque sont fusionnées. La mémoire est utilisée comme cache pour le disque. Dans un premier cas, nous pouvons libérer de la mémoire en écrivant sur le disque. Dans un second cas, nous pouvons utiliser la mémoire disponible pour éviter les accès disque.

Chaque zone mémoire, en fonction de son utilisation (bibliothèque partagée, exécutable, données, cache disque), est associée à un *backing store* qui se trouve être la zone du disque qui permet de stocker son contenu en cas de nécessité.

4.1.1.2.4 MMU

L'unité de gestion mémoire (MMU pour *Memory Management Unit* en anglais) est un composant responsable de l'accès à la mémoire lorsque le processeur l'exige. Effectuer des traitements logiciels sur chaque accès mémoire serait en effet bien trop coûteux en temps, et, de surcroit, difficile à réaliser (puisque le logiciel a lui-même besoin d'accéder à la mémoire).

Ce dispositif fait intégralement partie de l'unité centrale et accède, de ce fait, à ses registres.

Les attributions de ce dispositif sont :

- La traduction d'adresses logiques en adresses linéaires, par le biais de son unité de segmentation (cf. section 4.1.1.3, Segmentation) ;
- La traduction d'adresses linéaires en adresses physiques par le biais de son unité de pagination (cf. section 4.1.1.4, Pagination) ;
- Le contrôle de tampon ;
- L'arbitrage du bus ;
- La protection de la mémoire, assurée par le MPU, soit le *Memory Protection Unit*.

Cette unité effectue donc une traduction entre l'adresse virtuelle demandée par le processeur et l'adresse physique, au sens où l'entend le matériel.

Nous explicitons les notions de segmentation et de pagination dans les sous-chapitres y relatifs (voir plus bas), ces dernières étant liées à la notion de mémoire virtuelle, le MMU faisant partie intégrante de ce processus.

Le principe de protection de la mémoire consiste, quant à lui, à empêcher un programme donné d'accéder à l'espace mémoire utilisé par un autre programme, voire par le système d'exploitation lui-même. Il s'agit donc d'allouer à chaque programme une zone mémoire protégée, à laquelle aucun autre programme n'a accès (en lecture comme en écriture). Il va de soi que cette caractéristique s'avère cruciale pour la stabilité du système.

Dans le cas où un programme tenterait d'accéder à une zone mémoire située hors de la plage qui lui avait été réservée, une interruption est levée par le MMU. Cette dernière est interceptée par le processeur qui fait dès lors parvenir une instruction au processus concerné, ayant pour effet de le stopper.

4.1.1.3 Segmentation

La segmentation est une technique gérée par l'unité de segmentation du MMU, utilisée par les systèmes d'exploitation pour diviser la mémoire physique (segmentation pure) ou de la mémoire virtuelle (segmentation avec pagination) en segments.

4.1.1.3.1 Segmentation simple

La segmentation simple, comme celle utilisée dans le mode réel (cf. section 4.1.1.5, Mode d'adressage réel (*real mode*)) des processeurs x86, consiste à spécifier, lors de chaque accès à la mémoire, un registre de segment. Ce dernier contient une valeur qui détermine à partir d'où l'adresse mémoire est spécifiée.

À titre d'exemple, si un segment est indiqué comme commençant à l'adresse physique 0x1000 et que nous demandons l'adresse 0x42, nous utiliserons en réalité l'adresse 0x1042. L'adresse spécifiée en complément du segment est généralement appelée *offset*.

Le x86 possède un registre spécifique à la pile et un autre spécifique au code (là où sont chargées les instructions), ce qui lui permet de déterminer parfois automatiquement le registre de segment à utiliser.

4.1.1.3.2 Segmentation avancée

Dans le cadre de la segmentation avancée, le registre de segment ne contient pas directement l'adresse physique de base du segment mais indique une entrée dans une table appelée **table de descripteurs de segments**.

Cette dernière contient, pour chacun des segments, l'adresse de début et de fin du segment, ainsi que des informations de protection (i.e. lecture seule, utilisable uniquement par le noyau, etc.).

Lors d'un accès mémoire, les opérations suivantes sont réalisées :

1. Le descripteur de segment est chargé depuis la table ;
2. Les modes de protection sont vérifiés ;
3. L'adresse du début (appelée **base**) du segment est ajoutée à l'*offset* ;
4. Une comparaison est effectuée pour vérifier que cette somme ne dépasse pas le sommet du segment.

À partir de l'architecture i386 et supérieure, plusieurs tables de descripteurs de segment sont disponibles. Il existe une GDT (*Global Descriptor Table*) qui contient les segments communs à tous les processus, et une LDT (*Local Descriptor Table*) qui contient ceux qui sont spécifiques à chaque processus.

4.1.1.3.3 Utilisation de la segmentation

La vocation première de la segmentation est de régler le problème de la relocation. Il suffit en effet de positionner les registres de segments aux bonnes valeurs et le programme peut être chargé à n'importe quel endroit au sein de la mémoire.

La segmentation peut être utilisée pour implémenter du *swap*. Le contenu complet d'un segment peut être parfaitement transféré sur le disque dur, ce segment étant ensuite marqué comme non valide. Dès lors qu'un accès sera initié sur ce segment, le MMU (cf. section 4.1.1.2.4, MMU) signalera l'erreur au système d'exploitation, lequel devra recharger le segment en question.

Dans le cas où seul le noyau est habilité à spécifier les descripteurs de segments, la segmentation permet d'instaurer une première forme de protection de la mémoire. En effet, toute zone mémoire qui ne se trouve par dans un segment configuré pour l'application ne pourra pas être utilisé.

Les descripteurs de segments contiennent en général des informations spécifiques aux modes de protection, un segment pouvant être déclaré en lecture seule, par exemple. La protection entre espace noyau et espace utilisateur peut être configurée au niveau des segments : un segment marqué **noyau** (ou superviseur) ne pourra être utilisé que par le noyau, alors qu'un segment marqué **utilisateur** le sera par les deux.

La segmentation peut enfin être utilisée pour implémenter de la mémoire partagée en créant un segment spécifique, disponible pour plusieurs processus.

4.1.1.3.4 Avantages et limites de la segmentation

La segmentation présente l'énorme avantage d'être très simple et efficace à implémenter au niveau matériel, ce dernier sachant gérer très rapidement addition et comparaison.

Plusieurs désavantages caractérisent toutefois la segmentation, dont trois d'entre eux en particulier.

La segmentation fournit malheureusement une faible granularité. Il est nécessaire, par exemple, pour le *swap* de déplacer un segment entier vers le disque, ce qui est susceptible de coûter très cher.

Étant donné qu'un segment doit être continu en mémoire physique, le risque de fragmentation constitue dès lors un problème. Partons du principe qu'au sein d'une mémoire d'une taille de 256 Ko, il existe un espace disponible de 96 Ko entre 32 Ko et 128 Ko et un autre espace disponible de 64 Ko entre 192 Ko et 256 Ko. Il ne sera pas possible de loger un segment de 128 Ko en mémoire, alors que cette dernière possède pourtant bien un espace résiduel de 160 Ko, les deux zones précédemment évoquées étant respectivement trop petites pour accueillir ces données.

Enfin, la segmentation n'est pas parfaitement transparente pour le programme qui l'utilise, en particulier lorsqu'on utilise de la mémoire partagée. Dans ce cas de figure, le programme doit nécessairement être « conscient » du fait qu'il utilise un segment. Or, les langages d'un niveau plus élevé que l'assembleur (comme par exemple le C) ne connaissent pas le concept de segment.

4.1.1.4 Pagination

Le principe de la pagination, ou plus exactement de mémoire paginée (*Expanded Memory* en anglais), consistait à faire utiliser par les programmes une partie de la mémoire initialement réservée aux périphériques, à l'époque des architectures d'adressage réel (cf. section 4.1.1.5, Mode d'adressage réel (*real mode*)), au sein desquelles les programmes n'étaient autorisés qu'à 1 Mo d'adressage dont uniquement 640 Kio[20] étaient disponibles pour la mémoire vive normale (appelée également mémoire conventionnelle).

Nous aborderons plus précisément dans ce sous-chapitre, le principe de mémoire virtuelle paginée qui nous intéresse plus directement dans le contexte du présent document.

La pagination a pour but de régler les principaux problèmes de la segmentation (cf. section 4.1.1.3, Segmentation).

Les adresses mémoires émises par le processeur s'avèrent être des adresses virtuelles, ces dernières indiquant la position d'un mot dans la mémoire virtuelle (cf. section 4.1.1.2, Principe de la mémoire virtuelle).

Cette mémoire virtuelle est formée de zones de taille similaire, appelées pages. Une adresse virtuelle peut donc être considérée comme un couple {numéro de page, déplacement dans la page}.

La mémoire vive se compose également de zones de même taille, appelées cadres (*frames* en anglais), dans lesquelles les pages peuvent prendre place (taille d'un cadre = taille d'une page).

Un mécanisme de traduction[21] assure la conversion des adresses virtuelles en adresses physiques, en se basant sur une table des pages (*Pages Table* en anglais) qui permet d'obtenir le numéro du cadre qui contient la page recherchée. L'adresse physique obtenue est le couple formé par le numéro de cadre et le déplacement.

[20] Le symbole Kio correspond à kibioctet qui est un préfixe binaire. Sa valeur est de 2^{10}, tout comme un ko (kilo-octet) en usage traditionnel.
[21] *Translation* en anglais ou génération d'adresse.

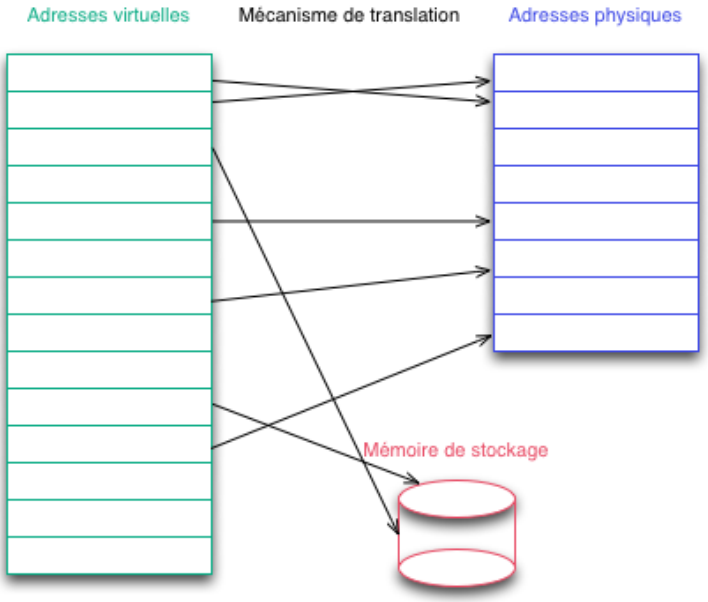


Figure 4-1 : Pagination (Source : fr.wikipedia.org/wiki/Mémoire_virtuelle)

Le nombre de pages peut être supérieur au nombre de cadres et c'est là tout l'intérêt du mécanisme en question. Comme illustré à la figure 4-1, les pages qui ne sont pas stockées en mémoire le sont sur un autre support, généralement un disque dur. Elles ne seront ramenées dans un cadre que lorsque le système en aura besoin.

La table des pages est indexée par le numéro de page. Les lignes qui y sont contenues sont appelées **entrées dans la table des pages** (*Pages Table Entry*, abrégé PTE en anglais) et contient le numéro de cadre. Cette table étant susceptible d'être située à n'importe quel emplacement de la mémoire, un registre spécial est destiné à conserver son adresse, s'agissant de la *Page Table Register* ou PTR.

Le mécanisme de traduction précité fait partie du MMU, lequel contient également une partie de la table des pages stockée dans une mémoire associative formée de registres rapides. La table des pages située en mémoire n'a donc pas à être consultée lors de chaque accès mémoire.

La figure 4-2 illustre l'exemple d'une machine dont le processeur génère des adresses virtuelles sur 32 bits qui lui permettent ainsi d'accéder à 4 Gio de mémoire. Les pages ont une taille de 4 Kio. Nous pouvons déduire que le champ de déplacement occupe les 12 bits de poids faible et le champ numéro de page les 20 bits de poids fort. Nous pouvons également noter la présence d'un champ spécial appartenant à chaque PTE, sa largeur étant réduite à un bit pour des raisons de simplification. Ce bit se trouve être un bit de validité. Si ce dernier est à 0, le numéro de cadre peut être considéré comme non valide.

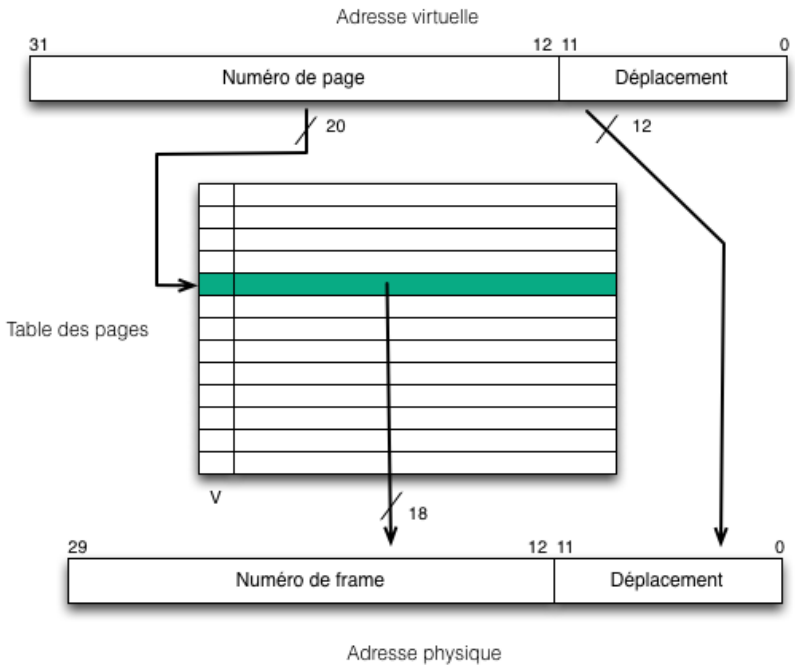


Figure 4-2 : Table des pages (Source : fr.wikipedia.org/wiki/Mémoire_virtuelle)

Il faut donc disposer d'une technique permettant de mettre à jour cette PTE afin de la rendre valide doit être disponible.

Trois cas peuvent se produire :

1. L'entrée est valide et se substitue donc au numéro de page pour former l'adresse physique ;
2. L'entrée dans la table des pages est non valide. Il faut donc lui trouver un cadre libre en mémoire vive et mettre le numéro correspondant dans l'entrée concernée de la table des pages ;
3. L'entrée dans la table de pages est valide mais correspond à une adresse de la mémoire de masse qui contient le contenu du cadre. Un mécanisme devra dès lors ramener ces données pour les placer en mémoire vive.

Dans les deux derniers cas, une interruption, communément appelée défaut de page (*Page Fault* en anglais), est générée par le matériel. Son effet consiste à donner la main au système d'exploitation qui a désormais la charge de trouver un cadre disponible en mémoire centrale dans le but de l'allouer au processus responsable du défaut de page dont il est question. Il devra éventuellement recharger le contenu de ce cadre en utilisant celui disponible sur la mémoire de masse (qui se trouve en principe dans la zone d'échange du disque dur ou *swap* en anglais).

Il se peut que la mémoire centrale ne possède plus aucun cadre libre. Dans cette éventualité, un algorithme de pagination est responsable de choisir une page « victime ». Soit cette page se verra immédiatement réaffectée au processus demandeur, soit elle sera sauvegardée sur le disque dur, alors que l'entrée de la table des pages qui la référence sera mise à jour.

4.1.1.4.1 Utilisations de la pagination

La relocation est gérée tout simplement en fixant l'adresse dans la mémoire virtuelle du code. Il peut par la suite se trouver n'importe où au sein de la mémoire physique.

Le *swap* peut être géré avec une granularité d'une page. Le système d'exploitation peut sauvegarder une page sur le disque, puis marquer cette page comme non valide. Si les données de cette page doivent être accédées, le MMU (cf. section 4.1.1.2.4, MMU) va lever une erreur et donner le contrôle au système d'exploitation. Ce dernier peut alors relire les données du disque, les mettre dans un cadre de page libre et indiquer le nouveau cadre de page dans la table.

Réaliser une opération de *swapping* peut ainsi se faire de manière fine, en ne transférant qu'une partie des données d'un programme (celles qu'il utilise le moins souvent, si possible).

Au même titre que pour la segmentation, la protection est implémentée de deux façons. En premier lieu, tout cadre de page qui ne peut être atteint par aucune page ne peut pas être vu par une application. Ensuite, chaque page possède des protections, tout comme pour la segmentation : lecture seule ou non, accessible pour le noyau uniquement ou non. Chaque processus possède donc sa propre table de pages. Cette table de page locale au processus est nommée espace d'adressage.

La mémoire partagée peut être implémentée facilement avec la pagination, en indiquant tout simplement les mêmes cadres de pages pour des pages de deux espaces d'adressage différents. Ces pages peuvent se trouver à des adresses virtuelles différentes.

D'autres usages, comme la possibilité d'utiliser la pagination pour implémenter les fichiers projetés en mémoire (mmap) ou le *Copy-On-Write* (COW), ne seront pas décrits plus en avant dans la présente section.

4.1.1.5 Mode d'adressage réel (real mode)

Le mode réel se trouve être le mode de fonctionnement par défaut des processeurs compatible x86, qui est aujourd'hui désuet, le mode protégé lui étant préféré.

Il est caractérisé par un adressage direct de la mémoire sous forme de segment : *offset* et limité à 1 Mo, de l'adresse hexadécimale 00000 à FFFFF. Dans ce mode, le processeur ne peut faire fonctionner qu'un programme à la fois.

En cas de nécessité, un mécanisme d'interruption permet toutefois de suspendre l'exécution d'un programme pour en favoriser un autre.

4.1.1.6 Mode d'adressage protégé (protected mode)

Le mode protégé, intégré à l'architecture x86 en 1982, vient ajouter au mode réel des fonctionnalités telles que :

- La protection de mémoire par le biais des niveaux de privilèges (cf. section 4.1.1.7, Niveaux de privilèges) ;
- Le support de la mémoire virtuelle (cf. section 4.1.1.2, Principe de la mémoire virtuelle) ;

- La commutation de contexte, qui consiste à sauvegarder et restaurer l'état d'un processeur afin qu'une exécution puisse être reprise plus tard au même point. Plusieurs processus peuvent ainsi partager la même unité centrale. Il s'agit donc d'un mécanisme essentiel dans un contexte multitâche.

4.1.1.7 Niveaux de privilèges

Les niveaux de privilèges sont habituellement incarnés par la notion d'anneaux de protection. Ces derniers sont généralement inclus dans la plupart des architectures modernes de processeurs (comme l'Intel x86).

Le concept des anneaux de protection a été mis en œuvre au sein du système d'exploitation Multics, un prédécesseur fortement sécurisé de la famille des systèmes d'exploitation UNIX®.

Ces anneaux sont généralement disposés au sein d'une hiérarchie allant du plus privilégié (le plus sécurisé, généralement appelé *Ring0*) au moins privilégié (le moins sécurisé, portant le numéro le plus élevé).

Le matériel connait en permanence l'anneau de privilège courant des actions en cours d'exécution, et ce, grâce à un registre spécial du processeur. Connu sous le nom de RFLAGS (registre de drapeaux ou fanions) dans le cas des processeurs x86-64 (64 bits), compatible avec les registres EFLAGS et FLAGS hérités des familles x86 (32 bits) et précédentes (16 bits), il permet de connaître l'état du processeur à tout moment grâce aux différents bits qui le composent. L'un d'eux, IOPL (*I/O Privilege Level*) contient un nombre précisant le niveau de privilège du programme en cours d'exécution.

Les manières dont la main peut être passée d'un anneau à l'autre sont sévèrement limitées par le matériel qui impose également des restrictions aux types d'accès mémoire pouvant être exécutés au travers des anneaux. Ces restrictions matérielles ont été conçues pour limiter les possibilités d'infractions accidentelles ou malveillantes de nature à compromettre la sécurité du système.

L'utilisation efficace de cette architecture en anneaux exige une coopération étroite entre le système d'exploitation et le matériel (plus particulièrement le processeur). Généralement, seuls deux anneaux de sécurité sont exploités, s'agissant du mode noyau (ou *kernel* en anglais) et du mode utilisateur.

L'objectif consiste à empêcher divers composants applicatifs d'un système d'exploitation de se modifier les uns les autres, situation que nous pouvons considérer comme une faille sérieuses de sécurité. Les composants qui nous intéressent tous particulièrement, dans le cadre du système d'exploitation, sont précisément le noyau et les applications. S'il est tout à fait envisageable que le noyau puisse apporter certaines modifications aux données dynamiques appartenant à une application, le contraire ne doit pas se produire.

Processeurs x86

Les processeurs de la famille x86 implémentent quatre anneaux de privilèges mais, comme précisé plus haut, seuls deux d'entre eux sont réellement exploités par les systèmes d'exploitation actuels (comme Windows® ou les dérivés d'UNIX®).

Ces deux anneaux de protection sont le ring0, soit celui ayant le plus de privilèges, et le ring3. Comme illustré par la figure 4-3, les privilèges octroyés vont décroissant de l'anneau 0 à l'anneau 3. Le noyau du système d'exploitation fonctionne au niveau du premier, les interactions avec le matériel (comme l'unité centrale ou la mémoire) intervenant généralement à ce niveau. Les applications fonctionnent, quant à elle, au niveau du deuxième.

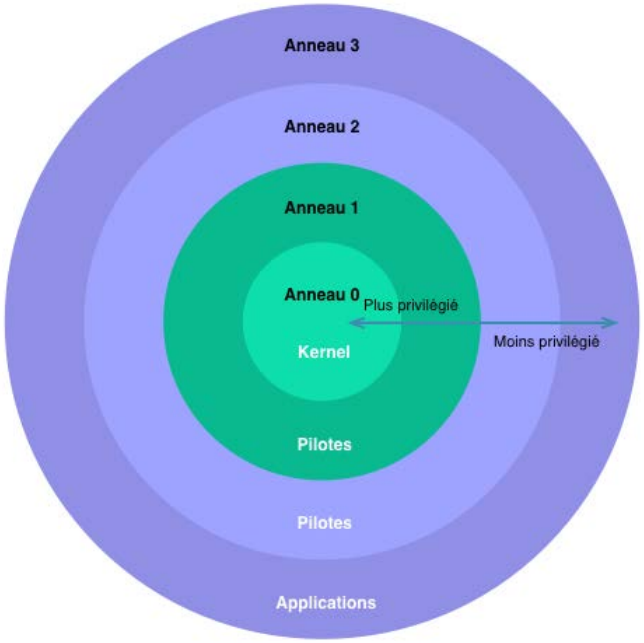


Figure 4-3 : Anneaux de protection (Source : fr.wikipedia.org/wiki/Anneau_de_protection)

La transition d'un mode à l'autre est assurée par l'instruction SYSENTER (langage assembleur).

4.1.1.8 Périphériques d'entrées/sorties

Il peut s'agir d'un disque dur, d'une clé USB, d'un contrôleur d'interface réseau (NIC pour Network Interface Controller), etc.

Le processeur utilise ces périphériques pour communiquer avec le monde extérieur. Il doit donc les gérer, leur transmettre des informations ou en recevoir de leur part. En principe, cette communication s'effectue grâce à des circuits spécialisés appelés interfaces programmables. La figure 4-4 illustre schématiquement un système à microprocesseur, comportant de la mémoire, une interface programmable, ainsi que des périphériques.

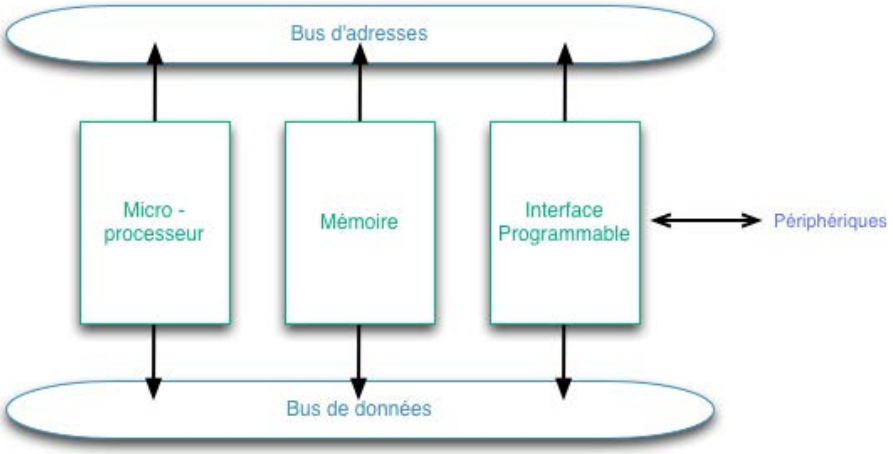


Figure 4-4 : Système de base à microprocesseur (Source : Accès direct en mémoire de R. Beuchat)

Selon le débit des informations à transmettre aux périphériques, plusieurs méthodes sont disponibles pour effectuer leur transfert. Ce choix doit être effectué lors de la conception du circuit.

Les trois principales méthodes existantes sont la scrutation, les interruptions ou les transferts par accès direct en mémoire (DMA pour *Direct Memory Access* en anglais).

La **scrutation** (appelée *polling* en anglais) nécessite une interrogation régulière des interfaces pour voir si ces dernières doivent être servies (par exemple, l'imprimante a-t-elle accepté le caractère qui vient de lui être envoyé).

Cette méthode occasionne une perte considérable de temps pour le processeur, ce dernier interrogeant de manière systématique des interfaces qui n'ont pas besoin d'être servies.

Cette méthode n'est utilisable que dans le cas où un nombre restreint de périphériques sont utilisés, voire dans le cas où le processeur doit servir le même périphérique pendant une courte durée mais à raison d'un taux de transfert important.

Comme son nom l'indique, la méthode de l'**interruption** consiste à interrompre le processeur chaque fois que cela s'avère nécessaire. Dans les faits, un signal part de l'interface programmable en direction du processeur pour lui signaler qu'elle a besoin d'être servie. Ce signal est communément appelé requête d'interruption (IRQ pour *Interrupt Request* en anglais). Le processeur va alors exécuter une routine de service de l'interruption et servir l'interface.

L'identification de l'interface qui a demandé à être servie doit cependant être identifiée. Cette phase est qualifiée de quittance d'interruption (*Interrupt Acknowledge*). La méthode utilisée pour gérer ladite phase diffère en fonction de la capacité du périphérique à répondre à la demande d'identification générée par le processeur. En effet, seulement certains périphériques s'avèrent être capables de le faire.

Le cas échéant, le périphérique fournit un vecteur d'interruption au processeur, s'agissant ni plus ni moins d'un numéro.

Le signal servant à initier la quittance d'interruption est appelé IACK (acronyme pour *Interrupt Acknowledge*).

Le processeur recherche l'adresse de la routine de traitement de l'interruption au sein d'une table, en comparant le numéro du vecteur d'interruption à celui de la routine en question. Avant de se rendre à l'adresse identifiée, il sauve son état courant, soit son PC (*Program Counter*[22]) et l'état de ses fanions (cf. section 4.1.1.7, Niveaux de privilèges).

Lorsque le processeur est confronté à un périphérique qui n'est pas capable d'émettre un vecteur d'interruption ou si le processeur lui-même n'est pas capable de générer le signal IACK, deux cas de figure sont possibles. Dans le premier cas, nous n'avons qu'un périphérique sollicitant une interruption. Le processeur n'aura pas de problème à savoir qui l'interrompt puisqu'il n'y a qu'un interrompant. Dans le second cas, plusieurs sources d'interruptions existent. La méthode de scrutation sera dès lors utilisée pour identifier la source en question.

La figure 4-5, ci-dessous, illustre le principe de connexion entre l'interface programmable et le processeur. Nous voyons que le vecteur d'interruption est véhiculé par le bus de données, permettant ainsi d'identifier la source de l'interruption.

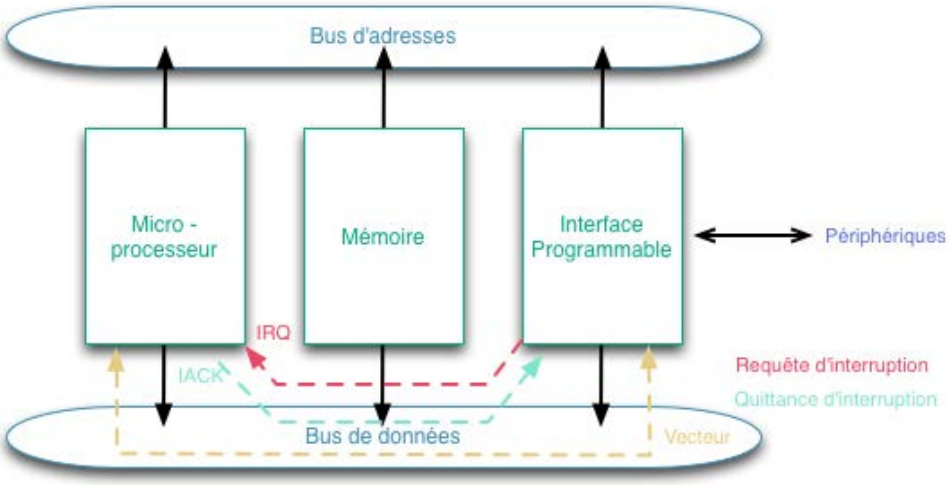


Figure 4-5 : Principe des signaux de requête et quittance d'interruption (Source : Accès direct en mémoire de R. Beuchat)

Quant au **DMA**, il vise en particulier à éviter l'implication du processeur dans la gestion des interfaces, ce dernier n'ayant pas vocation à le faire lors d'un transfert à haut débit. Il perd trop de temps en scrutation ou en traitement d'interruption. Une unité matérielle supplémentaire est capable de récupérer des données de l'interface programmable et de les transférer en mémoire sans passer par le processeur.

[22] Adresse de l'instruction qui était en cours d'exécution lorsque le processeur a été interrompu.

Cette unité peut être un simple circuit appelé contrôleur DMA, ce dernier n'étant rien d'autre qu'une interface programmable autorisée à demander l'accès au bus au processeur pour effectuer des transferts à sa place.

Le schéma 4-6, ci-dessous, illustre le principe de liaison d'un contrôleur DMA avec un microprocesseur et une interface programmable.

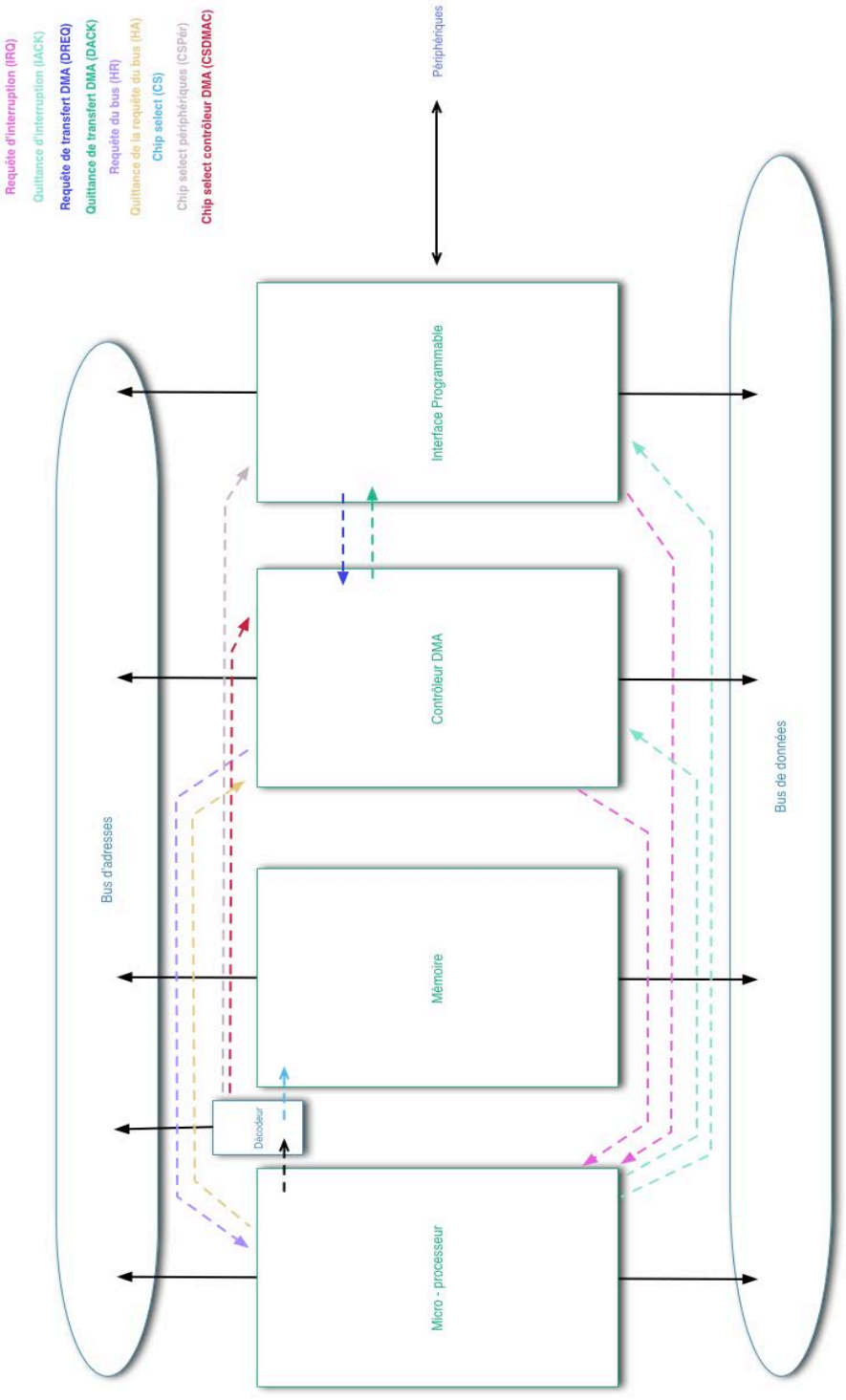


Figure 4-6 : Schéma de principe de liaison d'un contrôleur DMA avec un microprocesseur et une interface programmable (Source : Accès direct en mémoire de R. Beuchat)

4.1.2 Système d'exploitation

Un système d'exploitation n'est en fin de compte qu'un programme (plus précisément un ensemble de programmes) fonctionnant comme tel, mais au niveau du mode privilégié du processeur, appelé généralement mode noyau ou mode superviseur (cf. section 4.1.1.7, Niveaux de privilèges), lui permettant, en l'état, d'interagir avec la couche matérielle par le biais d'instructions sensibles appartenant au jeu d'instructions de l'unité centrale.

Le système d'exploitation est donc tourné vers sa couche inférieure, en émettant des instructions destinées à contrôler les différents périphériques matériels pour les allouer et les gérer comme ressources matérielles à l'attention des programmes.

Il est également tourné vers sa couche supérieure par le biais de la mise à disposition d'une partie de ses services à l'attention des programmes (interface d'appel système).

4.1.3 Application

Les applications sont exécutées par le processeur au sein de l'espace utilisateur (cf. section 4.1.1.7, Niveaux de privilèges) et se servent des appels système (*system call* en anglais, abrégé en syscall), soit de fonctions primitives fournies par le noyau d'un système d'exploitation, lorsqu'elles doivent solliciter le système d'exploitation pour accéder aux ressources matérielles.

L'espace d'adressage virtuel disponible pour un programme est défini par le système d'exploitation, le MMU (cf. section 4.1.1.2.4, MMU) lui permettant de le faire, étant précisé que le système d'exploitation peut parfois faire évoluer à la hausse les quantités d'espace d'adressage virtuel maximales (cf. section 4.1.1.2, Principe de la mémoire virtuelle) initialement prévues par l'architecture du système. Le système d'exploitation donne en quelque sorte l'illusion au programme qu'il possède une mémoire qui lui est propre.

Nous rappelons que, conformément au principe du mode d'adressage protégé (cf. section 4.1.1.6, Mode d'adressage protégé (*protected mode*)), qui implique la mise en œuvre du principe de la mémoire virtuelle et des niveaux de privilèges, les applications ne peuvent empiéter sur l'espace mémoire dévolu au noyau du système d'exploitation.

4.2 La genèse de la virtualisation

4.2.1 Idée originelle

Commençons préalablement par rappeler schématiquement le fonctionnement habituel d'une machine physique (cf. figure 4-7).

Comme nous l'avons vu à la section précédente, les applications se servent des appels système pour obtenir le concours du système d'exploitation lorsqu'elles nécessitent d'accéder aux ressources physiques de la machine.

En effet, conformément aux principes abordés au sein des sections 4.1.1.6 (cf. section 4.1.1.6, Mode d'adressage protégé (*protected mode*)) et 4.1.1.7 (cf. section 4.1.1.7, Niveaux de privilèges), seul le noyau du système d'exploitation peut accéder aux ressources matérielles.

Lorsqu'une application utilise une telle primitive, le processeur bascule du mode utilisateur au mode superviseur, permettant ainsi au noyau du système d'exploitation de lui fournir les ressources matérielles nécessaires.

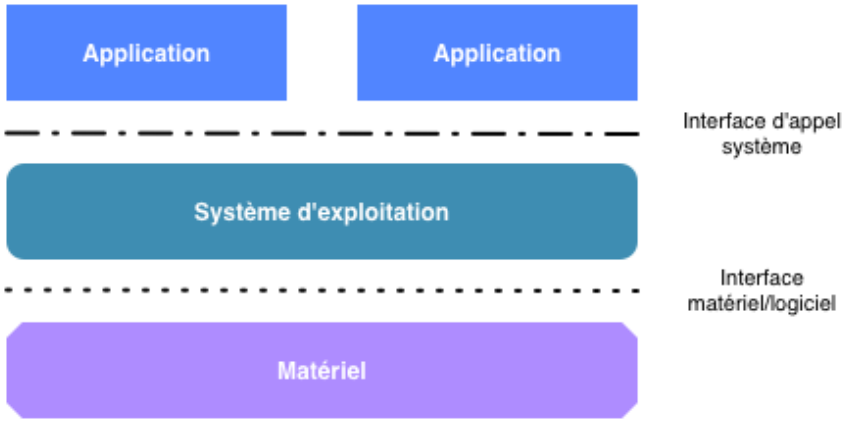


Figure 4-7 : Couches d'une machine physique (Source : Virtualization de Steve Gribble)

En toute logique, nous pourrions parfaitement envisager d'ajouter une couche de virtualisation, soit une interface, entre le matériel et les systèmes d'exploitation invités (donc virtuels), conformément à ce que nous voyons à la figure 4-8. Cette méthode consisterait à faire fonctionner les systèmes d'exploitation invités comme une application quelconque, au niveau utilisateur (*ring3)*, la couche de virtualisation opérant au niveau superviseur (*ring0*).

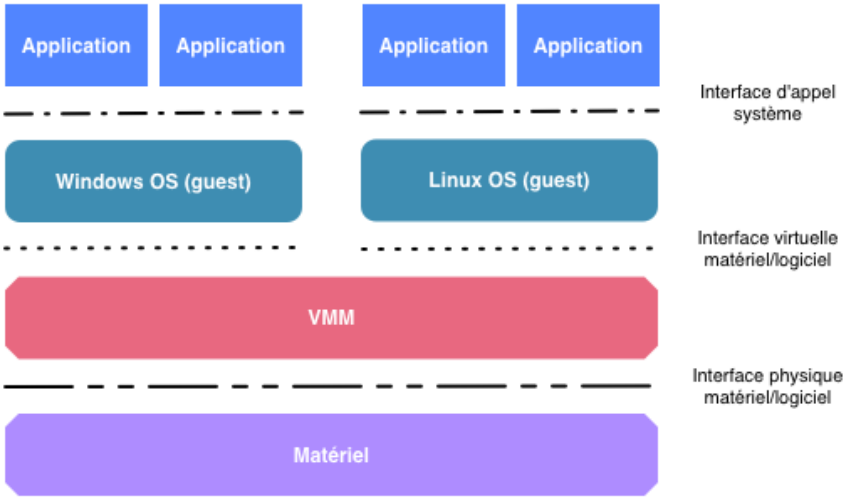


Figure 4-8 : Couches et interfaces avec VMM (Source : Virtualization de Steve Gribble)

Quoique pertinente, cette manière de faire laisse toutefois entrevoir une difficulté potentielle. Un système d'exploitation est en effet prévu pour fonctionner au niveau noyau (*ring0*) et répartir les ressources du processeur entre les différentes applications en utilisant un certain nombre d'instructions sensibles (cf. section 4.2.2.2, Classification des instructions processeur). Lorsqu'il se trouve en situation d'invité, sur un VMM, ce même système d'exploitation ne doit pas pouvoir modifier les ressources matérielles, sous peine de provoquer un plantage du système. Ce privilège doit rester l'apanage du seul VMM.

En d'autres termes, comment être certain qu'une application fonctionnant sur un système d'exploitation invité ne cause des problèmes à ce dernier, le principe des niveaux de privilèges n'étant pas intégralement respecté ? L'application précédemment évoquée pourrait d'ailleurs générer des problèmes au niveau du VMM lui-même. Deux systèmes d'exploitation clients différents (par exemple, Linux™ et Windows®) pourraient se compromettre l'un l'autre ou éventuellement compromettre le VMM.

4.2.2 Exigences liées à la virtualisation (Popek et Goldberg, 1974)

Les exigences de Popek et Goldberg liées à la virtualisation constituent un ensemble de conditions qui doivent être réunies par une architecture informatique pour qu'elle puisse mettre en œuvre la virtualisation. Ces principes ont été publiés par Gerald J. Popek et Robert P. Goldberg en 1974, par le biais de leur article intitulé : « Formal Requirements for Virtualizable Third Generation Architectures »[23] .

Ces exigences représentent donc un moyen pratique de déterminer si une architecture donnée peut prendre en charge efficacement la virtualisation. De plus, elles font office de lignes directrices pour la conception d'environnements virtuels.

Il convient de préciser que, même si les recherches de Popek et Goldberg sont basées sur des architectures de troisième génération[24] (par exemple, IBM System/360, Honeywell 6000, DEC PDP-10), le modèle utilisé est suffisamment général pour être étendu à des machines modernes. Ce modèle comprend un processeur fonctionnant soit en mode utilisateur, soit en mode noyau, et possédant un accès linéaire à une mémoire uniformément adressable. Il est supposé qu'un sous-ensemble du jeu d'instructions est disponible uniquement en mode noyau et que la mémoire est adressée par rapport à un registre de relocalisation[25]. Les entrées/sorties et les interruptions ne sont pas modélisées.

4.2.2.1 Définition d'un VMM

Lorsqu'un VMM est mis en œuvre et afin qu'il puisse être considéré en tant que tel, il doit respecter trois critères. Ces derniers serviront par la suite de base à la définition des théorèmes relatifs à la virtualisation.

La première contrainte à considérer est l'**équivalence**. En effet, toute application exécutée sur le VMM devrait présenter un effet identique à celui démontré dans le cas où il aurait été exécuté directement sur la machine originale. Les seules différences observées devraient avoir pour origine la disponibilité des ressources physiques du système, ces dernières pouvant modifier le temps d'exécution d'un programme.

Le second critère est l'**efficacité**. Il implique qu'un sous-ensemble statistiquement dominant des instructions exécutées par le processeur virtuel, doivent l'être par le processeur réel, sans intervention logicielle du VMM.

[23] http://www.dc.uba.ar/materias/so/2010/verano/descargas/articulos/VM-requirements.pdf.
[24] Il s'agit des ordinateurs à circuits intégrés, généralement produits entre 1963 et 1971.
[25] Fonctionne en lien avec le MMU et contient une constante qui, ajoutée à l'adresse logique utilisée par l'unité centrale, empêche un programme de connaître la valeur réelle utilisée pour accéder à la mémoire.

Il convient donc de préciser que ce critère exclurait *de facto* l'émulation (cf. section 4.3.1.3, Émulation) et les techniques de virtualisation totale (cf. section 4.3.1.1, Virtualisation totale (traduction binaire)).

Le troisième critère est le **contrôle de ressources**. Il sous-tend que le VMM doit bénéficier du contrôle exclusif des ressources qui doivent être partagées. Toute application doit donc passer par le VMM pour pouvoir accéder à une telle ressource.

D'après Popek et Goldberg, un VMM doit se conformer à ces trois propriétés pour être considéré comme tel. D'après l'ouvrage de référence *Virtual Machines : Versatile Platforms For Systems And Processes*[26], la plateforme qui répondrait aux critères d'équivalence et de contrôle des ressources pourrait être considérée comme un VMM, alors que celle qui répondrait également au critère d'efficacité pourrait être considérée comme un VMM efficace.

Popek et Goldberg partent du principe que l'architecture de l'ISA (*Instruction Set Architecture)* de la machine physique chargée d'exécuter le VMM, doit correspondre aux trois critères mentionnés ci-dessus.

4.2.2.2 Classification des instructions processeur

Les prérequis élémentaires mentionnés au sous-chapitre précédent peuvent être considérés comme une base ayant donné lieu à des règles plus spécifiques. Avant de parvenir à l'élaboration de leurs différents théorèmes, Popek et Goldberg ont d'abord élaboré une classification des différentes instructions processeur en trois différents groupes :

- Les instructions **privilégiées** : ces dernières exigent du processeur qu'il se trouve en mode privilégié (mode noyau) pour être exécutées. Si tel n'est pas le cas, elles devront être « piégées » (de l'anglais *trap*) par le système afin d'être traitées par le VMM. Le piège en question, ou exception, consiste en un changement de contexte et un traitement de l'instruction par le VMM qui décide concrètement de la suite qu'il convient de lui donner. Il peut décider de l'ignorer ou d'en émuler son fonctionnement ;
- Les instructions de **contrôle sensibles** : ces dernières tentent de modifier la configuration des ressources physiques au sein du système. Sont incluses dans ce groupes les instructions qui sont susceptibles de modifier la quantité de mémoire allouée à un processus ou celles visant à changer le mode au sein duquel évolue le processeur sans malgré tout être piégée (cf. instructions privilégiées) ;
- Les instructions de **comportement sensibles** : ces dernières sont les instructions dont le comportement ou le résultat dépend de la configuration des ressources, à savoir le contenu du registre de relocalisation ou le mode dans lequel se trouve le processeur. Ce groupe inclut les instructions qui visent à lire une zone de mémoire spécifique ou à se déplacer au sein de la mémoire vive.

Il convient de préciser qu'un groupe supplémentaire d'instructions existe, s'agissant du groupe des instructions non-privilégiées. Ces dernières n'exigent pas un degré de privilèges élevé pour être exécutées. Le processeur pourra dès lors se trouver en mode noyau comme

[26] De James Edward Smith et Ravi Nair.

en mode utilisateur lors de leur exécution. Il s'agit par exemple de certaines opérations élémentaires comme l'addition, la multiplication, etc.

4.2.2.3 Théorèmes de la virtualisation

Après avoir procédé à la définition du VMM et à la classification des instructions processeur, Popek et Goldberg énoncent enfin l'aspect fondamental de leur théorie, à savoir l'élaboration de trois théorèmes, lesquels sont :

Le **premier théorème** porte sur les **conditions de mise en œuvre d'un VMM**, stipulant que pour se faire, les instructions sensibles soient ajoutées au groupe des instructions privilégiées afin que, comme ces dernières, elles soient piégées et traitées par le VMM. La propriété de contrôle de ressources est de ce fait assurée. Les instructions non-privilégiées doivent, quant à elle, être exécutées nativement (i.e. efficacité). La propriété de l'équivalence est ainsi également respectée.

Ce théorème fournit par ailleurs une technique élémentaire pour implémenter un VMM. Cette dernière est appelée ***trap-and-emulate*** et est abordée plus bas (cf. section 4.2.3, *Trap-and-emulation*). En résumé, toutes les instructions sensibles se comportent parfaitement bien tant que le VMM les piège et émule ces dernières. C'est pour cette raison qu'on qualifie souvent cette méthode de **virtualisation classique**.

Il convient de préciser que l'architecture à l'origine de la virtualisation des environnements x86, la virtualisation totale, ne répond pas à ce théorème car un certain nombre d'instructions posent problème. Les travaux de Mendel Rosenblum viseront précisément à contourner ce problème et le succès rencontré par ce dernier seront à l'origine de la création de la société VMware® (cf. section 1.2.2, Nécessité d'une virtualisation x86, section 4.2.5, Virtualiser l'architecture x86 et section 4.3.1.1, Virtualisation totale (traduction binaire)).

Le **second théorème** énonce qu'il est possible de mettre en œuvre une **virtualisation récursive** sur une architecture processeur donnée, pour autant que :

- cette dernière réponde au premier théorème ;
- qu'un VMM sans dépendance temporelle (*timing dependencies*) peut être élaboré pour elle.

Certaines architectures, comme la x86 sans assistance du matériel (cf. section 4.3.3.1, Virtualisation de l'accès au processeur), ne répondent pas à cette condition et ne peuvent dès lors pas être virtualisées selon la manière classique. Ces architectures peuvent cependant être totalement virtualisées (dans le cas du x86 cela signifie au niveau de l'unité centrale et du MMU) en utilisant d'autres techniques comme la traduction binaire qui permet de remplacer les instructions sensibles qui ne génèrent pas de mécanisme d'exception (piège) (cf. section 4.3.1.1, Virtualisation totale (traduction binaire)). Ces dernières sont souvent appelées instructions **critiques**.

Quant au **troisième théorème**, il précise qu'un **moniteur de machine virtuelle hybride** peut être mis en œuvre si le groupe des instructions sensibles exécutables en mode utilisateur est un sous-ensemble du groupe des instructions privilégiées.

4.2.3 *Trap-and-emulation*

Cette méthode, découlant directement du premier théorème de Popek et Goldberg est considérée comme mettant en œuvre la virtualisation au sens **classique** du terme. Il s'agit d'une manière générique de faire qui est destinée à s'appliquer à toute architecture disponible sur le marché.

Le concept répond aux conditions générales suivantes :

- Le VMM doit s'exécuter en mode noyau, la machine virtuelle hébergée s'exécutant, quant à elle, en mode utilisateur. Ces deux niveaux de privilèges doivent donc être mis en œuvre par l'architecture (cf. section 4.1.1.7, Niveaux de privilèges) ;
- La machine réelle dispose d'un mécanisme de réimplantation dynamique d'adresses, lui permettant de mettre en œuvre la mémoire des machines virtuelles hébergées.

De plus, toute instruction sensible (cf. section 4.2.2.2, Classification des instructions processeur) doit provoquer une exception (*trap*) lorsqu'elle est exécutée en mode utilisateur.

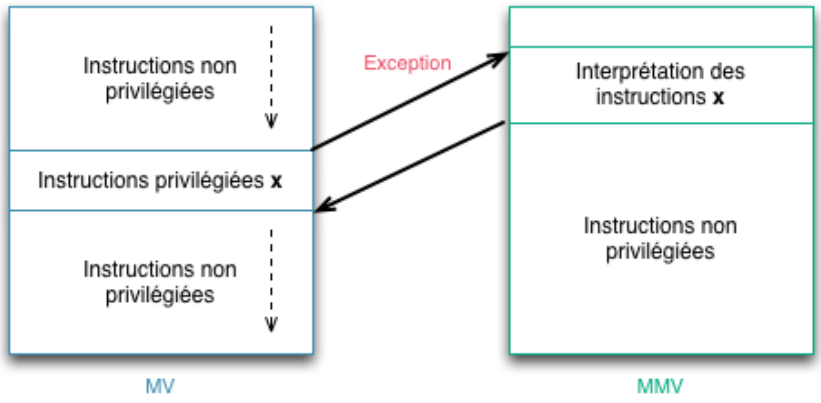


Figure 4-9 : Traitement d'une instruction sensible (Source : Concept de machine virtuelle d'Alain Sandoz)

Après avoir piégé l'instruction sensible en question, le VMM émule son effet, et ce, au niveau du matériel virtuel qu'il fournit au système d'exploitation invité (cf. figure 4-9).

4.2.4 Performance dans la pratique

Le critère d'efficacité abordé dans la définition d'un VMM par Popek et Goldberg ne concerne que l'exécution des instructions non-privilégiées, ces dernières étant exécutées de manière native. C'est d'ailleurs ce qui distingue un VMM d'un logiciel d'émulation matériel.

Malheureusement, même sur une architecture qui répond aux critères de Popek et Goldberg, les performances d'une machine virtuelle peuvent être significativement différentes que celles observées sur une machine réelle. Les premières expériences réalisées sur le System/370 d'IBM®, lequel répondait aux spécifications du premier théorème, ont démontré que les performances de la machine virtuelle pouvaient se situer 21% au-dessous de celles de la machine native (au niveau de certains points de repère en particulier).

Piéger les instructions ayant levé des exceptions et en émuler les effets génèrent en effet un coût substantiel. Les ingénieurs d'IBM® ont traité cette problématique par l'ajout d'un certain

nombre d'assistances de type matériel, à divers niveau. Nous pouvions en compter jusqu'à cent sur les dernières versions du System/370. Il convient de préciser que les solutions de virtualisation contemporaines n'échappent pas à l'assistance matérielle, les fabricants ayant eux-mêmes entrepris de modifier l'architecture de leurs produits dans le but d'introduire un support matériel à la virtualisation (cf. section 4.3.3, Assistance matérielle).

Le concept de mémoire virtuelle lui-même a constitué un jalon important dans le développement de l'assistance matérielle du System/370. En effet, lorsque le système d'exploitation client implémente le principe de la mémoire virtuelle (cf. section 4.1.1.2, Principe de la mémoire virtuelle), même l'exécution des instructions non-privilégiées peut subir un ralentissement, s'agissant d'une pénalité imposée par la nécessité d'accéder aux tables de traduction (*translation tables*) qui ne sont pas utilisées lors de l'exécution native.

4.2.5 Virtualiser l'architecture x86

L'IA-32 (x86) n'est pas l'unique architecture qui ne répond pas aux critères de Popek et Goldberg. Les autres architectures concernées (qui disposent d'instructions sensibles mais non-privilégiées) sont simplement largement moins distribuées que celles appartenant à la famille x86. C'est la raison pour laquelle ce chapitre est spécialement consacré à cette architecture.

L'architecture x86, largement plébiscitée sur le marché des processeurs et équipant, de ce fait, la plupart des ordinateurs (stations de travail comme serveurs), ne répondait pas aux critères de Popek et Goldberg, et ce, jusqu'en 2005.

C'est à cette époque seulement que les leaders du marché des processeurs, Intel® et AMD®, firent évoluer leurs produits (processeurs compatibles x86) pour que ces derniers prennent en charge la virtualisation au niveau matériel (cf. section 4.3.3.1, Virtualisation de l'accès au processeur).

En effet, l'ISA x86 se prête mal à la virtualisation à cause de 17 instructions sensibles, mais non-privilégiées qui ne provoquent pas d'exception lorsqu'elles sont exécutées en mode utilisateur. Ces instructions peuvent être classées en deux groupes :

- Les **instructions sensibles affectant les registres**, qui peuvent lire ou changer des valeurs dans des registres sensibles et/ou dans des zones mémoire, s'agissant par exemple du registre de l'horloge ou des registres d'interruptions :
  - SGDT, SIDT, SLDT ;
  - SMSW ;
  - PUSHF, POPF ;
- Les **instructions relatives à la protection du système**, qui référence le système de protection de la mémoire ou le système de relocation d'adresses :
  - LAR, LSL, VERR, VERW ;
  - POP ;
  - PUSH ;
  - CALL, JMP, INT n, RET ;
  - STR;
  - MOV.

Nous constatons dès lors que les instructions de cet ISA ne sont donc pas toutes sur un pied d'égalité. Les instructions mentionnées ci-dessus **peuvent modifier la configuration des ressources du processeur**. Nous rappelons donc qu'elles doivent être interceptées. Comme nous l'avons vu précédemment, ce type de traitement est évident pour toutes les instructions privilégiées, le système d'exploitation étant alors exécuté en anneau 3, à l'instar des applications. Dans ce cas, toutes les requêtes d'instructions privilégiées déclenchent une erreur systématiquement traitée par le VMM. C'est nettement plus compliqué pour les 17 instructions **critiques** (sensibles et non privilégiées) évoquées plus haut, car ces dernières **ne déclenchent pas d'exception**. Elles doivent dès lors être détectées au coup par coup par le VMM, puis réinterprétées.

Différentes techniques permettent de virtualiser les architectures qui ne répondent pas aux critères de Popek et Goldberg, dont l'architecture x86 en mode d'adressage protégé. Nous aborderons certaines d'entre elles à la section 4.3, qu'elles soient logicielles comme la traduction binaire (utilisée à l'origine par WMware®) et la paravirtualisation, ou matérielles comme l'évolution du jeu d'instructions des processeurs.

4.3 Techniques de virtualisation

4.3.1 Virtualisation logicielle

4.3.1.1 Virtualisation totale (traduction binaire)

La traduction binaire (*Binary translation*) est une technique d'émulation consistant à traduire le jeu d'instructions d'une architecture source vers celui d'une architecture de destination. Cette technologie permet à un VMM de virtualiser une plateforme **non-modifiée** (donc intégralement virtualisée). Nous faisons dans ce cas référence au concept de virtualisation totale ou complète.

La machine virtuelle invitée n'a donc absolument pas « conscience » d'être virtuelle dans un tel contexte. Le système d'exploitation invité bénéficie d'un environnement physique émulé à son attention par le VMM. Seul ce dernier est à même de démarrer une machine virtuelle. Il est également le seul à être autorisé à accéder à la couche matérielle.

La figure 4-10, ci-dessous, illustre le principe de la virtualisation totale. Un système d'exploitation est installé de manière classique sur la couche matérielle. Un logiciel de virtualisation (par exemple, VMware Workstation™) fait ensuite office de VMM et émule autant de matériel virtuel qu'il y aura de machine virtuelle. Nous faisons référence, dans ce cas précis, à la **virtualisation hébergée**.

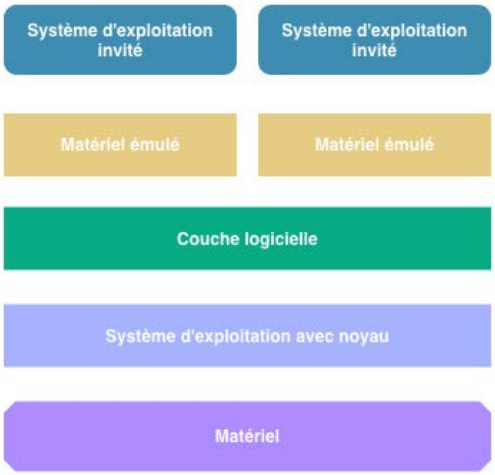


Figure 4-10 : Virtualisation totale correspondant à une mise en œuvre de type virtualisation hébergée (Source : Les différents types de virtualisation : La virtualisation totale par Antoine Benkemoun - http://www.antoinebenkemoun.fr/2009/07/les-differents-types-de-virtualisation-la-virtualisation-totale)

Pour se faire, le VMM est séparé des systèmes d'exploitation des machines virtuelles en étant placé au niveau du *ring0*. Quant aux systèmes d'exploitation des machines virtuelles, ils sont positionnés en *ring1*. Les applications fonctionnant au niveau des systèmes d'exploitation des machines invitées restent exécutées au *ring3* (cf. section 4.1.1.7, Niveaux de privilèges). Cette organisation est cependant propre à l'architecture x86.

La figure 4-11 illustre le principe des anneaux de protection. À gauche, la virtualisation n'est pas mise en œuvre. Le noyau évolue donc en anneau 0 (mode superviseur) et les applications en anneau 3 (mode utilisateur). À droite, dans un contexte de virtualisation, l'hyperviseur prend en quelque sorte la place du superviseur qui évolue cette fois en anneau 1. Quant aux applications, elles sont toujours exécutées en anneau 3.

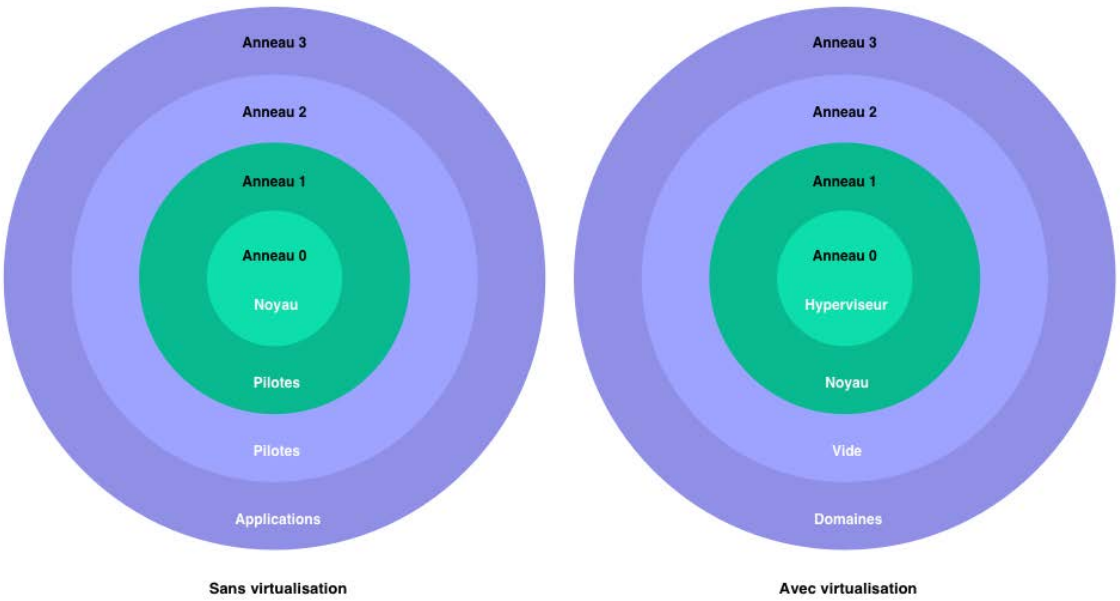


Figure 4-11 : Anneaux de protection avec ou sans virtualisation (Source : Les anneaux de protection système par Antoine Benkemoun - http://www.antoinebenkemoun.fr/2009/08/les-anneaux-de-protection)

La figure 4-12 illustre à nouveau les principes décrits au précédent paragraphe mais sous forme de couche, en faisant correspondre ces dernières avec les différents anneaux. Le schéma met également en évidence les instructions critiques à l'attention du processeur de la machine virtuelle qui sont interceptées à la volée au niveau de la couche de virtualisation puis traduites à l'aide de la traduction binaire afin de pouvoir être traitées par le processeur physique du serveur hôte.

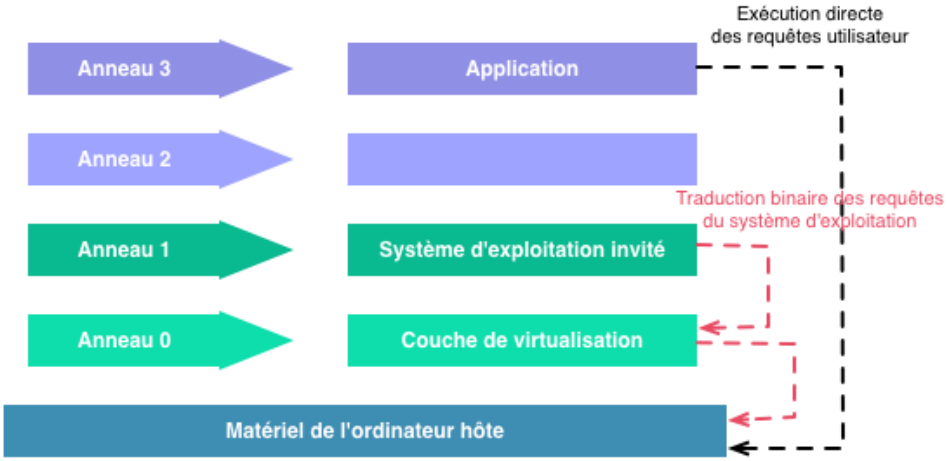


Figure 4-12 : Virtualisation à l'aide de la technique de la traduction binaire, du point du vue des anneaux de protection (Source : La virtualisation des serveurs x86 par John Marcou - http://root-lab.fr/2011/06/04/la-virtualisation-de-serveurs-x86)

Lorsqu'AMD® introduit sur le marché l'extension du jeu d'instructions x86 d'Intel® (AMD64), reprise par Intel® sous le nom de Intel 64, le nombre d'anneaux est réduit à deux. En effet, hormis certains d'entre eux (comme OS/2 notamment qui en utilisait trois), la plupart des systèmes d'exploitation n'utilisaient que l'anneau 0 et l'anneau 3. Il était cependant capital, pour les solutions de virtualisation – la virtualisation totale n'étant pas la seule concernée par cette problématique – d'utiliser un troisième cercle pour cloisonner le VMM.

Il a donc été décidé de mutualiser l'anneau 3 entre les applications et les systèmes d'exploitation, l'hyperviseur étant situé au sein de l'anneau 0 (cf. figure 4-13).

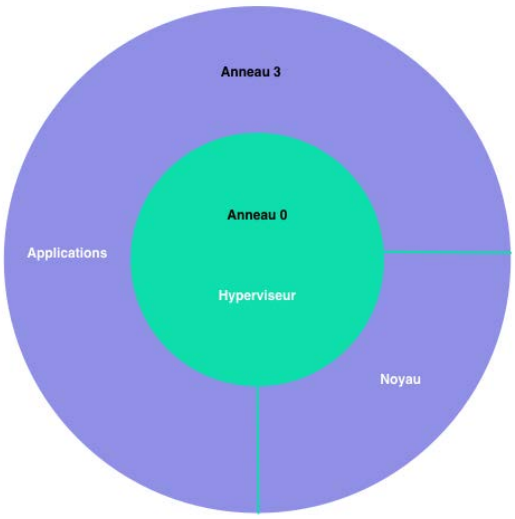


Figure 4-13 : Mutualisation de l'anneau 3 sur architecture 64 bits (Source : Les anneaux de protection système dans le cas du 64-bit par Antoine Benkemoun - http://www.antoinebenkemoun.fr/2009/08/les-anneaux-de-protection-systeme-dans-le-cas-du-64-bit)

AMD® et Intel®, conscients de l'intérêt croissant porté par les différents acteurs du marché de la virtualisation, ont finalement décidé d'inclure dans leurs processeurs des instructions propres à la virtualisation (cf. section 4.3.3.1, Virtualisation de l'accès au processeur). Par la même occasion, un anneau **-1** a été ajouté afin d'éviter la mutualisation de l'anneau 3, ce qui a permis de revenir à une disposition plus aboutie des composants applicatifs parmi les anneaux de protection (cf. figure 4-14).

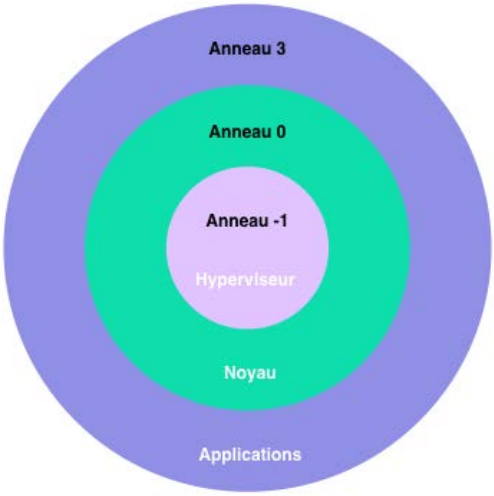


Figure 4-14 : Ajout d'un anneau supplémentaire, destiné à l'hyperviseur (Source : Les anneaux de protection système dans le cas du 64-bit par Antoine Benkemoun - http://www.antoinebenkemoun.fr/2009/08/les-anneaux-de-protection-systeme-dans-le-cas-du-64-bit)

Comme nous nous le laissions précédemment entendre, la traduction binaire est utilisée pour traiter les instructions critiques référencées à la section 4.2.5, Virtualiser l'architecture x86. Nous avons vu que ces instructions, lorsqu'elles proviennent du système d'exploitation invité et qu'elles sont destinées au processeur virtuel, sont en quelque sorte interceptées à la volée puis traduites afin de pouvoir être exécutées par le processeur physique (cf. figure 4-15).

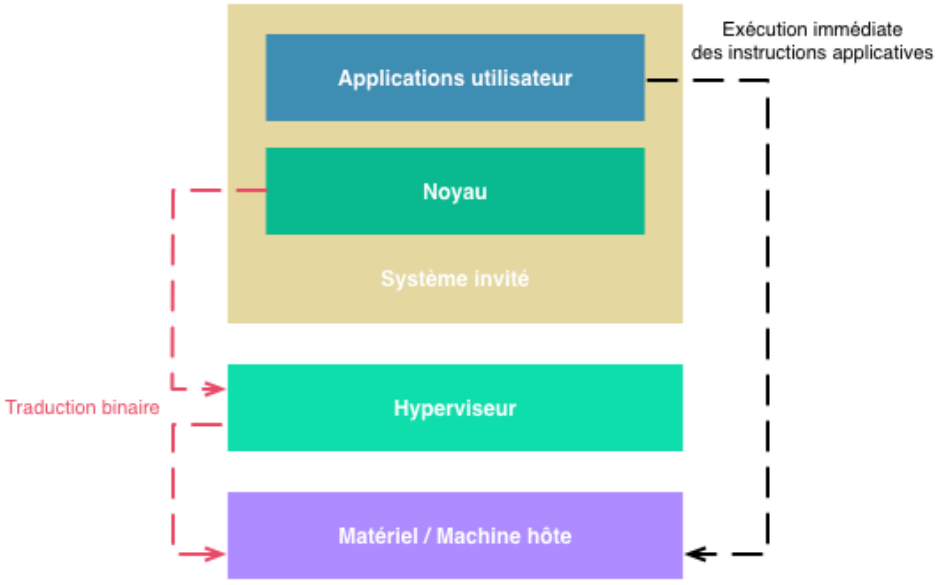


Figure 4-15 : Virtualisation complète avec traduction binaire (Source : Questions actuelles d'informatique, La virtualisation de François Santy et Gaëtan Podevijn)

Il convient de préciser qu'en vertu des critères de Popek et Goldberg, cette technique de virtualisation n'en est *a fortiori* pas une, puisqu'elle ne répond pas au critère d'efficacité. En effet, le VMM exécute de manière logicielle un nombre significatif d'instructions.

Nous précisons que l'architecture physique émulée présente pour avantage un haut niveau de compatibilité avec la plupart des systèmes d'exploitation. Il s'agit d'un élément primordial lors de l'installation d'une machine invitée, lorsque cette dernière ne bénéficie pas encore des pilotes de la machine virtuelle. Toutefois, cette technique n'est destinée qu'à la virtualisation d'un système dont l'architecture est similaire à celle de la machine physique hôte. Un système fonctionnant par exemple sur une architecture Intel x86 ne pourra pas virtualiser un système compilé pour une architecture ARMv8.

Cette technique a été portée au niveau de l'architecture processeur x86 par WMware® en 1999, ce qui a contribué de manière significative à l'encrage de la firme comme un des éditeurs leader du marché de la virtualisation.

Les deux principaux types de traduction binaire sont la traduction binaire statique et la traduction binaire dynamique.

4.3.1.1.1 Traduction binaire statique

Un traducteur fonctionnant sur la base de cette technique, convertit la totalité du code d'un fichier exécutable provenant de l'architecture source en un code qui fonctionne sur l'architecture de destination, sans avoir à exécuter le code préalablement (comme c'est d'ailleurs le cas avec la traduction binaire dynamique). Cette technique est difficile à mettre en œuvre, dès lors que le traducteur ne peut découvrir de manière systématique la totalité du code. En effet, certaines parties d'un code exécutable ne peuvent être atteignables qu'au travers de branches indirectes dont la valeur n'est connue qu'une fois le code en cours d'exécution.

Généralement, l'exécution d'un code exécutable par le biais de cette technique offre des performances se situant en deçà de la rapidité observée nativement.

4.3.1.1.2 Traduction binaire dynamique

La traduction binaire dynamique se fait par sélection de petites séquences de code – typiquement de l'ordre d'un seul bloc de base – puis par mise en cache du résultat de la traduction de ces courtes séquences. Le code est seulement traduit comme il est découvert et quand cela s'avère possible. Des instructions de branchement sont prévues pour qu'il soit possible de pointer vers les parties de code déjà traduites et enregistrées (*Memoization*).

La traduction binaire dynamique diffère de l'émulation car elle élimine la « boucle lecture-décodage-exécution » propre à cette technique - qui constitue *de facto* un goulot d'étranglement majeur – au prix toutefois d'un important surcroit de consommation implicite (*overhead*). Ce surcoût est heureusement amorti par le fait que les traductions de séquences de code sont exécutées plusieurs fois par la suite.

Il convient de préciser que les traducteurs binaires dynamiques les plus avancés se servent de la technique de la recompilation dynamique. Le code est étudié afin que les parties

exécutées le plus grand nombre de fois puissent être mises en évidence, ces dernières étant ensuite drastiquement optimisées.

4.3.1.2 Paravirtualisation

La paravirtualisation consiste à modifier les instructions processeur du système d'exploitation invité au niveau du noyau, afin que ce dernier puisse communiquer directement avec l'hyperviseur (VMM). Nous qualifions dès lors ces instructions d'hyper-appels (*hyper-calls*). Dans ce cas, l'hyperviseur fournit une interface d'hyper-appels pour les opérations critiques du noyau, évoquées maintes fois (celles relatives notamment à la gestion des interruptions et à celle de la mémoire). Dans ce cas de figure, les traditionnels appels système sont remplacés.

Conformément à ce que nous avons déjà pu observer dans le cadre de la virtualisation totale, l'hyperviseur est également positionné au niveau de l'anneau 0, le système d'exploitation invité au niveau 1, les applications, quant à elles, étant exécutées au niveau 3.

La figure 4-16 illustre le cheminement des hyper-appels et met en relation les différentes couches avec les anneaux de protection correspondants.

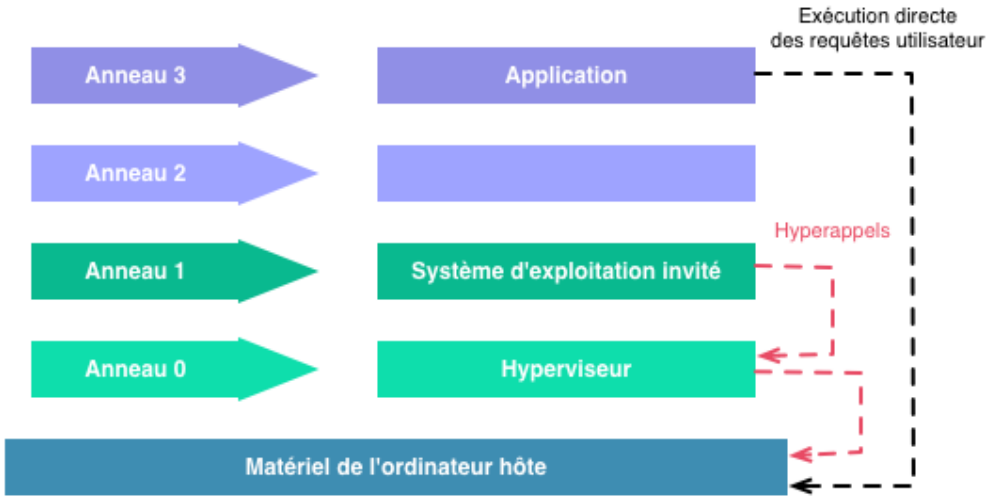


Figure 4-16 : Virtualisation à l'aide de la paravirtualisation, du point de vue des anneaux de protection (Source : La virtualisation de serveurs x86 par John Marcou - http://root-lab.fr/2011/06/04/la-virtualisation-de-serveurs-x86)

Cette technique a l'avantage d'améliorer les performances du système invité, les dispositifs de bas niveau étant directement accessibles par ce dernier par l'intermédiaire de pilotes paravirtualisés. Son rendement est pratiquement comparable à celui d'un système d'exploitation directement installé sur la machine physique.

Contrairement au scénario en vigueur dans le cadre de la virtualisation totale, le système d'exploitation invité a ici « conscience » d'être virtualisé. En d'autres termes, ce dernier doit être « porté », soit adapté pour être à même de communiquer avec l'hyperviseur plutôt qu'avec la machine physique. La figure 4-17, illustre ce « portage ». Les noyaux modifiés des systèmes d'exploitation invités sont à même de communiquer directement avec l'hyperviseur, ce dernier se chargeant d'interagir avec le matériel.

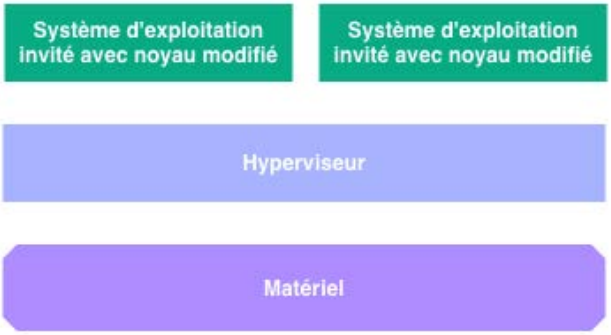


Figure 4-17 : Paravirtualisation basée sur l'existence d'un hyperviseur (Source : La paravirtualisation par Antoine Benkemoun - http://www.antoinebenkemoun.fr/2009/08/la-paravirtualisation)

La conséquence principale de cette manière de faire réside dans le fait que si l'éditeur du système d'exploitation que nous désirons virtualiser n'y a pas apporté les modifications nécessaires, il devient impossible de l'utiliser dans un tel contexte. En outre, des complications liées à la maintenance et au support peuvent se manifester, le noyau du système d'exploitation ayant été modifié.

Par conséquent, cette technique de virtualisation n'était pas du tout adaptée lorsqu'il s'agissait de virtualiser des environnements propriétaires (Windows®, par exemple). Cette méthode s'avère plutôt pertinente lorsqu'il s'agit de virtualiser des environnements libres (Linux™, par exemple), leur noyau pouvant être modifié sans autre forme de procès. C'est la raison pour laquelle les parties prenantes au *Xen Project*[27], dont le but était précisément la mise au point d'un hyperviseur, se sont passablement appuyés sur le monde des systèmes d'exploitation libres pour parvenir à sa mise au point.

Nous précisons toutefois qu'il serait incorrect de réduire l'usage des solutions basées sur la paravirtualisation au monde des systèmes d'exploitation libres. En effet, depuis l'arrivée de nouveaux processeurs implémentant de nouvelles instructions propres à la virtualisation (cf. section 4.3.3.1, Virtualisation de l'accès au processeur), les environnements non-modifiés – comme Windows® – peuvent également être virtualisés selon cette méthode.

4.3.1.3 Émulation

L'émulation peut être considérée comme la capacité à substituer un dispositif informatique par un logiciel. Il est ainsi possible de faire fonctionner un système d'exploitation invité sur un système d'exploitation hôte alors même que l'architecture de leur processeur respectif n'est pas similaire. Nous pourrions dès lors envisager l'installation d'un système d'exploitation Solaris™ compilé pour l'architecture SPARC (*Scalable Processor ARCitecture*) sur un système compilé pour un processeur x86. L'ordinateur entier est donc émulé.

Dans ce cas de figure, le système invité n'a pas « conscience » de son statut de système invité, pas plus que le fait qu'il fonctionne dans une architecture qui lui est étrangère.

L'émulation matérielle, d'ordinaire particulièrement lente, peut être considérée comme viable en tant qu'option de virtualisation de par le fait que les logiciels d'émulation et pilotes

[27] http://www.xen.org/.

récents, ainsi que les processeurs 64 bits des machines hôtes, offrent une qualité d'infrastructure adéquate.

Ce type de technologie est tout à fait pertinente pour les développeurs qui doivent travailler sur l'élaboration de pilotes ou de diverses autres technologies prévues pour des plateformes que les moyens de l'entreprise qui les emploie ne permettent pas d'acquérir ou de supporter, faute de personnel en suffisance.

4.3.2 Virtualisation au niveau noyau

Cette technique consiste en la mise en œuvre d'un noyau (généralement Linux™) compilé de telle sorte qu'il puisse être utilisé dans le niveau de privilège correspondant à l'utilisateur (s'agissant dès lors d'un **noyau invité**). Cette technique permet donc de bénéficier de plusieurs machines virtuelles invitées, appelées généralement **système de fichiers racine** sur un hôte unique.

4.3.3 Assistance matérielle

Le concept de virtualisation ou assistance matérielle, très similaire à la virtualisation du système d'exploitation (et en particulier de la machine sous-jacente), divise le matériel en segments indépendants et gérés individuellement. Nous précisons que dans certains cas de figure, la virtualisation matérielle est requise pour que la virtualisation d'une machine (station de travail ou serveur) puisse se faire.

4.3.3.1 Virtualisation de l'accès au processeur

Le fait que l'essor de la virtualisation de serveurs ou autres stations de travail ait été conséquent depuis le début des années 2000 a poussé les constructeurs de processeurs, tels qu'Intel® et AMD®, à faire évoluer leurs produits en conséquence. Ces processeurs sont respectivement connus sous les appellations d'**Intel® VT-x**, **VT-c** ou **VT-d** pour Intel® (mis en vente en 2005) et **AMD-V™** pour AMD® (mis en vente en 2006).

Intel® VT-x apporte par exemple deux types d'innovations majeures pour les techniques actuelles de virtualisation des architectures IA-32 et Intel 64 (et compatibles). **VMX** et un concept de **bascule assistée**.

VMX

La première innovation est relative aux niveaux de privilèges ou anneaux de protection. Nous y avons fait référence à plusieurs reprises dans le présent document (cf. section 4.1.1.7, Niveaux de privilèges, section 4.3.1.1, Virtualisation totale (traduction binaire) et section 4.3.1.2, Paravirtualisation).

La technologie VT introduit un nouveau mode d'exécution au sein des processeurs concernés, connue sous le nom de VMX[28]. Ce dernier comporte un niveau racine (*VMX root operation*) et un niveau correspondant aux anciens anneaux 1 à 3 (*VMX non-root operation*). Le niveau *VMX root operation* est destiné à l'hyperviseur alors que le niveau VMX non-root operation fournit un environnement contrôlé par le VMM et conçu pour supporter une VM.

[28] http://www.intel.com/technology/itj/2006/v10i3/1-hardware/5-architecture.htm.

Chaque niveau d'opération comporte les quatre niveaux de privilèges (*ring0* à *ring3*). Le noyau du système d'exploitation invité peut donc évoluer au sein de l'anneau 0, les applications exécutées en son sein, au niveau 3, le VMM disposant de son propre jeu d'anneaux (parfois appelé l'anneau **-1**).

La figure 4-18 illustre le fonctionnement de la technologie VMX, tel que décrit ci-dessus.

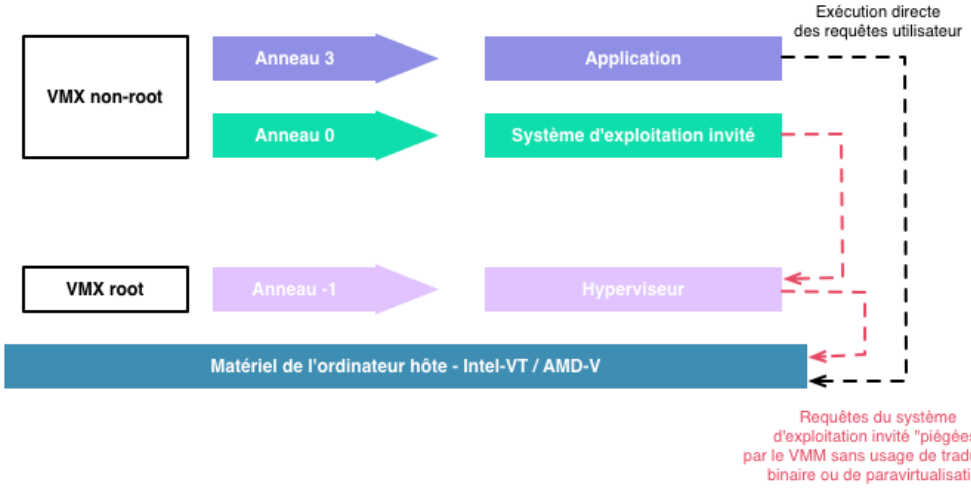


Figure 4-18 : Technologie VMX et anneaux de protection (Source : La paravirtualisation par Antoine Benkemoun - http://www.antoinebenkemoun.fr/2009/08/la-paravirtualisation)

Nous observons donc que la modification du noyau des systèmes d'exploitation invités (paravirtualisation) ou la traduction binaire n'est plus nécessaire (quoi que cette dernière reste nécessaire pour d'autres besoins).

Bascule assistée

Le deuxième apport propre à cette technologie est lié au partage des ressources entre systèmes d'exploitation invités. Ces derniers peuvent s'exécuter sans connaissance aucune de la présence d'autres systèmes invités. L'hyperviseur fournit une tranche de temps d'exécution à un système invité avant de le suspendre pour en attribuer une à un autre invité. Ce processus est répété inlassablement. Le contexte des invités doit donc être sauvegardé et restauré lors de chaque « bascule » (par exemple, l'état des registres et des caches).

VT-x définit ainsi deux nouvelles **transitions** : la première, appelée **VM entry**, est une transition du VMX root operation vers le VMX non-root operation. La deuxième, appelée **VM exit**, est une transition du VMX non-root operation vers le VMX root operation. Ces deux transitions sont gérées par le Virtual Machine Control Structure (VMCS), un segment mémoire de quelques kilo-octets réservé dans l'espace d'adressage physique du système sous-jacent, qui contient une *guest-state area* (zone d'état invité) et une *host-state area (*zone d'état hôte). Chacune de ces zones possède des champs contenant les différents composants de l'état processeur. Un VM entry charge l'état du processeur depuis la zone d'état invité alors qu'un VM exit sauvegarde l'état du processeur dans la zone d'état invité puis charge l'état du processeur depuis la zone d'état hôte.

VM entry et VM exit ne sont pas les seules nouvelles instructions dont dispose l'hyperviseur. VM launch, VM resume, ainsi que d'autres instructions de contrôle existent désormais pour effectuer plus rapidement les opérations évoquées plus haut.

Les opérations du processeur changent substantiellement au sein du VMX non-root operation. Les changements les plus importants résident dans le fait que plusieurs instructions et événements provoque des VM exits. Certaines instructions (par exemple, INVD) causent des VM exits inconditionnels et par conséquent ne peuvent jamais être exécutées dans le VMX non-root operation. D'autres instructions (par exemple, INVLPG) et tous les événements peuvent être configurés pour faire de même mais conditionnellement en utilisant le VM-execution control fields du VMCS. La main peut ainsi passer de l'hyperviseur à une machine virtuelle et inversement de manière souple.

Cette technologie permet un niveau de contrôle fin à l'hyperviseur pour gérer le partage de temps processeur entre les différents systèmes invités à chaque fois qu'un état déterminé du processeur – un déclencheur qui indique qu'une interruption est possible – est atteint.

Elle présente aussi l'avantage d'être peut coûteuse en ressources puisque cette structure de contrôle est gérée au niveau du processeur et non par la couche logicielle.

Les spécificités d'AMD-V étant très proches de celles de l'Intel® VT-x, nous ne procéderons pas à la sa description précise dans la présent chapitre. Toutes les informations relatives aux processeurs d'AMD® comportant cette technologie étant bien évidemment disponibles sur Internet[29].

Les technologies Intel® VT-c et VT-d seront abordées plus précisément à la section 4.3.3.3, Virtualisation de l'accès aux périphériques d'entrées/sorties, puisqu'elles concernent respectivement l'amélioration des performances au niveau des entrées/sorties de type réseau (soit la virtualisation de la communication) et l'accès direct aux différents chipsets de la machine hôte par la machine invitée.

4.3.3.2 Virtualisation de l'accès à la mémoire

Dans le contexte classique d'une machine physique, seul un système d'exploitation est installé. Ce dernier gère dès lors l'adressage de la totalité de la mémoire vive. En résumé, il décide d'en allouer une quantité suffisante à chaque application en cours d'exécution. L'espace correspondant à cette dite quantité sera compris entre un certain bit d'adresse (appelons-le X) et un autre (appelons-le Y).

Cette adressage est réalisé sur la base du principe de la mémoire virtuelle (concept abordé à la section 4.1.1.2, Principe de la mémoire virtuelle) qui représente logiquement la mémoire physique disponible, qu'elle soit répartie sur une ou plusieurs barrette(s), voire sur le disque dur (*swap*). Nous avons vu que la correspondance entre cette mémoire virtuelle et la mémoire physique était assurée par le MMU qui, pour se faire, maintient à jour la table des pages (cf. section 4.1.1.4, Pagination).

[29] http://www.amd.com/us/press-releases/Pages/Press_Release_98372.aspx.

Sur un ordinateur où fonctionnent parallèlement plusieurs systèmes d'exploitation, il est évident que chacun d'entre eux ne peut avoir accès qu'à une partie seulement de l'espace totale de mémoire vive. C'est là qu'intervient le VMM, au moment de la création par ses soins d'une VM. Dans un contexte idéal, le système d'exploitation n° 1 devrait utiliser les adresses allant de 0 à X et le système d'exploitation n° 2, les adresses allant de X + 1 à Y. Malheureusement, un système d'exploitation débute toujours son adressage à l'adresse 0, des conflits entre les différents systèmes d'exploitation pouvant dès lors survenir.

Pour éviter une telle éventualité, il est donc nécessaire que le VMM intercepte les accès effectués par les différents systèmes d'exploitation vers le MMU puis les interprète. Le VMM doit alors créer une page de table factice pour chacun des systèmes d'exploitation en cours d'exécution. Cette dernière doit faire correspondre les adresses virtuelles sollicitées par le système d'exploitation invité à celles qui lui ont été réservées par le VMM dans la mémoire physique. Ce tableau de correspondance intermédiaire est appelé *Shadow Page Table* (SPT).

Un simple accès à la mémoire peut dès lors devenir une opération complexe, le VMM étant dans l'obligation de gérer les accès mémoire de la VM, avec pour corollaire un ralentissement substantiel du VMM au niveau de ses propres accès, en particulier lors de périodes durant lesquelles les demandes d'accès à la mémoire sont fréquentes.

C'est pour cette raison qu'Intel® et AMD® ont inventé respectivement les **Extended Page Tables** (EPT) et **Rapid Virtualization Indexing** (RVI). Ces deux technologies ont pour but d'implanter le SPT au niveau matériel. Chaque système invité dispose donc en plus d'une table des pages et d'un MMU, d'un EPT ou RVI, ce qui a pour conséquence de leur permettre un accès direct à leur domaine d'adressage réservé, sans provoquer l'intervention du VMM. L'ensemble des adresses mémoire physiques des systèmes invités est recompilé dans l'EPT/RVI, la gestion de la mémoire du système client étant dès lors entièrement prise en charge par le matériel, avec pour corollaire une rapidité accrue. Une évaluation[30] effectuée par la société VMware® démontre par exemple que RVI offre un gain de performance allant jusqu'à 42% par rapport à l'usage d'une SPT.

AMD® a intégré cette technologie dans ses processeurs Opteron de troisième génération (Barcelona) et Intel® a fait de même à partir de l'architecture Nehalem. Il s'agit pour Intel® de la seconde génération de sa technologie prévue pour la virtualisation.

4.3.3.3 Virtualisation de l'accès aux périphériques d'entrées/sorties

Nous avons vu précédemment que l'accès au processeur est transparent dès lors que la machine physique est équipée d'un processeur dernière génération, AMD-V™ ou Intel® VT-x (cf. section 4.3.3.1, Virtualisation de l'accès au processeur).

L'accès aux périphériques d'entrées/sorties (E/S) reste, quant à lui, du ressort du VMM. Chaque hyperviseur doit en effet créer ses propres périphériques d'E/S virtuels. Il s'avère donc nécessaire de partager une unique ressource d'E/S entre plusieurs machines virtuelles.

[30] http://www.vmware.com/pdf/RVI_performance.pdf

Il s'agit en quelque sorte d'un partage de type logiciel (*software-based sharing*). Des techniques d'émulation sont utilisées pour fournir à la machine virtuelle des périphériques d'E/S logiques. La couche émulée s'interpose entre le système d'exploitation invité et le matériel physique sous-jacent, permettant au VMM d'intercepter tout le trafic généré par le pilote de la machine virtuelle.

La couche émulée peut notamment résoudre les multiples requêtes d'E/S provenant de toutes les machines virtuelles et les sérialiser (de l'anglais *serialization*) au sein d'un unique flux d'E/S qui peut être pris en charge par le matériel sous-jacent.

Les approches de type logiciel habituellement utilisées en matière de partage de périphériques sont l'**émulation** de ces derniers ou les **pilotes paravirtualisés**.

Dans le cas de l'émulation de périphériques par le VMM, les opérations d'E/S doivent en quelque sorte transiter par deux piles d'E/S différentes. L'une étant située au niveau de la machine virtuelle, l'autre au niveau du VMM.

Contrairement à la technique de l'émulation qui imite un périphérique existant et utilise le pilote correspondant déjà existant au sein du système d'exploitation invité, le modèle basé sur les pilotes paravirtualisés repose sur l'usage d'un pilote de type *front-end* au niveau de la machine invitée fonctionnant de concert avec un pilote *back-end* disponible au niveau de l'hyperviseur. Ces pilotes sont optimisés pour le partage et permettent d'éviter d'avoir à émuler un périphérique entier. Le pilote *back-end* se charge de la communication avec la couche physique. Nous avons précédemment évoqués ces différences de fonctionnement au sein des sections dévolues à la virtualisation totale et à la paravirtualisation.

La figure 4-19 illustre le partage logiciel de périphérique.

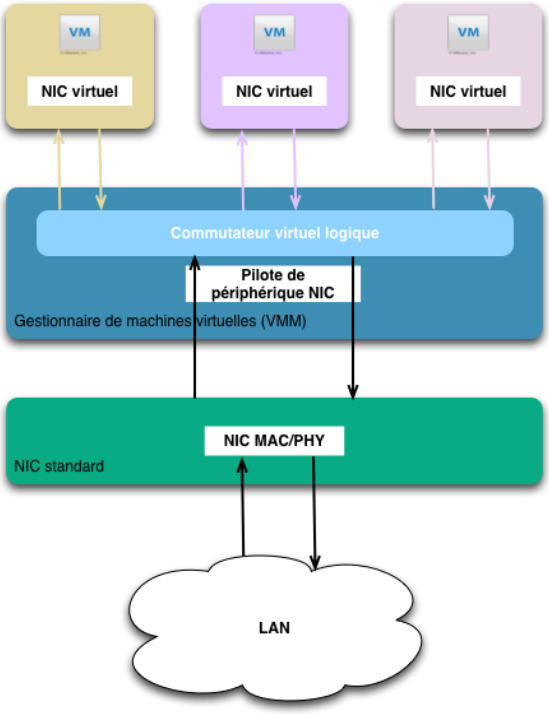


Figure 4-19 : Partage basé sur l'usage de logiciel (Source : PCI-SIG SR-IOV Primer, An Introduction to SR-IOV Technology de l'Intel® LAN Access Division)

L'émulation de périphériques ou les pilotes paravirtualisés ne permettent pas de bénéficier pleinement des fonctionnalités avancées offertes par les périphériques d'E/S. De plus, comme nous l'avons évoqué à la section 4.3.1, Virtualisation logicielle, un usage important de ressources processeur supplémentaires (*overhead*) est provoqué par la nécessité pour le VMM d'implémenter un périphérique au niveau logiciel. La conséquence est généralement la réduction du débit maximum d'E/S possible sur ce dernier (donc l'inverse de la situation souhaitable).

À ce propos, Intel® a démontré, par le biais de tests intensifs, qu'en utilisant uniquement l'émulation matérielle, un contrôleur Ethernet 10 Gbit/s ne parvenait à atteindre qu'un débit maximum de 4.5 à environ 6.5 Gbit/s (la variation étant due aux différentes architectures de serveur sur lesquelles les tests ont été menés).

Des pilotes adaptés aux périphériques virtuels doivent être de surcroît développés. Les performances liées à ce type d'accès sont loin d'être une panacée (en particulier pour les cartes graphiques). De plus, le VMM doit reproduire la gestion de la mémoire des périphériques eux-mêmes, ce qui s'avère parfois insuffisant. En effet, si un programme s'exécutant sur la machine invitée tente d'accéder directement au matériel, il risque tout simplement de ne pas fonctionner.

La virtualisation des E/S a pour but de rendre les systèmes hôtes suffisamment évolutifs pour permettre la prise en charge d'un nombre croissant d'invités. Il est dès lors impératif de tirer parti des ressources inutilisées autant que faire se peut. Obtenir des performances quasi natives au niveau des opérations d'E/S devient donc un impératif.

Une solution de virtualisation des E/S doit donc faire en sorte que les machines invitées soient isolées, au même titre qu'elles le sont déjà lorsqu'un environnement fonctionne sur une machine physique séparée. Nous l'avons vu précédemment, cette isolation est nécessaire au niveau de l'accès à la mémoire, en séparant l'espace totale de cette dernière entre les différentes machines virtuelles. Pour les mêmes raisons, il est impératif de séparer les flux d'E/S, les interruptions y relatives et – dans le cas du partage de périphériques – d'isoler les opérations de contrôle, les opérations d'E/S et les erreurs.

Le partage de ressources matérielles exige parfois que nous passions outre la couche de virtualisation en permettant à la machine virtuelle d'accéder directement aux périphériques concernés. D'autres mécanismes sont donc nécessaires pour renforcer le principe d'isolation précédemment évoqué.

Tout comme dans le cas de la virtualisation de l'accès à la mémoire, les constructeurs ont proposé une nouvelle technologie à cette attention, connue sous l'appellation d'**IOMMU**. Les processeurs embarquant cette dernière sont connus sous les noms de **VT-d** pour Intel® ou de **Vi** pour AMD®.

En informatique, l'unité de gestion de la mémoire des périphériques d'entrées/sorties (IOMMU) est semblable à une unité de gestion de la mémoire (MMU) qui relie un bus d'entrées/sorties compatible DMA (cf. section 4.1.1.8, Périphérique d'entrées/sorties) à la mémoire principale. Comme un MMU traditionnel, qui traduit les adresses virtuelles en adresses physiques, l'IOMMU se charge du « mappage » des adresses virtuelles

correspondant aux périphériques d'E/S aux adresses physiques correspondantes La traduction mémoire est donc améliorée et une protection de la mémoire est fournie, donnant la possibilité à un périphérique d'accéder directement la mémoire de l'hôte (DMA). Il est donc possible de passer outre la couche E/S émulée du VMM avec pour corollaire une amélioration conséquente des performances des machines virtuelles.

Une fonctionnalité de la technologie VT-x d'Intel® permet à une VM d'accéder directement à une adresse physique, ce qui autorise le pilote d'un périphérique situé au niveau de la machine virtuelle de pouvoir écrire directement dans les registres du périphérique d'E/S. Intel® VT-d fournit une capacité similaire destinée aux périphériques d'E/S leur permettant d'écrire directement dans l'espace mémoire d'une machine virtuelle (par exemple une opération DMA). Ce processus est illustré par la figure 4-20, ci-dessous.

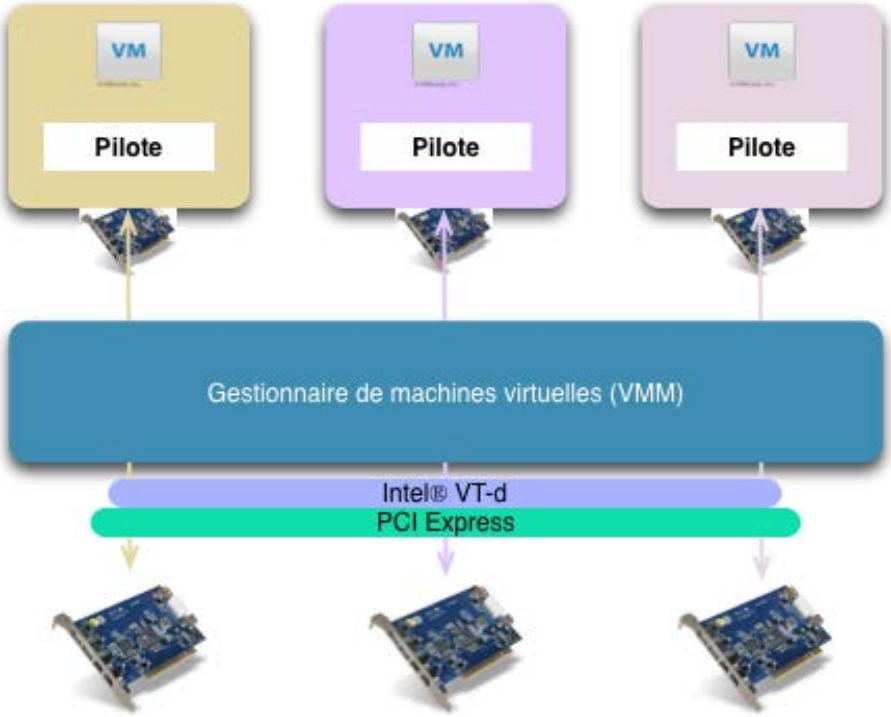


Figure 4-20 : Affectation direct (Source : PCI-SIG SR-IOV Primer, an Introduction to SR-IOV Technology de Intel® LAN Access Division)

Certaines unités offrent également une protection de la mémoire contre des périphériques au comportement douteux.

La figure 4-21 compare le fonctionnement de l'unité de gestion de la mémoire des périphériques d'E/S avec le fonctionnement de l'unité de gestion de la mémoire (cf. section 4.1.1.2.4, MMU).

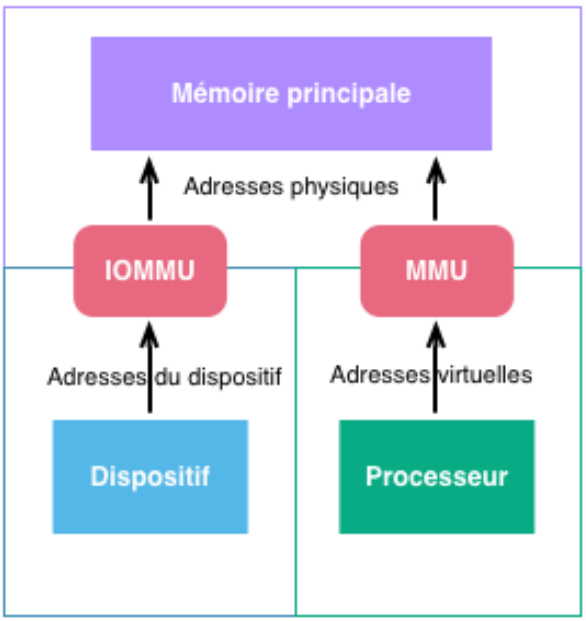


Figure 4-21 : Comparaison de l'unité de gestion de la mémoire des entrées/sorties (IOMMU) et de l'unité de gestion de la mémoire (MMU) (Source : en.wikipedia.org/wiki/IOMMU)

Les technologies d'Intel® et d'AMD® permettent donc de créer un accès virtuel aux périphériques d'entrées/sorties.

Contrairement à VT-x et AMD-V, qui dépendent du processeur, ces deux technologies dépendent du chipset (ou tout de même du processeur dans le cas des modèles d'Intel® intégrant le contrôleur PCI-Express).

Cette technologie présente cependant un désavantage important. En effet, en lieu et place de partager un périphérique entre un système hôte et un système client, un accès direct au périphérique est fourni à la machine invitée. Il s'agit donc bien d'un accès direct et exclusif et non d'un partage en tant que tel. Par exclusif, nous entendons que pendant le laps de temps durant lequel la machine invitée bénéficie d'un accès direct au périphérique concerné, la machine hôte perd le sien. En d'autres termes, un périphérique physique ne peut être assigné qu'à une seule VM à la fois.

Il est bien évidemment possible de dédier un périphérique physique à une machine virtuelle en particulier (par exemple, une carte graphique).

L'industrie a cependant pris conscience de cette problématique et a développé de nouveaux périphériques capables d'être nativement partagés. Ces derniers reproduisent les ressources nécessaires à chacune des VM pour que ces dernières puissent être connectées au périphérique d'E/S, le flux de données principal pouvant s'effectuer sans l'aide du VMM. Nous évitons donc les tempêtes d'interruption matérielle (IRQ) générées par les machines virtuelles lorsqu'elles sont nombreuses sur le même hôte physique.

La spécification qui permet à ce type de périphériques de pouvoir procéder de la sorte est connue sous l'appellation SR-IOV[31].

SR-IOV permet à un périphérique compatible PCIe de créer plusieurs instances de lui-même. Le BIOS sur système hôte, ainsi que l'hyperviseur doivent pouvoir supporter cette spécification.

L'idée de **fonctions physiques** (PFs pour *Physical Functions* en anglais) et **fonctions virtuelles** (VFs pour *Virtual Functions* en anglais) est introduite par cette spécification, au niveau du périphérique.

Les fonctions physiques regroupent les fonctions complète du périphérique (PCIe), incluant les capacités étendues SR-IOV. Ces capacités sont utilisées pour configurer et gérer la fonctionnalité SR-IOV.

Les fonctions virtuelles contiennent les ressources nécessaires à la gestion des flux de données (entrant et sortant) mais ne possèdent qu'un jeu limité de ressources liées à la configuration. Il s'agit donc de fonctions allégées.

Une fonction racine unique (*Single Root Function*) – par exemple, un port Ethernet unique – peut apparaitre comme étant plusieurs périphériques physiques.

Un dispositif compatible SR-IOV (habituellement un hyperviseur) peut fournir un certain nombre de VFs indépendantes et configurables, chacune d'entre elles possédant son propre espace de configuration PCI, et en assigner une ou plusieurs à une VM. Les technologies de traduction mémoire, telles que celles embarquées dans VT-x ou VT-d, autorisent l'usage des techniques d'assistances matérielles qui permettent les transferts directs de type DMA vers et de la VM, contournant ainsi le traitement logiciel au sein du VMM.

La figure 4-22 illustre l'association des fonctions virtuelles avec leur espace de configuration respectif.

[31] *Single Root I/O Virtualization and Sharing*, s'agissant d'une spécification mise en place par le PCI-SIG (*Peripheral Component Interconnect Special Interest Group).* Ce dernier est un consortium de l'industrie de l'électronique, responsable des spécifications des bus PCI, PCI-X, et PCIe.

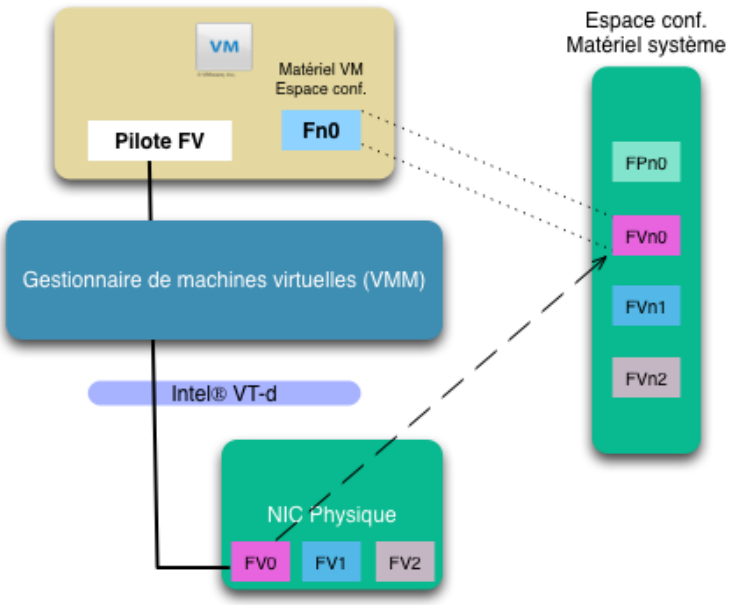


Figure 4-22 : Association des VF avec leur espace de configuration (Source : PCI-SIG SR-IOV Primer, An Introduction to SR-IOV Technology de Intel® LAN Access Division)

Nous faisions également référence à la technologie **VT-c** d'Intel® dans la section liée à la virtualisation de l'accès au processeur. L'acronyme VT-c signifie Intel® Ethernet Virtualization Technology for Connectivity.

Cette assistance matérielle vise à :

- Améliorer la vitesse à laquelle les données transitent au sein d'une plateforme multi-cœur architecturée Intel® ;
- Améliorer les performances de traitement des données au travers des multiples files d'attente au niveau du contrôleur de réseau ;
- Fournir une connectivité directe aux VM à la NIC, ainsi qu'une protection des données entre les VM.

4.4 Mise en œuvre

4.4.1 Virtualisation des systèmes d'exploitation

4.4.1.1 Hyperviseur

L'hyperviseur, également appelé VMM, est, à proprement parlé, une **plateforme de virtualisation**. Quoique le terme hyperviseur qualifie en soit l'élément générique que constitue la plateforme en question, il est souvent utilisé par abus pour qualifier une technique de virtualisation en soi, s'agissant en particulier de la paravirtualisation (même si cette technique illustre parfaitement le rôle de l'hyperviseur).

L'hyperviseur, hormis le fait de présenter aux machines virtuelles invitées la plateforme évoquée ci-dessus, gère leur exécution. Il permet en particulier le partage des ressources matérielles entre les différentes instances de systèmes qu'il héberge. L'hyperviseur est généralement installé sur le matériel même avec pour fonction d'exécuter les machines

virtuelles invitées qui, elles-mêmes, opèrent en tant que serveurs ou en tant que stations de travail.

Le terme hyperviseur trouve son origine dans le fait qu'historiquement, au sein de l'architecture x86, le noyau du système d'exploitation est le **superviseur** (cf. section 4.1.1.7, Niveaux de privilèges) qui, à ce titre, gère les interruptions. C'est donc en toute logique qu'un hyperviseur gère les interruptions en environnement virtuel.

Nous avons vu que dans le cas de la virtualisation de systèmes d'exploitation non-modifiés (contrairement à ce qui se passe dans un contexte de paravirtualisation), la couche de virtualisation se situe au-dessus d'un système d'exploitation hôte et donc du noyau de ce dernier. Nous pouvons donc considérer que le superviseur (noyau) va gérer l'hyperviseur, ce qui constitue tout de même une contradiction. Les notions d'hyperviseur de type 1 ou de type 2 ont donc été créées pour différencier l'hyperviseur lié à un contexte de virtualisation totale de celui lié à un contexte de paravirtualisation.

4.4.1.1.1 Type 1 ou natif

Ce type d'hyperviseur, également qualifié de ***bare metal*** (littéralement métal nu), s'**exécute directement sur l'environnement matériel**, comme illustré par la figure 4-23. Il s'agit en fait d'un noyau allégé et optimisé pour héberger les machines virtuelles invitées. Il s'inscrit historiquement dans un contexte de paravirtualisation même si, nous l'avons vu précédemment (cf. section 4.3.1.2, Paravirtualisation), l'assistance matérielle procurée par les processeurs AMD-V™ et Intel® VT-x permet de virtualiser un système d'exploitation dont le noyau n'a pas été modifié (cf. section 4.3.3.1, Virtualisation de l'accès au processeur). La machine invitée a donc parfaitement « conscience » d'être virtuelle.

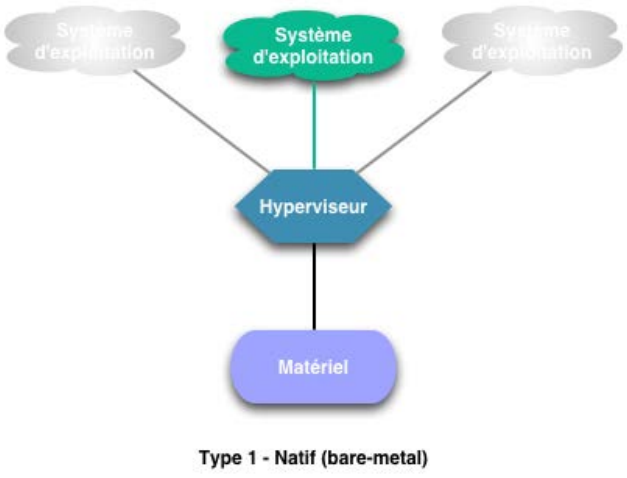


Figure 4-23 : Hyperviseur bare metal (Source : fr.wikipedia.org/wiki/Hyperviseur)

Les hyperviseurs de ce type occupant une bonne place sur le marché sont :

- Citrix **XenServer**™[32] ;
- VMware **ESX / ESX(i)**™[33] ;

[32] http://www.citrix.com/English/ps2/products/product.asp?contentID=683148.
[33] http://www.vmware.com/products/vsphere/esxi-and-esx/index.html.

- Microsoft **Hyper-V**™[34] ;
- Red Hat **Entreprise Virtualization**™[35]**.**

4.4.1.1.2 Type 2 (ou virtualisation hébergée)

L'hyperviseur de type 2 porte relativement mal son nom puisqu'il ne s'agit pas véritablement d'un hyperviseur en tant que tel. En effet, nous ferons plus volontiers référence ici à une couche de virtualisation ou, pour être précis, à un **logiciel s'exécutant à l'intérieur d'un système d'exploitation**, la machine virtuelle invitée s'exécutant à un troisième niveau au-dessus du matériel (cf. figure 4-24). Comme nous l'avons vu précédemment dans le cadre de la virtualisation totale (cf. section 4.3.1.1, Virtualisation totale (traduction binaire)), la machine invitée n'a pas « conscience » d'avoir été virtualisée.

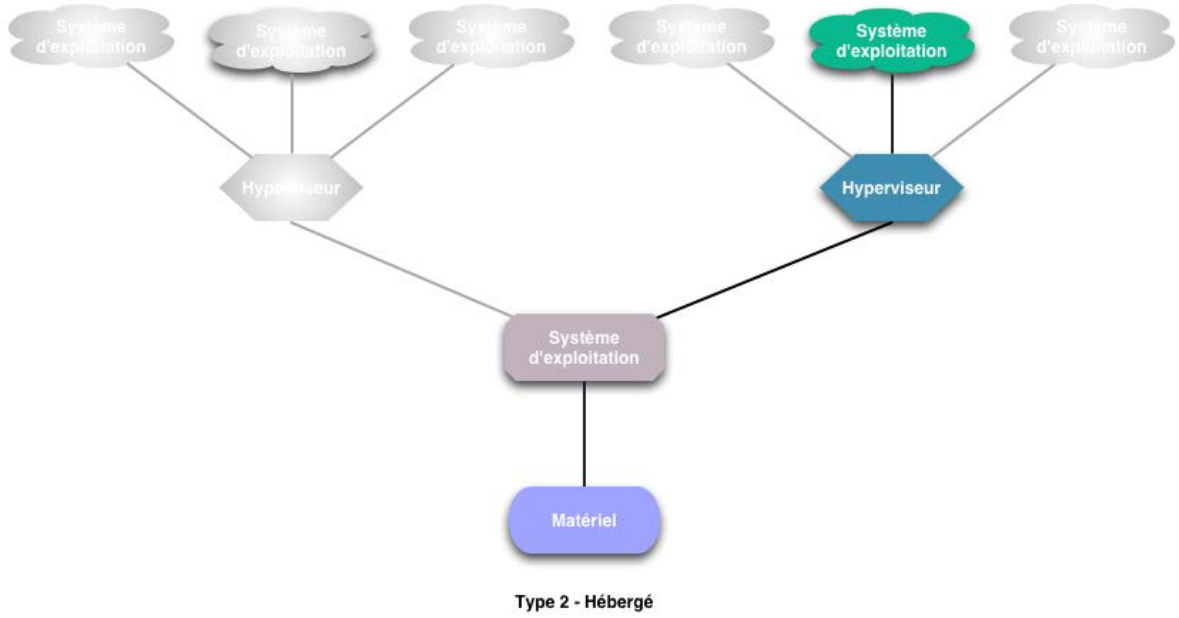


Figure 4-24 : Hyperviseur de type 2 (Source : fr.wikipedia.org/wiki/Hyperviseur)

Les hyperviseurs de ce type, occupant une place de choix sur le marché, sont :

- **VMware Workstation**™[36] ;
- Oracle **VirtualBox**™[37] ;
- **Parallels Desktop**™[38] ;
- Microsoft **Virtual PC**™.

4.4.1.2 Émulateur

Les meilleurs exemples d'émulateurs logiciels que nous pouvons citer sont **Bochs**™[39] et **QEMU**™[40].

[34] http://www.microsoft.com/en-us/server-cloud/hyper-v-server/default.aspx.
[35] http://www.redhat.com/products/virtualization/.
[36] http://www.vmware.com/fr/products/desktop_virtualization/workstation/overview.html.
[37] https://www.virtualbox.org/.
[38] http://www.parallels.com/fr/computing/.
[39] http://bochs.sourceforge.net/.
[40] http://wiki.qemu.org/Main_Page

Bochs™

Cet émulateur, écrit en C++, est libre et gratuit. Il fonctionne sur les environnements UNIX®, Linux™, Windows® et Mac OS/X™ mais prend en charge uniquement les architecture x86.

Ce logiciel très sophistiqué prend en charge un vaste éventail de matériel, lui permettant d'émuler toutes les architecture IA-32 et Intel 64. Il prend également en charge des processeurs multi-cœurs mais ne tire pas complètement avantage du SMP à ce jour.

QEMU™

Cet émulateur est également libre et gratuit et fonctionne sur un nombre limité d'architectures, s'agissant d'IA-32, d'Intel 64 et de PPC[41]. Il est capable d'émuler des systèmes IA-32, Intel 64, ARM, SPARC, PowerPC, MIPS et m68k.

Nous précisons que les solutions de Microsoft® que sont **Virtual PC™**[42] et **Virtual Server™**[43] utilisent l'émulation pour fournir l'environnement nécessaire aux machines virtuelles. Il ne s'agit pas de solutions sur lesquelles nous pourrions envisager de baser une infrastructure de production, ces dernières ayant été conçues pour héberger gratuitement quelques machines.

Il est à noter que les performances des machines virtuelles équipées du système d'exploitation Windows® sont excellentes, à tel point qu'il est difficile de savoir que nous utilisons une machine virtuelle.

4.4.1.3 Niveau noyau

Nous évoquerons au sein de cette section, deux solutions bien connues mettant en œuvre la virtualisation au niveau noyau. Ces solutions sont **KVM™** [44]et **User-mode Linux™**[45].

KVM™

KVM™ pour Kernel Virtual Machine est un QEMU™ modifié. À la différence de ce dernier, KVM™ utilise les technologies AMD-V™ et Intel® VT (cf. section 4.3.3.1, Virtualisation de l'accès au processeur). Le **noyau Linux™** est utilisé comme hyperviseur et fonctionne comme un module que l'on peut charger dans le noyau.

Cette solution permet de virtualiser de nombreux systèmes d'exploitation invités architecturés IA-32 ou Intel 64, comme Windows®, Linux™ et FreeBSD®.

Le composant noyau de KVM™ est inclus dans la ligne principale de Linux™, depuis la version 2.6.20.

User-Mode Linux™

UML ou Linux™ en mode utilisateur se sert d'un noyau exécutable et d'un système de fichiers racine pour créer une machine virtuelle. La console par le biais de laquelle nous

[41] PowerPC.
[42] http://www.microsoft.com/windows/virtual-pc/.
[43] http://www.microsoft.com/windowsserversystem/virtualserver/.
[44] http://www.linux-kvm.org/page/Main_Page.
[45] http://user-mode-linux.sourceforge.net/.

accédons à la machine virtuelle est simplement une session au sein d'un terminal en ligne de commande.

Utiliser UML peut est très pratique lorsqu'il s'agit d'héberger des serveurs virtuels, mais surtout :

- Pour le développement de noyau ;
- Pour effectuer des expérimentations avec un nouveau noyau ou de nouvelles distributions ;
- Comme bac à sable[46].

Procéder de la sorte permet en effet de préserver son environnement Linux™ principal.

UML est inclus dans tous les noyaux 2.6.x.

4.4.2 Virtualisation des processus (à noyau partagé)

Cette technique tire avantage de la possibilité unique, existant au sein des systèmes d'exploitation UNIX® ou Linux™, de partager le noyau avec d'autres processus du système.

La mise œuvre de cette méthode est basée sur la fonctionnalité **chroot**, pour *change root*, ou modifier la racine. Le système de fichiers racine d'un processus est donc modifié par cette fonctionnalité afin qu'il soit isolé, avec pour but de fournir un espace utilisateur bénéficiant d'une certaine sécurité. Ces espaces, au sein desquels les systèmes de fichiers racine sont isolés, sont qualifiés de **prison chroot** ou de **conteneur**.

Un programme, un ensemble de programmes ou un système d'exploitation – dans le cas de la virtualisation à noyau partagé – fonctionnant dans un environnement chroot est protégé, le système emprisonné étant « persuadé » de fonctionner sur une machine réelle avec à sa disposition un système de fichiers.

Au registre des avantages, nous pouvons mentionner le fait que les performances obtenues peuvent être qualifiées de natives. En effet, les systèmes ainsi constitués partagent le même noyau, soit celui du système d'exploitation hôte. Aucune couche de virtualisation supplémentaire n'est dès lors nécessaire pour assurer la communication avec le matériel, assurant du même coup des performances natives (celles du noyau et du matériel sous-jacent). Il n'en va pas ainsi lorsqu'un hyperviseur est présent, puisqu'il doit gérer lui-même les accès au matériel. Il est également utile de souligner que l'isolation de ces systèmes est accrue, améliorant ainsi la sécurité. Nous soulignons de plus qu'une densité plus élevée de systèmes virtualisés sur un même système hôte peut ainsi être rendu possible, si nous considérons la mémoire comme facteur limitatif. Cette méthode est en effet plus proche d'un système faisant fonctionner plusieurs applications qu'un hyperviseur faisant fonctionner plusieurs machines invitées.

Quant au désavantage majeur de cette technique, il réside dans le fait qu'il soit bien évidemment impossible de virtualiser des systèmes d'exploitation dont le noyau est différent

[46] De l'anglais *Sandbox*, qui consiste en un environnement de test pour exécuter des logiciels en cours de développement ou de provenance douteuse.

de celui du système hôte. Différentes distributions[47] Linux™ pourront par exemple être virtualisées si le système hôte est basé sur un environnement Linux™. Il ne sera par contre pas possible d'héberger un système d'exploitation Windows® sur le même environnement.

Les hébergeurs web utilisent cette méthode depuis un certain temps pour proposer à leurs clients les espaces mutualisés dont ils ont besoin. Ces derniers ne sont pas conscients d'accéder à une machine virtuelle et ne peuvent en aucun cas accéder au système hôte depuis la machine virtuelle dont ils disposent.

Les **zones Solaris™ (containers)**[48] et **OpenVZ™**[49] peuvent être considérés comme des exemples concrets de virtualisation à noyau partagé.

Solaris Containers™

Cette solution est intégrée au système d'exploitation Solaris™ d'Oracle®. Les zones en question sont en fait des prisons (*jails*) BSD (*Berkeley Software Distribution*). Chacune d'entre elles contient sa racine virtuelle propre qui imite un système d'exploitation et un système de fichiers complet.

Chaque zone ne voit que ses propres processus et systèmes de fichiers. La zone « croit » qu'elle est un système d'exploitation complet et indépendant. Seule la zone globale, qui peut être considérée comme le VMM, à « conscience » de la virtualisation mise en œuvre.

Tout comme dans le cas d'UML (cf. section 4.4.1.3, Niveau noyau), les zones Solaris™ peuvent faire office de bac à sable. Cette solution est très professionnelle, facile d'utilisation et offre des performances natives.

OpenVZ™

Le noyau OpenVZ™ est optimisé pour la virtualisation et son fonctionnement est très proche de celui des zones Solaris™, à ceci près que nous pouvons faire fonctionner plusieurs distributions Linux™ à l'aide du même noyau.

Les zones ainsi crées sont connues sous les appellations de VE ou VPS. Elles sont isolées les unes des autres et parfaitement sécurisées. Chaque conteneur peut être considéré comme un serveur propre. Il peut être redémarré indépendamment des autres conteneurs, bénéficier d'un accès *root*, d'utilisateurs, d'une adresse IP, de mémoire, de processus, de fichiers, d'applications, d'une librairie système et de fichiers de configuration, au même titre que n'importe quel environnement.

Nous précisons que l'hyperviseur **Parallels Server Bare Metal™**[50] de la société éponyme est basé sur OpenVZ™.

[47] Ici, nous entendons par distribution un système d'exploitation basé sur le noyau Linux™, incluant un large panel de logiciels intégrés. Les distributions les plus utilisées sont basées sur Debian, comme la distribution éponyme, ainsi qu'Ubuntu™ ou Mint™.
[48] http://www.oracle.com/technetwork/server-storage/solaris/containers-169727.html.
[49] http://wiki.openvz.org/Main_Page.
[50] http://www.parallels.com/products/server/baremetal/sp/.

5 Domaines d'application

Au cours de ce chapitre, nous évoquerons les différents domaines au sein desquels la virtualisation peut être mise en pratique. Il s'agit en quelque sorte du prolongement de la section 4.4, relative à la mise en œuvre des techniques de virtualisation, à ceci près que cette dernière faisait référence à la virtualisation des serveurs uniquement.

Ce chapitre vise donc à offrir au lecteur une vue d'ensemble des domaines d'une infrastructure physique qui contiennent des périphériques susceptibles d'être virtualisés. Il n'a cependant pas vocation à apporter des détails techniques fondamentaux mais plutôt à mettre en évidence les concepts et avantages y relatifs.

Nous verrons également que les méthodes d'abstraction de la couche physique peuvent différer en fonction des domaines concernés. En effet, si virtualiser un serveur ou virtualiser une station de travail revient pratiquement à mettre en œuvre les mêmes principes de virtualisation, virtualiser les applications, le stockage ou le réseau fait appel à d'autres concepts technologiques.

Nous passerons rapidement sur la virtualisation des **serveurs** dont nous avons abondamment parlé dans les chapitres précédents pour nous concentrer sur la virtualisation des **stations de travail** (cf. section 5.2, Stations de travail) et des **applications** (cf. section 5.3, Applications). Ainsi, nous sortirons quelque peu du centre de données pour nous rapprocher des utilisateurs. Puis nous regagnerons ensuite le *datacenter* pour évoquer la virtualisation du **stockage** (cf. section 5.4, Stockage) et du **réseau** (section 5.5, Réseau).

5.1 Serveurs

Les serveurs ont été les premiers dispositifs à avoir été virtualisés, et ce, pour les raisons que nous avons précédemment évoquées, tant au chapitre 1, Introduction, portant sur l'historique de la virtualisation, qu'au chapitre 3, Bénéfices de la virtualisation, évoquant les bénéfices apportés par cette technologie. Nous avons également évoqué longuement la virtualisation des serveurs au chapitre 4, Fondamentaux technologiques de la virtualisation, portant sur les fondamentaux technologiques de la virtualisation.

Aussi, nous n'approfondiront pas le sujet dans la présente section, cette dernière étant simplement destinée à rappeler au lecteur que la virtualisation des serveurs demeure un domaine d'application clé. En effet, il s'agit généralement de la première étape du processus menant à la virtualisation complète du centre de données, puis du reste du système d'information.

Nous rappelons toutefois brièvement que la virtualisation des serveurs consiste à consolider plusieurs serveurs virtuels (qu'il s'agisse de serveurs *mail*, de serveurs d'applications, de serveurs de fichiers, de l'Active Directory, etc.), généralement dédiés, sur le même serveur physique, ce dernier faisant dès lors office d'hyperviseur (cf. section 4.4.1.1, Hyperviseur). Les principes fondamentaux de cette technologie résident dans le fait de faire abstraction du matériel sous-jacent en le rendant virtuel, de telle manière à pouvoir y installer un système d'exploitation. Virtualiser un serveur ou une station de travail revient donc pratiquement au même. La virtualisation de serveur ou de station de travail est d'ailleurs régulièrement

qualifiée de virtualisation de système d'exploitation, ce qui, de notre point de vue, n'est pas suffisamment précis. En effet, comme nous l'avons vu plus haut, installer un système d'exploitation sur une machine virtuelle, exige au préalable qu'un matériel virtuel soit fourni par l'hyperviseur. En conséquence, nous virtualisons aussi bien la machine sous-jacente que le système d'exploitation qui y sera installé.

Les principales solutions de virtualisation serveurs sont :

- **VMware vSphere**[51] (**ESX** ou **ESX(i)**)[TM] ;
- Microsoft **Hyper-V**[TM52] ;
- Citrix **XenServer**[TM53] ;
- Oracle **VM Server**[TM54] (pour X86 ou SPARC et uniquement pour la virtualisation des bases Oracle®) ;
- Red Hat **Entreprise Virtualization**[TM55].

5.2 Stations de travail

5.2.1 Concept

La virtualisation des postes de travail est l'évolution logique de la virtualisation des serveurs. Le poste de travail se résume à une machine virtuelle disponible sur un serveur localisé au sein du centre de données. Les utilisateurs ne font que de se connecter à cette machine virtuelle, généralement via des terminaux légers.

La figure 5-1 illustre le concept décrit ci-dessus. Nous pouvons distinguer, au centre, un poste de travail virtuel situé au sein du centre de données. L'utilisateur peut accéder à cette station par le biais de tout type de client. La station de travail contient, quant à elle, tant les applications que les données, mais également les paramétrages propres à l'utilisateur.

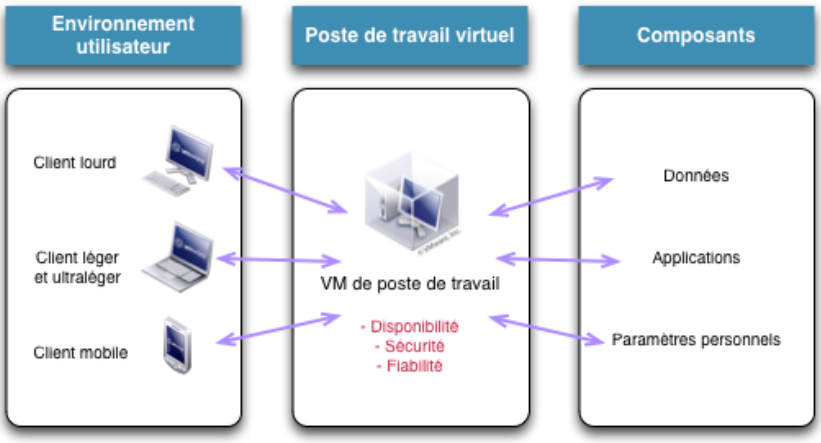


Figure 5-1 : Interactions avec un poste de travail virtuel (Source : VMware View[TM], fiche produit)

[51] http://www.vmware.com/products/vsphere-hypervisor/overview.html.
[52] http://www.microsoft.com/en-us/server-cloud/hyper-v-server/default.aspx.
[53] http://www.citrix.com/English/ps2/products/product.asp?contentID=683148.
[54] http://www.oracle.com/us/technologies/virtualization/overview/index.html?origref=http://www.google.ch/url?sa=t&rct=j&q=oracle%20vm&source=web&cd=2&ved=0CDIQFjAB&url=http%3A%2F%2Fwww.oracle.com%2Fus%2Ftechnologies%2Fvirtualization%2F&ei=2mRCUK_zM8fP4QS9sYD4Dw&usg=AFQjCNFG8ZPtny4c-Zu9MANF3cYdusrzJg.
[55] http://www.redhat.com/products/virtualization/.

Ce type de virtualisation répond à un certain nombre de défis auxquels doivent répondre les services informatiques :

- Les employés mobiles ont besoin d'un accès complet aux ressources informatiques (applications ou données), quel que soit le lieu où ils se trouvent ;
- Le nombre de périphériques à gérer explose. Il existe, de plus, une variété toujours plus importante de dispositifs. Nous sommes par exemple confronté à un nombre croissant de demandes provenant des employés, relatives à une connexion de tablettes au réseau local de l'entreprise ;
- les tâches en relation avec la gestion des ordinateurs de bureau doivent être simplifiées et rationalisées, afin d'éviter que les administrateurs soient systématiquement détournés des tâches stratégiques, d'autant que les besoins des utilisateurs se diversifient. Les coûts relatifs à certaines tâches clés, comme la configuration des stations de travail ou le déploiement de mise à jour, de correctifs ou de mise à niveau, doivent être réduits ;
- Les employés attendent de leur employeur qu'il favorise l'usage de dispositifs personnels, tels que *smartphones*, *laptop* ou tablettes. Cette tendance, de plus en plus patente au sein des sociétés, est qualifiée de « consumérisation » ;
- L'accroissement de la mobilité des employés, l'explosion du nombre de périphériques et la « consumérisation » augmentent les risques liés à la sécurité.

5.2.2 Avantages

5.2.2.1 Amélioration de la gestion

Tout comme pour la virtualisation des serveurs, la virtualisation des postes de travail permet de déplacer le système d'exploitation, les applications et les paramètres de l'utilisateur des stations clientes vers le centre de données. Cette centralisation de la gestion des postes client évite donc aux administrateurs d'avoir à se déplacer sur chaque poste pour résoudre différents problèmes logiciels. Il en va de même pour les problèmes matériels, puisque un terminal est moins complexe qu'un ordinateur et, de ce fait, moins sujet aux pannes.

La figure 5-2 illustre un client léger (*zero client*) Dell Wyse™ P20 conçu pour la solution VMware View™.

Figure 5-2 : Client zéro Dell Wyse™ P20 (Source : http://www.wyse.com/products/cloud-clients/zero-clients/P20)

De plus, vol et vandalisme sont évités ou les pertes financières y relatives limitées, le terminal ne servant à rien lorsqu'il est utilisé seul et n'étant pas très cher.

Les départements informatiques n'ont pas à se soucier d'avoir à gérer un grand nombre d'images système provenant de configurations matérielles différentes, puisque l'environnement de bureau est centralisé et peut être déployé sur différents types de périphériques, indépendamment du matériel composant ce dernier.

5.2.2.2 Diminution des coûts

En consolidant un certain nombre de postes de travail virtuels sur une plateforme centralisée, les infrastructures SHVD (*Server Hosted Virtual Desktop*) constituent un formidable levier d'économie pour les directions des systèmes d'information. D'après les principales études menées par Entreprise Management Associates® (EMA™), 71% des entreprises ayant virtualisé leurs postes de travail réalisent aujourd'hui des économies tangibles de l'ordre de 60% sur leurs coûts matériels, logiciels, ainsi que sur ceux relatifs à l'administration.

5.2.2.3 Amélioration de la productivité

Si un dispositif est en panne, l'employé peut accéder à sa station de travail virtuelle depuis un autre périphérique, et ce, sans avoir à restaurer données ou configuration.

La productivité des utilisateurs est améliorée par le fait que ces derniers peuvent accéder en tout temps et depuis n'importe où à leur environnement de bureau, à partir de n'importe quel type de périphériques (*smartphones*, tablettes, ordinateurs portables).

5.2.2.4 Renforcement de la sécurité

La sécurité des données stratégiques est garantie par le fait que ces dernières ne quittent pas le centre de données. Puisque les employés accèdent à leur environnement de travail depuis n'importe quel type de périphérique, ils ne ressentent plus le besoin de procéder à des copies locales de leurs fichiers, voire de les sauvegarder sur clé USB ou sur support optique pour les transférer d'un ordinateur à un autre.

De surcroit, conformément à la manière de faire lorsque nous travaillons avec des serveurs virtuels, la sauvegarde des stations de travail virtuelles est centralisée et simplifiée (cf. section 3.1.8, Optimisation de la sauvegarde).

Des stratégies peuvent être définies par les administrateurs pour fournir un niveau de sécurité supplémentaire. L'accès à une station peut être lié à un emplacement ou un type de réseau. L'accès à certaines données stratégiques peut par exemple être limité en fonction du lieu à partir duquel l'employé se connecte. Lorsqu'un employé quitte l'entreprise, ses privilèges d'accès peuvent être immédiatement supprimés, évitant ainsi l'exposition des données.

Ces différents avantages contribuent à relever les défis mentionnés au début de la présente section.

5.2.3 Solutions

Les principales solutions de virtualisation des stations de travail sont :

- VMware **View**[TM56] ;
- Citrix **XenDestop**[TM57] ;
- NEC **Virtual PC Center**[TM58] ;
- Quest **Desktop Virtualization**[TM59] ;
- Systancia **AppliDis Fusion**[TM60] ;
- Neocoretech **Desktop Virtualization**[TM61] (NDV®).

5.3 Applications

5.3.1 Concept

La virtualisation d'applications consiste à créer un environnement virtuel propre à l'application, comprenant ses propres clés de registre au sein d'une base de registre propre**,** son propre système de fichiers, ses propres *.dll, voire des bibliothèques tierces ou des *frameworks*. Cet environnement virtuel est créé à l'aide d'une solution logicielle d'encapsulation prévue à cet effet. L'application virtuelle fonctionne donc dans une « bulle » qui l'isole du système d'exploitation et des autres applications en cours d'exécution. Les conflits de *.dll sont évités et il devient parfaitement possible d'exécuter, par exemple, Internet Explorer 8 et 9 simultanément, sans qu'ils ne se télescopent, la couche virtuelle se chargeant d'intercepter les appels à la base de registre ou au système d'exploitation.

Les clés de registre et le système de fichiers ne constituent en rien des copies de ceux du système d'exploitation sur lequel l'application est exécutée. Ils contiennent simplement les modifications effectuées par l'application pour pouvoir fonctionner.

L'application n'aura donc accès qu'à ses propres versions de *.dll et fichiers de configuration système. Sur le même principe, elle ne pourra agir que sur sa propre base de registre. Aucun conflit avec d'autres applications n'est donc à craindre. D'où le concept de **bulle applicative étanche**.

Nous pouvons observer, sur la figure 5-3, la couche d'intégration chargée de l'interception des appels à la base de registre et aux applications (1) et les fichiers système propres à chaque application (2). Les applications demeurent indépendantes les unes des autres puisqu'encapsulées dans leur bulle respective (4).

[56] http://www.vmware.com/fr/products/desktop_virtualization/view/overview.html.
[57] http://www.citrix.fr/virtualization/desktop/xendesktop.html.
[58] http://www.nec.com/en/global/solutions/vpcc/.
[59] http://www.quest.com/desktop-virtualization/.
[60] http://www.systancia.com/fr/AppliDis-Fusion-4.
[61] http://www.neocoretech.com/produits/ndv-neocoretech-desktop-virtualisation/.

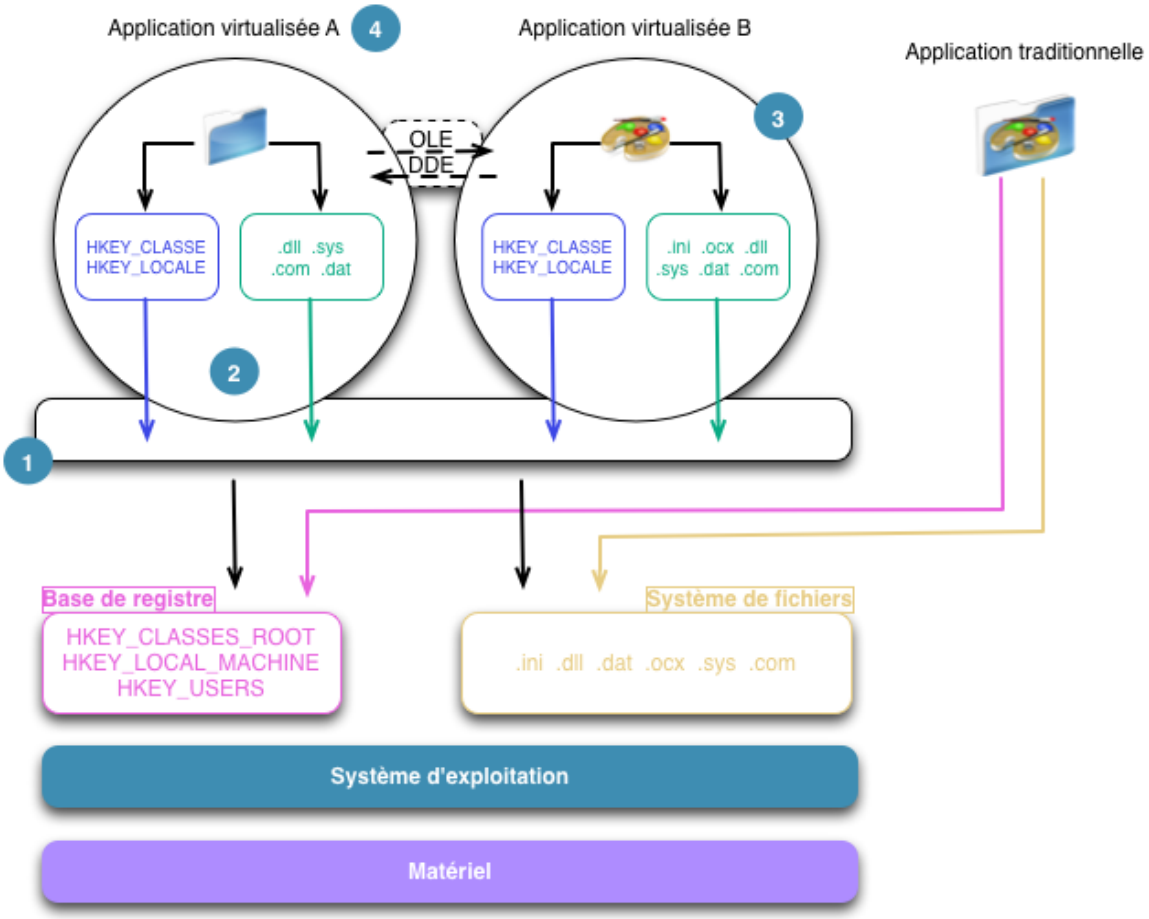


Figure 5-3 : Principe de virtualisation d'applications (Source : http://pro.01net.com/editorial/324015/la-virtualisation-dapplications)

Une application n'est donc plus installée au sens traditionnel du terme. La bulle applicative précédemment évoquée est simplement déployée sur demande sur le périphérique cible, soit entièrement pour une utilisation en mode déconnecté, s'agissant alors de virtualisation d'applications, soit en étant exécutée directement sur le serveur (*streaming*), s'agissant dès lors de virtualisation de sessions. Le mode de mise à disposition est généralement défini par le scénario d'accès (type de périphérique, qualité du réseau, localisation). Si l'application est exécutée directement sur le serveur, elle bénéficie en conséquence d'une puissance de calcul accrue.

5.3.2 Avantages

5.3.2.1 Élimination des conflits d'installation

Les bulles virtuelles applicatives ne nécessitent aucune modification de l'ordinateur hôte et s'exécutent indépendamment les unes des autres, évitant ainsi tout conflit potentiel entre applications ou entre les applications et le système d'exploitation. Les recours au Help Desk et les interventions des administrateurs sont ainsi réduits.

5.3.2.2 Portage d'application facilité

La virtualisation d'application rend possible le transfert de ces dernières vers une plateforme plus récente sans avoir à développer du code supplémentaire, à tester l'application pour la

nouvelle plateforme ou à la certifier à nouveau. Les déploiements d'applications anciennes vers un environnement plus récent sont par conséquent plus simples avec pour corollaire une mise en production plus rapide. Les temps d'interruption sont dès lors limités et les risques pour l'entreprise diminués.

5.3.2.3 Simplification des mises à niveau

Les mises à jour d'applications sont centralisées, une seule instance de chaque applicatif étant stockée côté serveur, au sein du centre de données. Aucune modification n'est par conséquent nécessaire côté client.

5.3.2.4 Simplification de l'administration du parc

Le déploiement, la configuration et la gestion des licences sont facilités par l'existence d'une plateforme dédiée et centralisée, permettant d'effectuer ces opérations sans exiger le déplacement d'un administrateur vers un quelconque poste client.

5.3.2.5 Amélioration du service aux utilisateurs mobiles

Un accès fiable et souple peut être garanti aux utilisateurs en déplacement, à partir de pratiquement tout périphérique à disposition. Aucune installation et aucun droit d'administrateur n'est requis pour se faire. Les applications peuvent même s'exécuter à partir d'un périphérique externe, comme une clé USB.

5.3.2.6 Renforcement de la sécurité

Lorsque l'application est exécutée directement sur le serveur, les données sont conservées à l'abri au sein du *datacenter*. Seuls les clics de souris, les frappes clavier et les mises à jour d'écran transitent par le réseau. De plus, il est possible de faire en sorte que seuls les utilisateurs autorisés puissent se connecter aux applications et par conséquent aux données y relatives. Il n'est nul besoin d'accorder des droits d'administrateur à un utilisateur pour exécuter une application qui l'exigerait en cas d'installation traditionnelle.

5.3.3 Solutions

Les principales solutions de virtualisation d'applications sont :

- Citrix **XenApp**™[62] ;
- VMware **ThinApp**™[63] ;
- Microsoft **Application Virtualization**™ (App-V™[64]) ;
- Novell **ZENworks® Application Virtulization**™[65] ;
- Systancia **AppliDis Fusion**™[66].

5.4 Stockage

La virtualisation du stockage, appelée également abstraction du stockage, existe depuis le début des années 2000. Son taux de pénétration du marché croit de façon rapide depuis lors. En effet, les entreprises ayant acquis au fil des années des volumes conséquents de

[62] http://www.citrix.fr/French/ps2/products/product.asp?contentID=186.
[63] http://www.vmware.com/fr/products/desktop_virtualization/thinapp/overview.html.
[64] http://www.microsoft.com/en-us/windows/enterprise/products-and-technologies/virtualization/app-v.aspx.
[65] http://www.novell.com/fr-fr/products/zenworks/applicationvirtualization/.
[66] http://www.systancia.com/fr/AppliDis-Fusion-4.

stockage, provenant de fabricants différents, ne possédant pas forcément les mêmes connectiques et, de surcroît, incorrectement exploités au niveau les LUN. Il n'est pas rare de provisionner un certain espace disque à l'attention d'un serveur pour se rendre compte par la suite que ce dernier a été surdimensionné. En multipliant l'espace ainsi perdu dans un centre de données contenant plusieurs armoires de type rack, l'espace en question peut devenir rapidement considérable.

La virtualisation du stockage permet de dimensionner correctement son stockage avec, pour corollaire, de conséquentes économies de moyens financiers issues notamment de la diminution du nombre de baies de disques, la diminution de la consommation électrique générée par la climatisation des locaux concernés et la diminution des coûts de possession[67].

5.4.1 Concepts

La virtualisation du stockage consiste à fédérer plusieurs ressources de stockage indépendantes et éventuellement hétérogènes en une ressource centralisée. Des volumes ou disques de stockage virtuels logiques sont créés à partir de ces différentes ressources et présentés par la suite aux serveurs concernés.

Ainsi, quoiqu'une entreprise ait fait l'acquisition de NAS, de baies de stockage iSCSI[68] et/ou de baies de stockage fibre optique, que ces dispositifs soient dotés de disques SATA, SAS ou SSD ou qu'ils aient été produits par différents fabricants, la société en question pourra tout de même centraliser la totalité de l'espace disque disponible et l'administrer à partir d'un seul outil. Lorsque l'acquisition de matériel de stockage supplémentaire s'avérera nécessaire, la société n'aura pas à se soucier de la marque ou de la technologie de ce dernier puisque cette nouvelle ressource sera virtualisée. L'ajout d'éléments de stockage se fait au sein de l'espace logique, en toute transparence, rendant ce type de solution extrêmement évolutive.

Nous pouvons partir du principe que la capacité de stockage ne dépend plus de la capacité physique d'un dispositif mais uniquement des contraintes budgétaires.

Un espace de stockage très important peut ainsi être mis à la disposition d'un serveur à partir de plusieurs baies originellement isolées, ce qui n'aurait pas été possible si elles étaient demeurées séparées. Un environnement récent peut également être pérennisé par ce biais (dans le cas, par exemple, d'une fusion de sociétés ayant donné lieu à l'existence au sein de la nouvelle structure de baies de stockage hétérogènes issues des anciennes entreprises). Enfin, l'administration du stockage peut être réalisée à partir d'une seule et unique console.

La virtualisation du stockage consiste en une abstraction de la couche physique du stockage, les serveurs stockant l'information sans se soucier de l'emplacement réel des

[67] Ensemble des frais se rapportant à la détention de matériel, comme les primes d'assurances, la location ou l'amortissement du local au sein duquel le matériel est situé, le nettoyage, etc.
[68] Protocole d'encapsulation servant à transporter un protocole de plus haut niveau, en l'occurrence celui correspondant au standard SCSI (Small Computer System Interface) qui définit un bus informatique dont la particularité est de déporter la complexité vers le périphérique lui-même. Il s'agit donc d'une interface plus rapide et plus universelle mais également plus complexe.

données. L'accès à ces dernières devient donc logique et non physique. Considérons, à titre d'exemple, trois baies de stockage virtualisées. La solution de virtualisation de stockage est capable de répartir physiquement les informations, selon des algorithmes qui lui sont propres, sur les trois baies, elles-mêmes logiquement entrelacées (RAID 0 ou *striping*). L'intégrité des données de chacune des baies est garantie par un RAID 5. Le serveur à qui l'espace de stockage a été alloué ne voit que la partie logique de ce stockage. Outre l'optimisation des performances d'accès et l'amélioration de la sécurité, cette technologie permet de simplifier considérablement l'administration des baies.

L'emplacement physique des données est répertorié à l'aide de métadonnées. Ces dernières assurent la mise en cohérence (*mapping*) des emplacements physiques des données avec les emplacements logiques correspondant. Lorsqu'un LUN est sollicité par une requête, la solution de virtualisation du stockage interroge les métadonnées afin de pouvoir déterminer quelles sont les baies qui stockent les données. La requête logique est donc convertie en requête physique. Ces métadonnées, ainsi que les fichiers de configuration du SAN virtuel, sont donc essentiels et doivent faire l'objet d'une attention particulière, notamment au niveau de leur sauvegarde. Cette remarque s'applique tout particulièrement au SAN composé d'une unique baie de stockage. Les données contenues sur cette dernière n'étant pas répliquées sur une autre baie.

L'administration du stockage se situe donc au niveau de la solution de virtualisation de stockage. L'administration des baies s'avère toujours nécessaire mais est de moindre complexité. En effet, seuls les volumes (LUN) sont créés au niveau des baies. Tout le paramétrage se trouve au niveau de la solution de virtualisation de stockage.

Les principaux fournisseurs de solutions de virtualisation de stockage sont DataCore™ et FalconStor®**.** Le premier propose un produit nommé **SANsymphony™-V**, qu'il n'hésite pas à qualifier de premier hyperviseur de stockage. Le second propose **Network Storage Server Virtual Appliance** (NSS VA).

L'architecture relative à la virtualisation du stockage peut être symétrique (in-band) ou asymétrique (out-band).

5.4.2 Architecture symétrique

Cette architecture, désignée également par le terme *in-band*, consiste à faire en sorte que les serveurs de production soient connectés au stockage par l'intermédiaire de l'*appliance* de virtualisation de stockage (qui peut également être un serveur dédié, voire être intégré aux commutateurs SAN). Ce dernier peut toutefois apparaitre comme un goulet d'étranglement (SPOF, *Single Point of Failure*) et doit dès lors impérativement être sécurisé. Cette sécurisation peut se faire par le biais de la création d'un *cluster* comprenant un deuxième serveur de virtualisation de stockage.

La figure 5-4, ci-dessous, illustre cette infrastructure et le goulet d'étranglement potentiel, puisque le serveur de virtualisation du stockage n'est pas dupliqué.

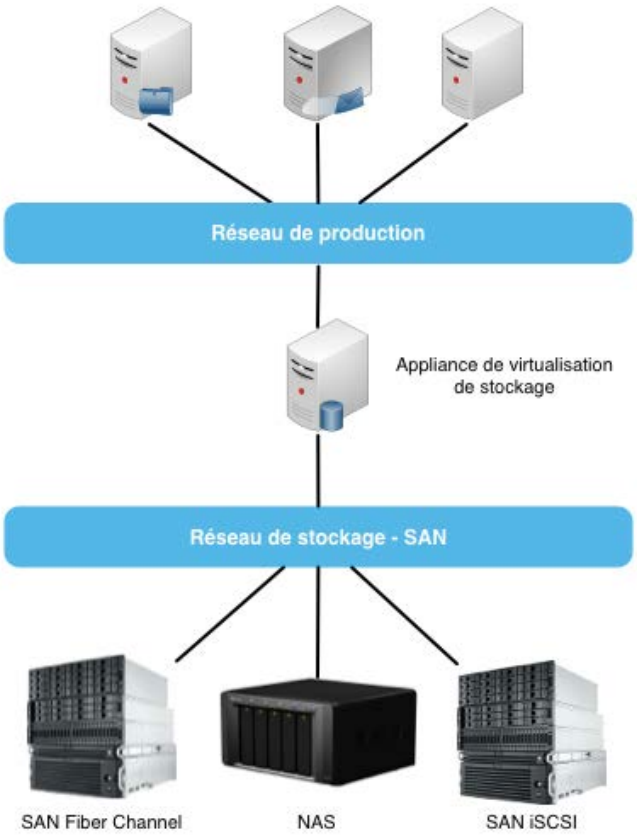


Figure 5-4 : Symétrique ou In-Band (Source : Bonnes pratiques, planification et dimensionnement des infrastructures de stockage et de serveur en environnement virtuel de Cédric Georgeot)

L'*appliance* de virtualisation du stockage doit être correctement dimensionnée, la totalité des requêtes transitant par cette dernière. Dans le cas contraire, les accès au stockage pourraient être peu performants, voire peu fiables. Une quantité importante de cache peut lui être octroyée pour pallier ce genre de problème. Nous précisons que le débit proposé par les liens Ethernet ou FC doit également être suffisamment performant, sans quoi le cache se trouvera rapidement saturé.

La plupart des solutions offertes par le marché s'avèrent être *in-band*, l'implémentation de cette technique étant assez souple. Il est en effet inutile d'ajouter une couche logicielle sur les serveurs, ce qui fait d'elle une solution totalement transparente.

5.4.3 Architecture asymétrique

Cette architecture, appelée également *out-band*, permet d'éviter le SPOF évoqué précédemment (section 5.4.2). Contrairement au processus en vigueur au sein d'une architecture *in-band*, les données ne transitent pas par le serveur de virtualisation de stockage, mais directement des serveurs de production à l'espace de stockage (SAN).

Pour assurer un tel traitement, un agent est installé sur les serveurs de production. Ces derniers s'en serviront pour communiquer avec le serveur de virtualisation de stockage. Le serveur de production utilise son agent pour indiquer au serveur de virtualisation de stockage les emplacements utilisés sur les baies. Le serveur de virtualisation de stockage passe par l'agent pour faire parvenir ses requêtes à un serveur de production (cf. figure 5-5).

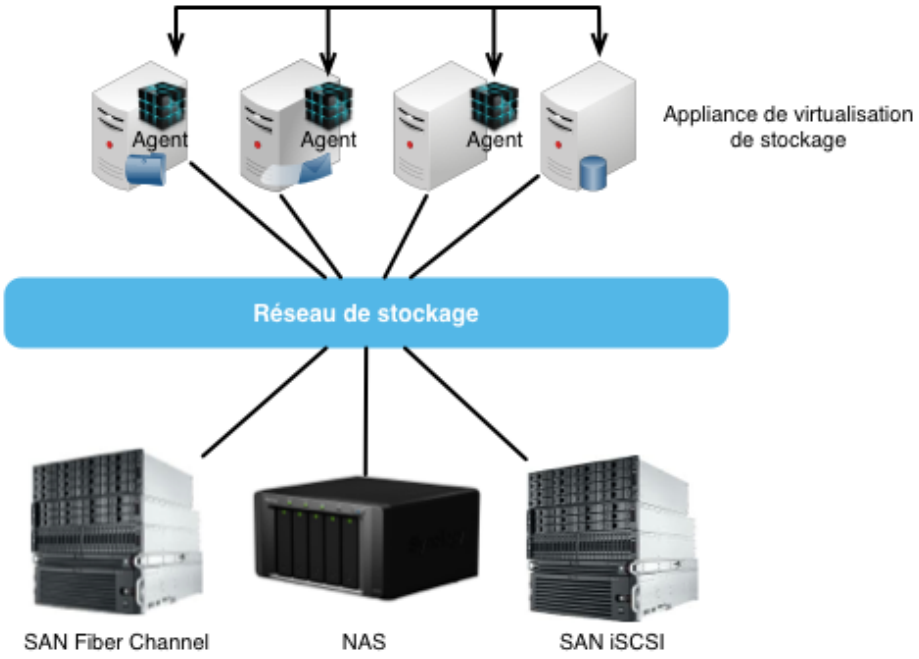


Figure 5-5 : Asymétrique ou Out-Band (Source : Bonnes pratiques, planification et dimensionnement des infrastructures de stockage et de serveur en environnement virtuel de Cédric Georgeot)

Cette architecture est plus performante car le serveur de virtualisation de stockage n'est sollicité que pour convertir les requêtes logiques en requêtes physiques. Les éventuels engorgements (*Bottleneck*) sont inexistants car les données ne transitent pas par celui-ci.

L'installation d'un agent sur les serveurs peut toutefois s'avérer problématique dans les environnements de virtualisation, ce qui a pour effet de limiter le nombre de déploiements de ce type d'architecture malgré des avantages évidents. Les entreprises préférant opter pour une architecture symétrique performante et simple à mettre en œuvre.

5.4.4 Thin Provisioning

Cette technique permet d'allouer des blocs à la demande pour un volume alors qu'habituellement l'allocation de l'ensemble des blocs était effectuée au moment de la création dudit volume (*Thick Provisioning*).

La figure 5-6 illustre le fait que l'espace inhérent à un disque logique peut être réparti au sein de plusieurs LUN, eux-mêmes créés à partir du *pool* de stockage.

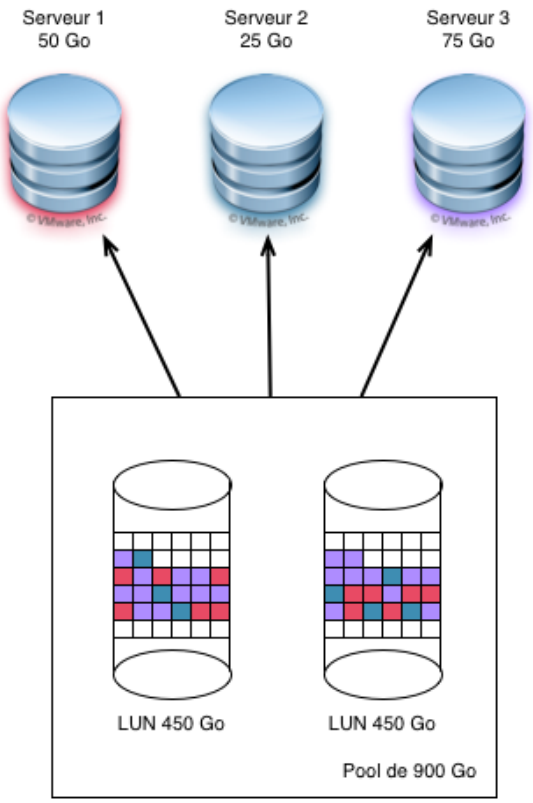


Figure 5-6 : Thin Provisionning (Source : Bonnes pratiques, planification et dimensionnement des infrastructures de stockage et de serveur en environnement virtuel de Cédric Georgeot)

On parle alors de sur-allocation du stockage, le serveur de virtualisation présentant plus de stockage aux différents serveurs que ces derniers n'en disposent réellement. Il s'avère bien souvent que la capacité d'un volume alloué à un serveur ait été surestimée. Cette technique autorise une grande flexibilité au niveau de l'allocation du stockage sans avoir, de surcroît, à estimer le taux de croissance éventuel.

En virtualisation de stockage, nous parlons de *pool* de ressources de stockage (éléments mis en commun). La virtualisation de stockage permet de provisionner à la demande les blocs (en répertoriant les emplacements physiques dans les métadonnées) au sein d'un tel *pool*. Lorsqu'un *pool* arrive à saturation, il convient simplement d'y rajouter une ressource de stockage quelle qu'elle soit.

Nous observons sur la figure 5-7 que le disque de gauche (vert) comprend un espace de 40 Go qui lui a d'ores et déjà été alloué de manière tout à fait classique (Thick Provisioning). Même si les données contenues sur ce disque n'occupent pas les 40 Go disponibles, cet espace ne peut pas être utilisé à d'autres fins. Les autres volumes (rouge et bleu) bénéficient du *Thin Provisioning*. Même si 60 Go ont été provisionnés pour le disque rouge, les 20 Go qui ne sont présentement pas utilisés sont simplement réservés. Ils ne seront véritablement utilisés que le jour ou le volume en aura réellement besoin et restent disponibles dans l'intervalle.

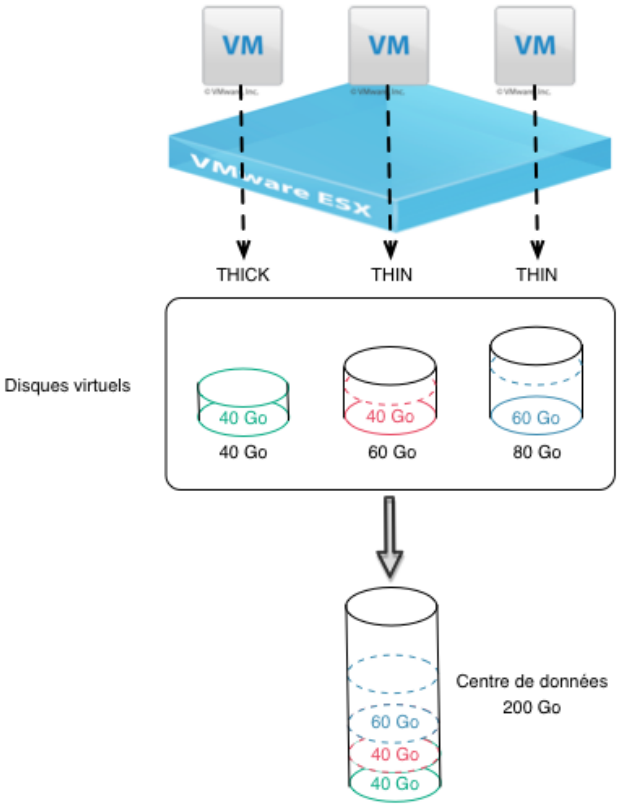


Figure 5-7 : Regroupement des blocs dans un centre de données (Source : www.vmware.com)

Grâce à cette technique, il est par exemple possible de provisionner un volume de 1 To sur un emplacement physique qui ne correspondrait qu'à 250 Go au sein du *pool*. Il est dès lors possible d'acquérir du stockage au fur et à mesure de l'accroissement de la demande. Les investissements initiaux dédiés au stockage peuvent ainsi être reportés sur un autre budget.

Nous attirons toutefois l'attention du lecteur sur les risques inhérents au concept d'*Over Provisioning* ou *Over Subscription*. Étant donné que le *Thin Provisioning* permet de signaler un espace de stockage virtuel plus important que la capacité réelle, un sur-provisionnement de stockage peut apparaitre. Il est dès lors nécessaire de surveiller de près l'utilisation du stockage, sans quoi l'espace physique pourrait brusquement manquer. Cette situation peut avoir pour conséquence un achat non budgété de matériel coûteux, et ce, dans l'urgence.

Fort heureusement, les solutions actuelles de virtualisation du stockage (notamment SANsymphony™-V de DataCore™) sont capables d'alerter les administrateurs par le biais de la console d'administration. Différents niveaux d'alarme sont disponibles, tels que de simples informations, des mises en garde ou des alertes critiques.

5.4.5 Fonctionnalités avancées

Comme nous l'avons vu plus haut, les solutions de virtualisation de stockage permettent de centraliser diverses ressources de stockage hétérogènes. Hormis cet avantage non négligeable, plusieurs fonctionnalités ont été élaborées afin d'améliorer drastiquement les performances offertes par cette technologie.

Une de ces fonctionnalités, essentielle pour la mise en œuvre d'une architecture performante, est le mécanisme de cache. Les puissants processeurs et la mémoire vive des serveurs x86-64 sur lesquels les hyperviseurs de stockage fonctionnent peuvent être mis à contribution par ces derniers pour la mise en œuvre de **méga-cache**, ce qui garantit une augmentation conséquente des performances pour des temps de latence diminués.

DataCore™ annonce, à titre d'exemple, que jusqu'à 1 To de mémoire vive peut être ainsi configuré en cache, par nœud. Ceci autorise, en tirant également parti des ressources E/S et de la capacité de l'unité centrale, l'exécution de certaines fonctions avancées des solutions de stockage contrôlées par l'hyperviseur. Les données écrites ou lues sur les disques migrent dès lors très rapidement vers ou à partir des caches. Les applications s'exécutent également plus rapidement que si elles devaient se connecter directement aux disques.

Il est également possible de mettre en œuvre le stockage à plusieurs niveaux (***tiering***) à partir d'un *pool*. Certaines solutions, à l'instar de celles proposées par Dell Compellent™, l'automatisent. Cette technique consiste à affecter différentes catégories de données à différents types de supports de stockage, afin de réduire le coût total du stockage. Les catégories dont il est question peuvent être fondées sur les niveaux de protections nécessaires, les exigences de performance, la fréquence d'utilisation, parmi d'autres considérations.

Ainsi, au sein d'un stockage multi-niveaux, les données essentielles à l'activité de l'entreprise ou récemment accédées peuvent être stockées sur des baies connectées par de la fibre optique et dotées de médias de haute qualité (donc passablement coûteux) comme des disques SSD, s'agissant du *pool* de *tier* 1. D'autres baies moins performantes peuvent servir à former d'autres *pools*, de *tier* 2 et 3. Les bonnes pratiques concernant l'organisation des différents niveaux correspondent généralement aux informations mentionnées dans le tableau 5-1 ci-dessous :

Niveau (*Tier*)	Pourcentage de la capacité globale de stockage	Type de disque
1	15%	SSD
2	25%	SAS (15'000 t/m)
3	60%	SAS (10'000 t/m) ou SATA (7'200 t/m)

Tableau 5-1 : Bonnes pratiques, en termes de pourcentage de la capacité par niveau et de type de disques dans un réseau de stockage (Source : Paul Santamaria, Storage Solution Architect chez DELL®)

Effectuer des migrations de données est grandement simplifié par le fait que l'accès à ces dernières est basé sur des métadonnées. Ainsi, lorsque le changement d'une baie de stockage s'avère nécessaire, les données déplacées doivent simplement faire l'objet d'une mise à jour des métadonnées. Ce procédé est totalement transparent pour les serveurs en production, ce qui rend ce type de solution extrêmement flexible.

Les disques virtuels disponibles par le biais des solutions de virtualisation de stockage peuvent être configurés en haute disponibilité (HA) et apparaitre comme des disques multiport partagés uniques, même si dans les faits ils se composent de deux images miroir distantes mises à jour simultanément. Ce type de procédé est particulièrement utile pour assurer l'intégrité des données en cas d'accident compromettant le *datacenter*.

5.5 Réseau

Nous avons vu précédemment que virtualiser consiste à faire abstraction de la couche matérielle physique en lui substituant du matériel virtuel. En toute logique, nous pourrions évoquer la virtualisation des cartes réseau des machines invitées comme étant liée au domaine de la virtualisation du réseau. Or, il n'en est rien. La virtualisation d'un tel périphérique d'E/S revient à virtualiser une partie de l'équipement et s'apparente de ce fait à la virtualisation des serveurs ou des stations de travail. Cette carte réseau virtuelle (ou vNIC) est toutefois indispensable pour se connecter à un réseau virtuel.

Comme c'est déjà le cas dans les autres domaines d'application évoqués dans le présent chapitre, la virtualisation du réseau consiste à partager la même infrastructure réseau physique entre plusieurs réseaux virtuels totalement isolés. Des logiciels de gestion – en particulier les composants de réseau virtuel – vont permettre d'assurer la communication entre les interfaces réseau évoquées ci-dessus et les interfaces physiques sous-jacentes. Une partie de la complexité est ainsi transférée du matériel vers le logiciel. C'est d'ailleurs également le cas pour les autres domaines d'application de la virtualisation.

Par réseau physique, nous faisons référence à la bande passante, aux ressources processeur des dispositifs réseaux tels que les routeurs, etc. Il ne s'agit donc pas uniquement d'une partition logique du réseau basée sur le VLAN (cf. section 5.5.1, VLAN).

Or les réseaux de communication se caractérisent par leur grande hétérogénéité, tant au niveau des couches « basses » regroupant l'infrastructure, les accès divers, etc., que les couches « hautes » rassemblant notamment les problèmes d'incompatibilité des différents formats de données, ou celle des logiciels applicatifs. Les interconnexions entre les différentes médias (fils de cuivre, fibre optique, etc.), l'interopérabilité et l'échange doivent pourtant être de mise pour que les transactions soient possibles au travers du réseau.

La virtualisation des réseaux exige la levée d'une quantité de verrous à tous les niveaux du modèle OSI. Nous avons décidé de ne pas entrer dans les détails à ce niveau, ceci dépassant le cadre du présent document.

De plus, le domaine de la virtualisation du réseau est aujourd'hui en grande mutation. Nous pouvons considérer d'une part que l'interconnexion de l'environnement virtuel avec le réseau physique est arrivée à maturité. Virtualiser les serveurs ou les stations de travail n'empêche nullement ces derniers d'accéder au réseau LAN ou WAN. Nous évoquons d'ailleurs au sein du présent chapitre le vSwitch (commutateur virtuel) et la virtualisation des contrôleurs hôte de bus, qui sont des éléments majeurs d'un réseau virtuel. Nous effectuons également un rappel du principe de VLAN qui est intimement lié à l'accès au réseau physique par les machines virtuelles.

D'autre part, le domaine de la virtualisation du réseau s'étend aujourd'hui au-delà du centre de données propre à une entreprise, auquel les éléments du précédent paragraphe se réfèrent. En effet, les acteurs du marché de la virtualisation s'intéressent de près à la notion de ***Software Defined Networking*** à laquelle nous faisons référence à la section 7.2.4, Perspectives, relative aux perspectives liées à la virtualisation. Ce nouveau paradigme

d'architecture réseau ouvre la voie à de multiples applications relatives au nuage (*cloud computing).*

Nous n'aborderons pas ces notions au sein de présent paragraphe mais nous tenons tout de même à attirer l'attention du lecteur sur le fait que les développements à venir en matière de virtualisation seront liés de près à la virtualisation du réseau.

5.5.1 VLAN

Le VLAN (de l'anglais *Virtual Local Area* pour Réseau local virtuel) n'est pas lié à la virtualisation au sens où nous l'entendons dans ce document, même si son principe s'en rapproche. En effet, même si le VLAN est un réseau local de type logique (regroupant un certain nombre de périphériques de manière logique et non physique), il existe indépendamment de la virtualisation, au travers d'une infrastructure réseau physique traditionnelle (routeurs et commutateurs physiques). Nous l'évoquons ici pour nous assurer que le concept sous-jacent est maitrisé, eu égard à son importance dans la virtualisation du réseau.

Le VLAN permet de se passer des limitations de l'architecture physique (notamment les contraintes d'adressage), en partitionnant logiquement le réseau physique. Des domaines de diffusions distincts sont ainsi créés.
Il existe différentes topologies de VLAN, basées sur le critère de commutation et le niveau auquel ce dernier s'effectue :

- Le niveau 1 – appelé **VLAN par port** (*Port-Based VLAN*) – définit un réseau virtuel en fonction des ports de raccordement sur le commutateur ;
- Le niveau 2 – appelé **VLAN par adresse MAC** (*MAC Address-Based VLAN*) définit un réseau virtuel en fonction des adresses MAC des stations. Ce type de VLAN est plus souple que le niveau 1, le réseau étant indépendant de la localisation du périphérique ;
- Le niveau 3. Plusieurs types de VLAN 3 existent :
    - Le **VLAN par sous-réseau** (*Network Address-Based VLAN*) associe des sous-réseaux en fonction de l'adresse IP source des datagrammes. Ce type de solution apporte un grand confort, la configuration des commutateurs se modifiant automatiquement dans le cas du déplacement d'un périphérique. Les informations contenues dans les paquets devant être analysées plus finement, une légère dégradation de performances peut être constatée ;
    - Le **VLAN par protocole** (*Protocol-Based VLAN*) définit un réseau virtuel par type de protocole (TCP/IP, IPX, AppleTalk™, etc.), regroupant ainsi toutes les machines utilisant un protocole identique au sein d'un même réseau.

Le VLAN permet d'offrir de nouveaux réseaux à un niveau supérieur à la couche réseau physique. Il s'agit donc d'un fort apparentement avec la virtualisation telle que nous l'avons évoquée dans ce document.

La conséquence de l'existence de ces réseaux virtuels est :

- Que le réseau peut être modifié par simple paramétrage des commutateurs, offrant ainsi une souplesse d'administration accrue ;

- Que les informations sont encapsulées au sein d'une couche supplémentaire, augmentant ainsi le niveau de sécurité ;
- Que la diffusion du trafic sur le réseau peut être réduite.

Les normes IEEE 802.1d, 802.1p et 802.1q définissent le VLAN.

Pour clore cette section, nous évoquerons également la notion de ***trunk*** qui est intimement liée à celle du VLAN. Un *trunk* est une interconnexion entre deux commutateurs qui vise à préserver l'appartenance de chaque trame à un VLAN en particulier. Chaque trame est donc encapsulée de façon à conserver son numéro de VLAN. La norme 802.1q (dot1q) de l'IEEE encadre ce fonctionnement.

5.5.2 Commutateur

Tout comme un serveur physique a besoin d'un commutateur physique pour se connecter au réseau, une machine virtuelle nécessite un commutateur virtuel pour faire de même. Les solutions y relatives proviennent tant des éditeurs de solutions de virtualisation que des constructeurs de périphériques réseau physiques.

Le commutateur virtuel n'est autre qu'un logiciel, intégré à l'hyperviseur, qui se comporte à l'identique de son homologue physique. Il est d'ailleurs raccordé au commutateur physique auquel est raccordée la machine physique hôte. La virtualisation des réseaux n'est donc aucunement destinée à éliminer les commutateurs physiques de l'infrastructure mais à permettre aux machines virtuelles de s'intégrer avec davantage de souplesse au réseau. Il s'agit notamment pour ces dernières de ne pas avoir à se soucier du nombre de cartes physiques disponibles.

Ce concept a été inauguré par le leader du marché, VMware®, s'agissant du vSwitch™. Il a cependant fallu rapidement corriger les importantes limitations dont souffrait ce dernier. En plus de ne pas être administrable, il n'était opérationnel qu'à l'échelle d'un seul hyperviseur, donc d'un seul serveur physique. Ainsi, lorsqu'une VM était déplacée d'un serveur physique à un autre, toutes les informations liées au réseau étaient perdues, données de *monitoring* comme VLAN.

Ces limitations ont été corrigées en avril 2009 par VMware® avec l'avènement du **Distributed vSwitch™** (DVS), appelé **vSphere Distributed Switch™**[69] au sein de l'hyperviseur actuel de l'éditeur. Ce dernier permet de configurer la commutation d'accès des machines virtuelles de l'ensemble du *datacenter* grâce à une interface centralisée, la mise en réseau des machines virtuelles étant dès lors fortement simplifiée.

Une fois un serveur physique ajouté à une ferme, le nouvel hyperviseur hérite simplement de la configuration des autres, informations de monitoring y compris.

VMware® et Cisco®[70] ont finalement collaboré à la mise au point du **Nexus 1000v™**[71] dont les fonctionnalités étendent celles du DVS. Ce commutateur virtuel s'intègre à la console de

[69] http://www.vmware.com/fr/products/datacenter-virtualization/vsphere/distributed-switch.html.
[70] Entreprise leader dans le domaine des solutions réseau (http://www.cisco.com/web/FR/index.html).

Cisco® et hérite de toutes les fonctions de la gamme Nexus, tant en termes de supervision qu'en termes de performances et de sécurité. Il permet notamment d'analyser le trafic aux niveaux deux et trois (cf. modèle OSI[72]) et d'agréger les liens en entrée et sortie du serveur physique. Il supporte également les protocoles et mécanismes standards de supervision et de sécurité, comme Netflow, SNMP, Radius, ACL (*Access Control List*) ou Private VLAN. Nous précisons que le Nexus 1000v est payant, contrairement au DVS qui est intégré en standard dans vSphere™ (hyperviseur de type 1 de VMware®).

Il convient de préciser que si le DVS et le Nexus 1000v™ sont disponibles uniquement chez VMware®, la communauté Open Source a mis au point **Open vSwitch™**[73], commutateur virtuel capable d'opérer au niveau de plusieurs hyperviseurs. Il s'agit notamment du switch virtuel par défaut de XenServer™ 6.0, parmi d'autres solutions d'hypervision.

L'importance de ces solutions de virtualisation de la commutation réseau est capital, dès lors qu'il s'agit d'industrialiser la virtualisation, avec pour corollaire l'obligation de gérer efficacement les problématiques de haute disponibilité, de performances et de sécurité. Les fonctionnalités d'un commutateur virtuel comme le DVS sont très importantes quand le taux de consolidation devient élevé et que les applications s'avèrent gourmandes en ressources réseau. Un trafic élevé entre certaines VM peut par exemple être détecté, une association logique de ces dernières pouvant être mis en place pour éviter le problème. En cas de déplacement de ces machines virtuelles vers un autre serveur physique, via un outil tel que VMware vMotion™[74], elles seront ainsi transférées toutes en même temps, ce qui réduira par la suite le trafic entre hyperviseurs.

Nous avons abordé les notions de VLAN et de *trunk* au sein de la section précédente dans un but bien précis. Un vSwitch™ peut parfaitement permettre aux machines virtuelles d'accéder à tout VLAN disponible. Nous n'entrerons cependant pas outre mesure dans les détails à ce sujet.

5.5.3 Autres éléments du réseau

La virtualisation induit des besoins de connectivité qui dépassent la sphère des commutateurs virtuels. Le marché de la connectivité réseau comprend un nombre peu important d'acteurs qui ont mis au point des fonctionnalités essentielles, s'agissant notamment des **contrôleurs hôte de bus** (HBA pour *Host Bus Adapter* en anglais). Ces derniers sont des cartes d'extension qui permettent de connecter des systèmes hôtes (généralement des serveurs) à des périphériques réseau ou de stockage. Il s'agit des lors d'un domaine critique lorsque l'on considère l'importance du réseau de stockage (SAN) dans une infrastructure virtuelle.

Les acteurs principaux de ce marché sont **QLogic®** et **Emulex®**. La technologie **LightPulse Virtual**[75], mise au point par cette société, a été conçue pour permettre une amélioration de la connectivité des machines virtuelles, en particulier avec les périphériques de stockage. Nous

[71] http://www.vmware.com/products/cisco-nexus-1000V/overview.html.
[72] Au niveau de la couche **liaison de données** et **réseau**, soit au niveau des trames et des paquets (http://fr.wikipedia.org/wiki/Mod%C3%A8le_OSI).
[73] http://openvswitch.org/.
[74] http://www.vmware.com/fr/products/datacenter-virtualization/vsphere/vmotion.html.
[75] http://www.emulex.com/solutions/data-center-virtualization/lightpulse-virtual-hba-technology.html.

l'évoquons au sein de cette section comme un exemple de virtualisation des HBA Fibre Channel.

Cette technologie est présente sur les contrôleurs hôte de bus Fibre Channel (FC) 4 et 8 Gbit/s d'Emulex® et sur les cartes convergentes (CNA pour *Converged Network Adapter*) Fibre Channel over Ethernet (FCoE) que l'on retrouvera au sein du matériel proposé par de nombreux constructeurs (notamment HP®, IBM® et Dell®). Le CNA est un périphérique d'E/S combinant un HBA et une NIC (*Network Interface Controller*).

LightPulse Virtual repose sur les technologies **N_Port ID Virtualization** (NPIV) et **Virtual Fabric**.

La première offre la possibilité aux utilisateurs de **virtualiser les fonctionnalités d'un adaptateur FC**. Ainsi, un port physique FC peut être partagé en plusieurs N_Port virtuels.

En effet, la virtualisation des serveurs x86 est fortement dépendante du SAN qui est à même de fournir l'espace de stockage partagé nécessaire à la mise en œuvre de la haute disponibilité. Comme nous l'avons vu précédemment, lorsque des machines virtuelles sont stockées sur une partition hébergée par un SAN, les serveurs hôtes y accèdent par le biais de cartes HBA. Des liaisons basées sur la fibre optique sont souvent mises en place à cet effet (fonctionnant sur la base du protocole Fibre Channel).

Dans un tel contexte, une partition peut être visible par plusieurs serveurs hôte. Il s'agit du *LUN Masking* (qui est parfois qualifié à tort de *zoning*). Des fonctionnalités qui permettent de migrer des machines virtuelles d'un hôte à un autre sans interruption de service, telles que vMotion™ de VMware® ou Quick Migration™ de Microsoft® s'appuient sur ce procédé.

Il convient de préciser que tous les LUN sont référencés comme étant connectés directement aux HBA des serveurs physiques, ces derniers (leurs HBA) étant eux-mêmes référencés au sein du SAN par un World Wide Name (WWN).

Dans un tel contexte, NPIV permet d'attribuer à chaque machine virtuelle une adresse de type WWN, autorisant ainsi plusieurs machines virtuelles à partager la même HBA et de disposer de son N_Port. Il devient donc possible, pour l'administrateur, de gérer au mieux le SAN et les interactions des machines virtuelles avec ce dernier. En effet, le SAN, qui sans l'usage de NPIV ne voit que les HBA physiques, est capable de voir chaque machine virtuelle grâce à cette technologie.

Il va sans dire que les ports ainsi attribués suivent la machine virtuelle en cas déplacement d'un hôte à un autre.

Quant à la seconde technologie, Virtual Fabric, elle **divise un SAN unique en plusieurs SAN logiques**, chacun possédant son propre jeu de services. Il est dès lors possible de consolider plusieurs SAN indépendants en un seul et unique SAN, en conservant la même topologie logique qu'avant la consolidation.

La figure 5-8, ci-dessous, illustre la virtualisation des fonctionnalités d'un adaptateur Fibre Channel et la division d'un SAN unique en plusieurs SAN logiques, fonctionnalités qui sont destinées à faciliter ce type de distribution.

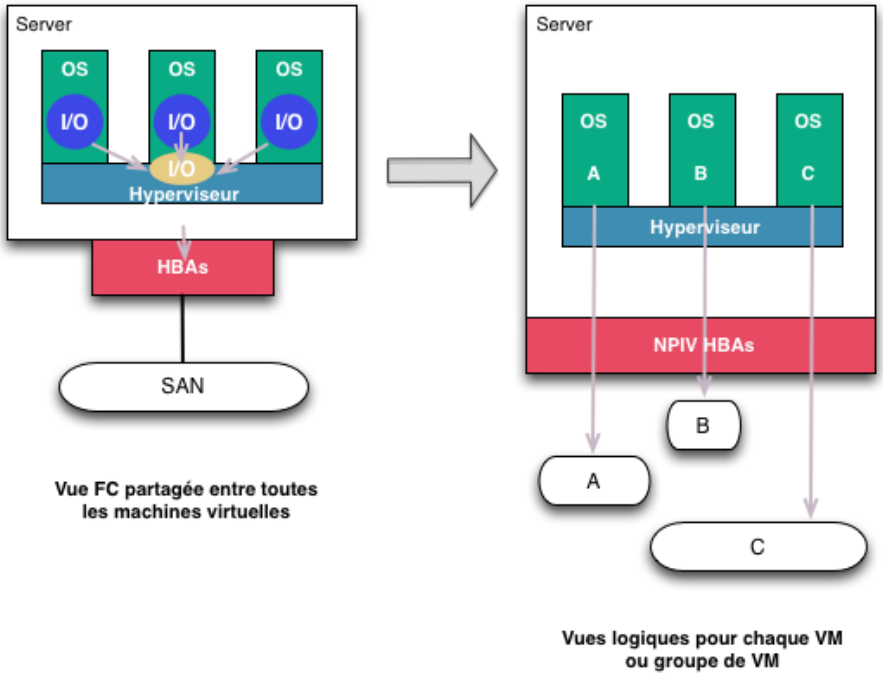


Figure 5-8 : Virtualisation des fonctionnalités d'un adaptateur Fibre Channel (Source : http://www.emulex.com/solutions/data-center-virtualization/lightpulse-virtual-hba-technology.html)

6 Construction de l'infrastructure virtuelle

Ce chapitre est destiné à accompagner le lecteur dans les choix en matière de serveurs et autres éléments de stockage auxquels il ne manquera pas d'être confronté s'il décide d'opter pour la virtualisation. Nous y référerons un certain nombre de critères et de bonnes pratiques auxquels il convient d'être attentif lorsque l'achat du matériel est en cours de planification. Ces critères et bonnes pratiques doivent être considérés par le lecteur comme une base à partir de laquelle il est sensé mener ses propres recherches.

Si le choix des éléments composant les serveurs est y abordé sous l'angle de la virtualisation, tel n'est pas le cas de ceux inhérents au stockage. En effet, nous nous intéressons tout particulièrement au stockage partagé, au sens classique, en tant que pièce maitresse de la virtualisation des serveurs. La virtualisation du stockage, et les *appliances* y relatives, ne sont pas évoquées dans le présent chapitre. En effet, le stockage n'a pas besoin d'être virtualisé pour fonctionner de concert avec un ou plusieurs serveur(s) hôte de machines virtuelles.

À la section 6.1, Planification, nous nous préoccupons de la manière avec laquelle évaluer les ressources existantes, en particulier en matière de serveurs, puisque ces derniers font généralement l'objet des premières mesures de virtualisation. Une connaissance précise de l'architecture matérielle des machines utilisées en production, ainsi que le degré de sollicitation de ces dernières sont en effet des éléments primordiaux pour juger des solutions à mettre en place. L'environnement virtuel n'en sera ainsi que plus performant.

Cette section est destinée à préparer la virtualisation des serveurs physiques existants. Il va de soi que si le projet de virtualisation dans lequel le lecteur est impliqué concernait une nouvelle implantation, la lecture de cette section ne serait pas véritablement utile. Nous soulignons toutefois que si tel était le cas, une planification adéquate demeurerait incontournable.

À la section 6.1.2, Dimensionnement des serveurs, les facteurs clés sur lesquels baser le choix des serveurs hôtes de machines virtuelles sont mis en évidence. Il s'agit en particulier, au terme de la lecture de cette section, de disposer des compétences nécessaires pour choisir une forme de serveur adaptée aux besoins et pour choisir avec pertinence les processeurs dont devront disposer les hyperviseurs. Au terme de la lecture de cette section, le lecteur doit être à même de choisir un type de mémoire vive pour ses serveurs hôtes et d'en définir la quantité. Certains principes de gestion de la mémoire propres aux hyperviseurs sont également abordés, leur fonctionnement influant sur les choix qui doivent être effectués en la matière. Enfin, ce dernier doit être capable de définir avec soin le nombre d'interfaces réseau dont les serveurs hôtes devront être équipés.

La section 6.1.2 s'achève sur la conversion P2V et les outils susceptibles d'accompagner le lecteur dans cette opération.

La section 6.1.3, Dimensionnement du stockage, est au stockage ce que la section précédente est aux serveurs. En effet, comme nous l'avons déjà mentionné au chapitre 2, Spécificités d'un environnement virtualisé, le stockage revêt une importance toute particulière en environnement virtuel puisqu'il héberge notamment les machines virtuelles. Il

s'agit dès lors pour lecteur de prendre des décisions appropriées, tant en matière de quantité d'espace disque ou de mémoire cache, qu'au niveau des contrôleurs de stockage ou des types de disques durs dont disposerons les baies de stockage.

Quant à la section 6.1.4, Choix du réseau de stockage, elle vise à accompagner le lecteur dans les choix qui s'imposeront à lui en matière de type de réseau de stockage. En effet, dimensionner correctement ses baies n'est pas tout. Encore faut-il interconnecter correctement ces dernières avec les serveurs hôtes. Cette section comporte dès lors des informations relatives aux protocoles et aux médias sous-jacents, permettant la connexion haut débit entre les serveurs et les systèmes de stockage.

Les notions d'agrégation, de redondance, ainsi que la répartition des charges que cette dernière autorise, sont également abordées dans la section 6.1.4.

Enfin, la section 6.2, Choix des fournisseurs, clôt le chapitre en dressant rapidement la liste des éditeurs de solutions de virtualisation, ainsi que celle des fabricants de matériel.

6.1 Planification

6.1.1 Capacités nécessaires

Une bonne planification est essentielle dans un projet de virtualisation. En effet, il est capital de connaitre les caractéristiques des données traitées au sein du système d'information pour dimensionner correctement l'infrastructure matérielle nécessaire et pour parvenir à la paramétrer. Il convient également d'être au fait des capacités offertes par le matériel existant.

Si nous considérons le domaine incontournable de la virtualisation des serveurs, nous devons considérer qu'à l'heure actuelle tous les serveurs sont potentiellement virtualisables. Même les serveurs consommant énormément de ressources, comme les serveurs de base de données, peuvent parfaitement être convertis en machine virtuelle sans forcément pénaliser les autres serveurs virtuels.

Il convient de préciser que la virtualisation de ce type de serveur, qui va croissant, est rendue possible grâce aux fonctionnalités offertes par les plateformes de virtualisation récentes, s'agissant du contrôle des E/S de réseau et de stockage. Ces fonctionnalités offrent en effet la possibilité de définir des priorités basées sur des règles propres aux applications métier sensibles. De plus, les éditeurs travaillent d'arrache-pied à une prise en charge optimale de ce type de serveur. À ce titre, la société VMware® a annoncé, durant le VMworld 2012[76], la mise en place de nouvelles fonctionnalités au sein de sa plateforme de virtualisation vSphere™ 5.1 comme Monster VM™, capable de prendre en charge jusqu'à 64 CPU virtuels (contre 32 l'année précédente) et de gérer plus de 1 million d'IOPS par VM. De quoi résoudre définitivement le problème de la virtualisation des applications critiques !

[76] Ayant eu lieu entre le 26 et le 30 août 2012 à San Francisco.

Une bonne analyse de l'infrastructure existante est nécessaire à l'obtention du meilleur ratio de consolidation possible. Elle permet également de prévoir au plus près le budget relatif au projet de virtualisation.

Les données les plus importantes à collecter sont :

- La quantité de **mémoire** installée et son taux d'utilisation ;
- La capacité des **processeurs** installés et leur taux d'utilisation ;
- La capacité des **disques durs** et l'espace consommé ;
- La capacité des **liens réseau** et leur taux d'utilisation.

Sans analyser ces informations, nous aurions tendance à acquérir la même « puissance serveur » dont nous disposons à l'heure actuelle. Il n'est probablement pas nécessaire d'en provisionner autant, puisque les serveurs sont généralement surdimensionnés. Les économies ainsi réalisées pouvant être allouées ailleurs.

Il convient d'ailleurs de tenir compte du fait que les serveurs ne sont pas utilisés de manière uniforme. Chacun d'entre eux occupe un rôle précis. En fonction de ces rôles, l'utilisation peut varier fortement d'une journée à l'autre ou durant une même journée. L'utilisation moyenne d'un serveur peut en effet correspondre à 20% de ses capacités, alors qu'elle atteindra 60% durant les pics d'activité. Il s'agit de tenir compte de ces pics d'activité que dans le cas où ces derniers surviennent régulièrement.

Nous précisons également que pour être considérée comme fiable, une analyse doit être menée sur une semaine au moins.

Cette opération d'audit préliminaire, habituellement qualifiée de ***capacity planning***, peut parfaitement être réalisée manuellement. Les caractéristiques pertinentes peuvent également être collectées à l'aide d'un logiciel spécialisé.

Nous précisons également que certains serveurs ne peuvent être virtualisés, l'utilisation de périphériques externes étant requise (USB, parallèle, série, etc.). Certains hyperviseurs autorisent toutefois la connexion directe de ce type de dispositifs au matériel virtuel. Il convient cependant de procéder à des tests préliminaires avant la mise en production de ce type de serveur virtuel.

6.1.1.1 Logiciels de planification de capacité

Les logiciels de ce type sont très nombreux. Certains d'entre eux sont particulièrement efficaces car développés spécifiquement pour évaluer une infrastructure physique destinée à être virtualisée. Ils sont toutefois **payants**. Nous citerons en exemple **PlateSpin Recon™**[77] de **NetIQ®** et **VMware® Capacity Planner™**[78]. Quant à la société **VKernel®**, appartenant à **Quest Software®**, elle commercialise un logiciel appelé **vOPS™**[79] qui est décliné en plusieurs versions. Les versions vOPS **Server Standard™** et vOPS **Server Entreprise™** sont payantes, tandis que la version vOPS **Server Explorer™**, moins élaborée, est gratuite.

[77] https://www.netiq.com/products/recon/.
[78] http://www.vmware.com/products/capacity-planner/overview.html.
[79] http://www.vkernel.com/.

Un certain nombre d'outils **gratuits** peuvent également permettre une planification adéquate de la capacité nécessaire aux serveurs. Nous dressons, ci-dessous, une liste non exhaustive de ces derniers, qu'ils soient prévus pour les environnements Windows® ou Linux™.

Windows®

- **Microsoft Assessment and Planning™[80] (MAP)** permet :
  - De collecter des informations essentielles, telles que l'inventaire matériel et logiciel ou l'analyse des performances ;
  - D'élaborer des rapports sur lesquels baser le dimensionnement d'une infrastructure virtuelle en devenir. Ces derniers, synthétisés dans un classeur Excel, comprennent des informations allant notamment des logiciels installés (version, service pack) à la mémoire utilisée, en passant par le nombre d'IOPS, le processeur le plus sollicité ou l'utilisation du réseau, et ce, serveur par serveur ;
  - De simuler une consolidation basée sur un serveur type qui est prévu à l'acquisition ;
  - D'exécuter une analyse du parc informatique pendant une période donnée. L'usage précis des ressources est ainsi délivré pour la période en question, serveur par serveur. L'estimation de la consolidation peut dès lors être effectuée sur la base d'un serveur type ;
- **Microsoft File Server Capacity Tool™[81] (FSCT)** a été conçu pour l'obtention de données portant sur les capacités offertes par les serveurs de fichiers uniquement. Il permet l'identification des goulots d'étranglement qui sont parfois observés sur ces derniers ;
- **Windows Performance Analyzer™[82]** est un utilitaire d'analyse de performances fourni au sein des kits de développement de Windows 7 et Windows Serveur 2008 R2. Il s'agit d'un outil basé sur l'Event Tracing for Windows (ETW). Il permet d'obtenir des informations précises sur l'utilisation des processus, des disques, des processeurs, etc. ;
- **Microsoft SQLIO™[83]** est un outil fournissant des informations spécifiquement liées aux capacités E/S d'une configuration donnée. Créé à l'origine pour tester les performances d'un serveur SQL en simulant des opérations d'E/S, il permet de simuler de la lecture, de l'écriture, de manière aléatoire ou séquentielle, avec des tailles d'E/S différentes ;
- Les systèmes d'exploitation Windows intègrent un **moniteur système** (PerfMon) qui dispose d'un nombre important de **compteurs de performance**[84]. Il peut être judicieux d'en surveiller une partie, s'agissant des compteurs :
  - **Disque physique** : renseignent sur les IOPS (Écriture disque/s et Lecture disque/s), sur la taille moyenne des E/S (Écriture disque, octet/s et Lecture disque, octet/s) ou sur les longueurs de file d'attente moyennes d'un disque (Longueur moyenne de file d'attente du disque). Ces dernières doivent par exemple être inférieures à 2, cette valeur représentant le nombre moyen des requêtes d'E/S ;

[80] http://www.microsoft.com/en-us/download/details.aspx?id=7826.
[81] http://www.microsoft.com/en-us/download/details.aspx?id=27284.
[82] http://msdn.microsoft.com/en-us/performance/cc709422.aspx.
[83] http://www.microsoft.com/en-us/download/details.aspx?id=20163.
[84] http://technet.microsoft.com/fr-fr/library/dd723635(office.12).aspx.

  - **Disque logique** (LUN) : donne des précisions sur la latence (Moyenne disque s/écriture ou Moyenne disque s/lecture). Ainsi, des valeurs comprises entre 1 et 15 correspondent à de bonnes performances. Si ces dernières se situaient entre 15 et 25, elles nécessiteraient d'être mises sous surveillance, alors qu'elles devraient déclencher un diagnostic si elles se situaient au-dessus de 25. Un compteur comme celui indiquant le temps d'accès au disque (Pourcentage du temps écriture du disque et Pourcentage du temps lecture du disque) devrait être surveillé de très près dans un environnement virtuel. Les valeurs y relatives ne devraient pas être supérieures à 75% ;
  - L'hyperviseur **Hyper-V™ dispose de ses propres compteurs**[85] ;
- **Microsoft Server Performance Advisor™**[86] est prévu pour offrir à Windows 2003 Serveur, les outils qui ont été intégrés d'office à partir de Windows 2008 Serveur. Il délivre des informations sur l'usage du processeur, de la mémoire, etc. Il permet également d'obtenir des données relatives aux rôles Active Directory, IIS, DNS, Service de fichiers, etc. ;
- **Microsoft File Server Resource Manager™** permet de mettre en évidence des informations relatives aux serveurs de fichiers, telles que l'espace disque consommé par utilisateur, les fichiers dupliqués ou ouvert le plus souvent, etc. L'analyse peut être effectuée pour une période donnée ou à un instant T.

Linux™

- **FIO**[87] permet d'effectuer des tests de stress mais également de mesurer les performances d'un serveur en lui soumettant des travaux ;
- **IOTOP**[88] est un logiciel similaire à la commande TOP mais s'appliquant à l'utilisation disque par processus, en temps réel ;
- **IOZONE**[89] est destiné à mesurer les performances d'un serveur. Il génère des rapports dans un tableur ;
- **Intel NAS Performance Toolkit™**[90] est prévu pour simuler des charges sur des serveurs de fichiers, comme des copies intensives, de la lecture de vidéos HD, etc. Il est capable de générer des graphiques.

6.1.2 Dimensionnement des serveurs

6.1.2.1 Forme du serveur (ou Facteur de forme)

Le format d'un serveur destiné à la virtualisation revêt une importance particulière. Passablement d'éléments reposeront par la suite sur le choix effectué à ce niveau, qu'il s'agisse de l'espace nécessaire au sein du centre de données, des capacités en matière de climatisation, etc.

Les serveurs adaptés à la virtualisation sont répartis en deux familles, les serveurs tour étant exclus d'emblée. En effet, même si ces derniers sont peu coûteux et relativement évolutifs, ils s'avèrent être passablement bruyants, encombrants, difficiles à gérer par rapport à

[85] http://technet.microsoft.com/en-us/library/cc768535%28BTS.10%29.aspx.
[86] http://www.microsoft.com/en-us/download/details.aspx?id=15506.
[87] http://freecode.com/projects/fio.
[88] http://guichaz.free.fr/iotop/.
[89] http://www.iozone.org/.
[90] http://www.intel.com/products/server/storage/NAS_Perf_Toolkit.htm.

d'autres formes de serveur et généralement trop peu puissants pour prendre en charge la virtualisation dans des conditions optimales.

Les serveurs éligibles pour la virtualisation sont généralement des serveurs **rack** ou des serveurs **lame** (*blade*).

Même si les serveurs issus de ces deux familles sont parfaitement adaptés à la virtualisation, le choix fait généralement débat. D'un côté, les coûts relatifs à l'acquisition de serveurs rack sont moindres et ces derniers n'exigent aucune modification au niveau de l'alimentation électrique du centre de données. D'un autre côté, l'administration centralisée, par le biais d'une seule console, de plusieurs serveurs lame est très appréciée. Au même titre, la mutualisation de plusieurs serveurs dans un même châssis, ce dernier comprenant alimentation électrique, refroidissement et accès au réseau, est fortement appréciée alors que la place disponible dans les centres de données n'est pas extensible à l'infini.

D'aucuns s'accordent pour mettre en avant la console de gestion centralisée, alors que certains avancent que cette dernière est efficacement remplacée par les consoles offertes par les plateformes de virtualisation (comme VMware vSphere™, pour ne citer qu'un exemple). Les uns mettent en avant l'excellent rapport qualité/prix des serveurs rack, ainsi que leur aptitude à être plus faciles à entretenir par l'équipe informatique que le serait un châssis comprenant plusieurs serveurs lame. Les autres mettent en avant que les commutateurs Ethernet et Fibre Channel intégrés au sein d'un tel châssis réduisent la complexité du câblage. Les premiers assurent qu'il est plus simple de faire évoluer un centre de données en utilisant des serveurs rack, en mettant en avant le fait qu'aucune adaptation des alimentations électriques n'est nécessaire. Les seconds assurent que la consolidation de ces mêmes alimentations au sein du châssis en réduit le nombre nécessaire, arguant qu'il en va également ainsi du nombre de ventilateurs.

Toutes ces affirmations sont exactes et parfaitement pertinentes. Aussi, nous ne prendrons pas position pour l'une ou l'autre des familles, chacune d'entre elles disposant de ses propres avantages. Il incombe au lecteur – en fonction de ses propres sensibilités et du contexte au sein duquel son projet de virtualisation doit voir le jour – de se forger sa propre opinion, en s'intéressant au grand nombre de prises de position qui abondent sur la toile ou dans la littérature y relative.

Toutefois, un élément incontournable, voire discriminant, dans le choix des serveurs est le nombre de cœurs physiques qu'il sera possible de réunir dans un nombre minimum de U. En effet, le nombre de VM par serveur dépend du nombre de cœurs physiques dont ce dernier dispose. Si l'espace disponible au sein du centre de données n'est pas suffisant pour pouvoir atteindre le nombre souhaité de cœurs en utilisant des serveurs rack, il sera peut-être nécessaire de privilégier des serveurs lame.

Nous précisons qu'un des coûts cachés d'un centre de données est relatif à la climatisation. En privilégiant les serveurs de haute densité, comme les serveurs lame, la puissance de réfrigération nécessaire est diminuée, puisque ces derniers chauffent moins. Des économies substantielles peuvent donc être générées par ce biais.

6.1.2.2 Choix des processeurs

Pour parvenir à atteindre un taux de consolidation optimal au sein d'une infrastructure virtuelle, le choix du processeur doit être effectué avec discernement.

Le premier élément dont il est nécessaire de tenir compte est le **type de réseau de stockage**. Le Fibre Channel (fibre optique) n'est pas aussi exigeant que le iSCSI en matière de consommation de ressources processeur. Le traitement des flux iSCSI peut nécessiter jusqu'à 10% des dites ressources (en cas d'usage intensif).

Ensuite, il est nécessaire de tenir compte des éléments mentionnés dans le tableau 6-1 pour dimensionner la **capacité processeur** :

Utilisation CPU	Nombre de VM par cœur (sans Hyper Threading)	Activation de l'Hyper Threading)
< 10/15%	3 à 5	Possible mais gains mineurs
< 25/30%	3 au maximum	non
> 30%	à examiner minutieusement	non

Tableau 6-1 : Bonnes pratiques quant au nombre de VM par cœur pouvant être déployées en fonction de l'utilisation du processeur (Source : Bonnes pratiques, planification et dimensionnement des infrastructures de stockage et de serveur en environnement virtuel de Cédric Georgeot)

Une fois les capacités processeur nécessaires identifiées (cf. section 6.1.1, Capacités nécessaires), il est important de prendre en compte les **pics d'utilisation du processeur** pour affiner les besoins. Ainsi, un processeur cadencé à 2 GHz, ayant des pics d'utilisation n'excédant pas 15%, correspond en fait à un besoin de 300 MHz. Il convient d'y ajouter une marge de 30%.

Un serveur excédant les valeurs stipulées dans le tableau ci-dessus n'est **peut-être** pas un candidat sérieux pour la virtualisation (cf. section 6.1.1, Capacités nécessaires).

Les processeurs actuels embarquent un nombre substantiel de technologies. L'une d'entre elles est l'**Hyper-Threading**. Un cœur physique peut ainsi exécuter deux tâches en parallèle en simulant l'existence de deux cœurs physiques visibles en tant que tels par le système d'exploitation.

Cet artifice ne saurait toutefois se substituer aussi efficacement à la présence de deux cœurs physiques bien réels ! En effet, en privilégiant les cœurs plutôt que les threads, le taux de consolidation sera bien plus élevé. Ainsi, l'Hyper-Threading peut même être désactivé en environnement virtuel, les gains de performances étant insignifiants. Cette technologie peut même dégrader les performances en cas de charge importante, les *threads* accédant simultanément aux mêmes ressources du cœur.

Les constructeurs avancent toutefois des gains de performances en cas d'utilisation de cette technologie, ce qui contribue à alimenter un débat que nous pourrions comparer à celui en vigueur pour le choix des formes du serveur (cf. section 6.1.2.1, Forme du serveur (ou Facteur de forme)).

Si le lecteur jugeait malgré tout intéressant d'activer l'Hyper-Threading, il pourrait tester l'apport éventuel de performance en procédant à des tests de vitesse, par le biais d'outils tels que VMware VMmark™[91] par exemple. Dans ce cas de figure, il s'agit toutefois d'affecter judicieusement les processeurs virtuels aux machines invitées. En effet, si tous les *threads* d'un même cœur sont affectés aux machines virtuelles, les autres cœurs ne seront pas sollicités. Il serait également pertinent de définir un pool de ressources processeur pour les applications les plus critiques.

Parmi les caractéristiques nombreuses des processeurs, la **compatibilité 64 bits** est absolument primordiale pour la virtualisation.

De plus, l'ISA du processeur doit comporter les instructions nécessaires à **l'assistance matérielle à la virtualisation**, s'agissait des technologies Intel® VT-x et AMD-V (cf. section 4.3.3.1, Virtualisation de l'accès au processeur). Des technologies telles qu'Intel EPT ou AMD RVI, nécessaires à l'accès aux techniques IOMMU (cf. section 4.3.3.2, Virtualisation de l'accès à la mémoire), devraient également figurer à la liste des caractéristiques du processeur. Une fonction NX Bit (Never eXecute) devrait aussi faire partie de cette liste. Cette technologie, développée par AMD® puis reprise par Intel® sous le nom de XD Bit (eXecute Disable) permet de dissocier les zones de mémoire contenant des instructions (exécutables) des zones contenant des données. Les espaces exécutables sont donc protégés, affranchissant le système des virus et chevaux de Troie utilisant les failles de dépassement de tampon. Cette caractéristique est très importante pour éviter la propagation d'infection d'une machine virtuelle à une autre.

Nous l'avons vu plus haut, le nombre de cœurs embarqués est décisif dans le calcul du nombre de VM par serveur. La fréquence associée est tout aussi importante. Certaines études ont démontré qu'une élévation de la fréquence du processeur génère un faible gain de performances en comparaison de l'augmentation importante de consommation électrique qu'elle provoque. En revanche, l'ajout de cœurs physiques d'une fréquence inférieure n'augmente que très peu la consommation électrique tout en dopant les performances. Mieux vaut dès lors privilégier des processeurs de milieu de gamme mais équipés du plus grand nombre de cœurs possibles.

Si toutefois, le lecteur souhaitait faire le choix d'une fréquence de fonctionnement élevée, il devrait s'assurer que les processeurs en question disposent d'une taille de mémoire cache L2/L3 la plus grande possible.

Un autre facteur clé dans le choix du processeur est sa **consommation électrique**. Il s'agit d'un coût caché qui mérite d'être pris au sérieux. Cette consommation étant publiée en Watts dans les fiches techniques des produits, un calcul basé sur la durée d'amortissement du produit peut s'avérer intéressant.

Un dernier point à prendre en compte dans la conception de l'infrastructure est de faire en sorte qu'**un serveur ne dépasse pas 90% d'utilisation du CPU**, une fois le nombre de

[91] http://www.vmware.com/products/vmmark/overview.html.

machines virtuelles consolidées sur ce dernier atteint. En effet, il s'agit de tenir compte des technologies de tolérance de panne proposées par les hyperviseurs.

VMware Fault Tolerance (FT) maintient en permanence deux instances d'un serveur virtuel sur deux hôtes physiques différents. VMware High Availability (HA), quant à lui, redémarre le serveur virtuel sur un autre hôte physique en cas de panne. D'autres technologies sont disponibles sur le marché, s'agissant notamment de Citrix XenMotion™ ou Microsoft Live Migration™. Elles exigent toutes que le double de ressources soit provisionné par serveur susceptible d'être configuré en haute disponibilité. C'est la raison pour laquelle l'importance des serveurs doit être correctement identifiée lors du *capacity planning*. Nous précisons à ce propos que le modèle idéal exige que tous les serveurs soient en haute disponibilité, ce qui implique des coûts associés en conséquence.

6.1.2.3 Bus et bande passante

Lorsque nous évoquons la conception d'infrastructures, une des questions récurrentes auxquelles nous avons affaire est le **goulot d'étranglement**. La puissance et l'architecture des serveurs disponibles actuellement sont si performantes que les goulots dont il est question surviennent plutôt **au niveau du stockage**. Ils apparaissent plus précisément au niveau des E/S. Les capacités de la baie de stockage, ainsi que sa configuration s'avèrent donc déterminantes (cf. section 6.1.3, Dimensionnement du stockage).

À titre d'exemples :

- Afin qu'une carte 10 Gb Ethernet puisse être exploitée correctement, un slot de type PCIe x8 est nécessaire. Si la carte en question est double port, il faut un slot PCIe x16 ;
- Si la baie de stockage permet un débit de 320 Mo/s, plusieurs liens Ethernet doivent être agrégés afin de pouvoir soutenir ce débit. Comme mentionné précédemment, il s'agit donc de connaitre le type de stockage qui sera utilisé, ainsi que le nombre de liens Ethernet nécessaires, pour pouvoir correctement choisir son serveur.

La réplication peut engendrer une certaines baisse de performances pendant les phases d'acquittement, en particulier en cas de réplication synchrone. Il convient tout de même de préciser que la baie de stockage primaire attend l'acquittement de la baie secondaire pour des raisons d'intégrité des données. Nous consentons dès lors à diminuer quelque peu les performances potentielles pour accroitre la sécurité.

Selon de type de réseau de stockage SAN, il est important de dimensionner correctement les interconnexions de type iFCP et FCIP (cf. section 7.1.2, Protocole de réplication).

En cumulant au **10 Gb Ethernet**, les **trames Jumbo** (*Jumbo frames*[92]) et des technologies telles que **VMware NetQueue™**, il est possible d'exploiter toute la puissance des interfaces correspondantes (10 Gb) et de s'affranchir ainsi des éventuels goulots d'étranglement évoqués au début de cette section.

[92] Ces trames Ethernet peuvent dépasser la longueur habituelle de 1500 octets pour atteindre une longueur allant jusqu'à 9000 octets environ.

Il convient, en dernier lieu, de rester critique envers les débits annoncés par les constructeurs qui s'apparentent bien souvent à des valeurs marketing.

6.1.2.4 Mémoire vive

Obtenir un ratio optimal de consolidation sur un serveur hôte n'a pas toujours été évident. Il y a encore quelques années, les plateformes n'offraient qu'un modèle statique d'allocation de mémoire. Par modèle statique, nous entendons qu'il était nécessaire d'estimer une certaine quantité de mémoire pour chacune des machines virtuelles puis de la leur allouer, sachant que cette quantité initiale ne pourrait être modifiée sans éteindre la machine. Il était donc impossible d'optimiser la répartition de la mémoire vive entre les différentes machines virtuelles.

Cette limitation peut heureusement être contournée grâce à certaines technologies d'optimisation, qu'il est capital de prendre en compte lors de la planification de l'infrastructure. Nous faisons en particulier référence au **Memory Overcommit™** de VMware® qui sera évoquée plus détails ci-dessous (cf. *Memory Overcommit*).

Il est également important de s'assurer que la mémoire est bien **de type ECC** afin qu'une vérification et une correction automatique d'erreur doit disponible.

Quant à la vitesse de la mémoire, elle va généralement de pair avec le choix du processeur.

Les fabricants proposent également deux types de mémoire différents, s'agissant de mémoire **avec ou sans tampon** (*Buffered / Registered memory* ou *Unbuffered / Unregistered memory*). La première embarque un registre entre la mémoire et le contrôleur mémoire. Il faut être conscient que si la mémoire sans tampon permet d'atteindre des performances de haut niveau, son prix est également élevé. La mémoire avec tampon offre, quant à elle, une grande stabilité, ainsi qu'une propension élevée à l'évolutivité, à des coûts raisonnables.

Pour dimensionner correctement la quantité totale de mémoire d'un serveur hôte, il faut additionner les besoins identifiés à l'aide du *capacity planning*, tout en tenant compte des pics.

En tenant comptes des éléments mentionnés dans le tableau 6-2, nous voyons qu'il est inutile de provisionner 28 Go de mémoire, car nous voyons que cette dernière n'est sollicitée qu'à hauteur de 12 Go avec des pics d'excédant pas 17 Go.

Serveurs	Mémoire disponible	Consommation moyenne	Consommation lors de pics
Windows® 2003	8 Go	4 Go	6 Go
Linux™ 2.6	4 Go	2 Go	3 Go
Windows® 2008 R2	12 Go	6 Go	8 Go
Total	**28 Go**	**12 Go**	**17 Go**

Tableau 6-2 : Capacité mémoire disponible, ainsi que consommation moyenne et lors des pics (Source : inspiré de Bonnes pratiques, planification et dimensionnement des infrastructures de stockage et de serveur en environnement virtuel de Cédric Georgeot)

Il convient tout de même d'ajouter 1 Go par tranche de 10 Go de pic mémoire, et ce, pour éviter une sollicitation trop importante du *swap*.

Enfin, il est capital de tenir compte de la **consommation de l'hyperviseur** lui-même et de l'ajouter à celle des pics mémoire précédemment établie. Nous retenons généralement, à titre d'exemple, une valeur de 2 Go pour VMware ESX™, 640 Mo pour son successeur ESX(i) et 1 Go pour Hyper-V. Ces valeurs doivent être vérifiées avec tout éditeur auquel nous pourrions avoir affaire.

Aussi, dans le cas où nous aurions 40 serveurs à virtualiser par le biais de VMware ESX™, exposant un total de pic mémoire de 60 Go, le calcul serait le suivant : 60 Go + 2 Go pour l'hyperviseur + 6 Go = 68 Go au total.

Ce calcul est adapté pour des machines virtuelles obtenues à partir de conversion P2V (cf. section 6.1.2.6, Conversion P2V). Si les VM ne sont pas basées sur des machines physiques et que, de ce fait, elles sont créées directement par le biais de l'hyperviseur, la sur-allocation mémoire permettra finalement de leur allouer plus de mémoire qu'en dispose réellement le serveur hôte physique (cf. *Memory Overcommit*).

Memory Overcommit

VMware® pratique la sur-allocation de mémoire en cumulant différentes technologies, s'agissant du ***Ballooning***, de la compression mémoire **(*Memory Compression*)** et du ***Transparent Page Sharing*** (TPS), appuyées par des algorithmes optimisés au niveau du noyau de l'hyperviseur et du ***swapping*** effectué par ce dernier.

Si nous nous référons à la technologie de VMware®, c'est que la gestion de la sur-allocation de la mémoire telle que proposée par cette société s'avère particulièrement aboutie et peut être considérée comme une référence en la matière. Notre but étant d'attirer l'attention du lecteur sur ce type de technologie et non pas de l'orienter vers VMware®. Si ce dernier décidait d'opter pour un autre éditeur, il serait dès lors à même d'évaluer la pertinence des techniques proposées par ce dernier pour mettre en œuvre la sur-allocation mémoire.

Le ***ballooning*** est à mettre en relation avec le fait que le système d'exploitation invité fonctionne de manière totalement isolée. Ainsi, il n'est pas conscient du fait qu'il évolue au sein d'une machine virtuelle, au même titre qu'il n'est pas conscient de l'état des autres machines virtuelles hébergées sur le même hôte physique. Lorsque ce dernier exécute plusieurs machines virtuelles et commence à être à court de mémoire, aucune des machines invitées ne peut libérer de la mémoire puisqu'elles ne disposent d'aucun moyen qui leur permette de détecter ce manque. C'est donc le *balloon driver* qui va permettre aux VM de prendre conscience du manque de mémoire en question.

Ce pilote est installé sur la machine virtuelle par le biais des VMware Tools™[93]. Ce pilote ne possède pas d'interface avec le système invité. Il communique avec l'hyperviseur par le biais d'un canal privé. Quand ESX™ ou ESX(i)™ désire récupérer de la mémoire précédemment allouée à une machine virtuelle, il fait en sorte que le *balloon driver* gonfle dans la mémoire de cette dernière pour y occuper un certain espace (par défaut, le pilote peut récupérer

[93] Suite de pilotes qui doivent être installés sur toute machine virtuelle d'un environnement virtualisé VMware®.

jusqu'à 65% de la mémoire allouée à la machine virtuelle, la valeur pouvant être définie à l'aide de la commande **Mem.CtlMaxPercent**). Comme ce pilote est exécuté au niveau de la VM cible, c'est donc le système d'exploitation invité qui gère l'effet du *balloon* driver à l'aide de ses propres algorithmes de gestion de la mémoire. Il cède ainsi d'abord la mémoire qu'il n'utilise pas puis, si besoin est, déchargera une partie du contenu de cette dernière dans son propre *swap*.

La figure 6-1 illustre les principes décrits ci-dessus.

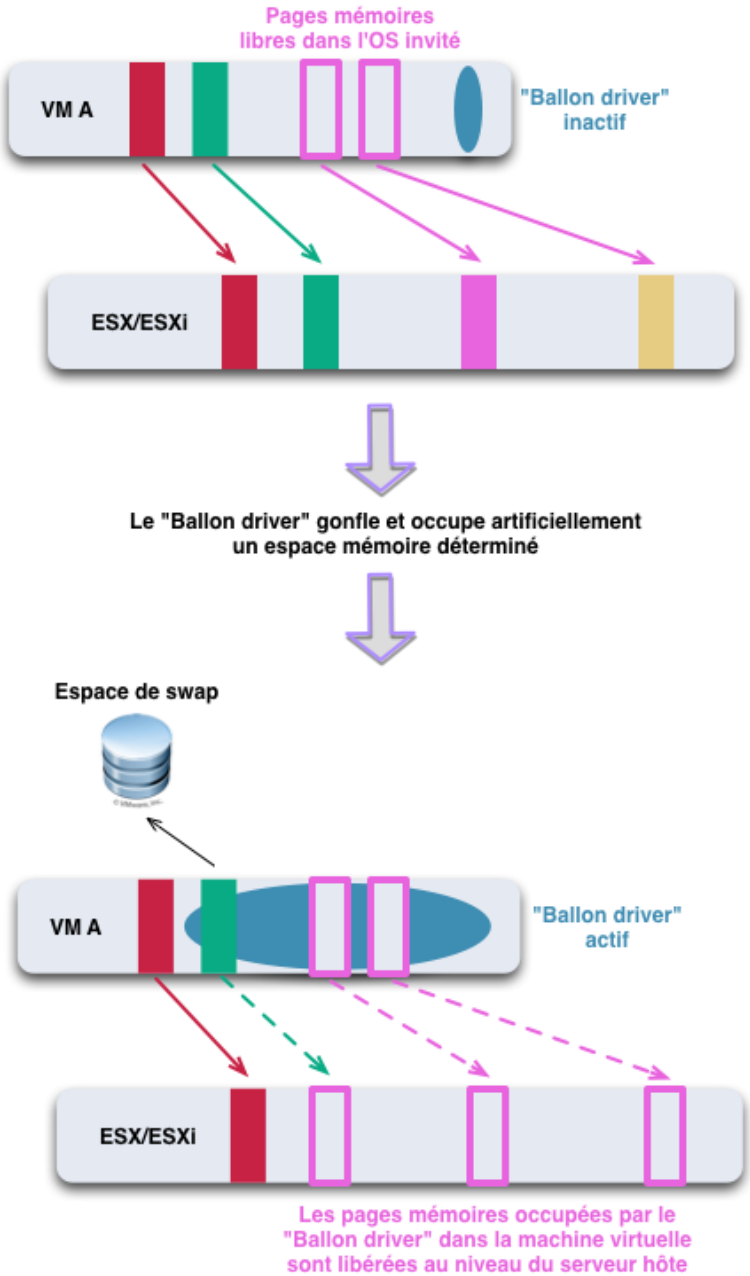


Figure 6-1 : Principe de fonctionnement du *balloon driver* (Source : http://wattmil.dyndns.org/vmware/19-gestionmemoiresousesx4?start=2)

VMware® a comparé les performances obtenues par le *ballooning* en comparaison avec celles obtenues à l'aide du *swapping* effectué par l'hyperviseur lors d'expériences menées

sur le sujet[94]. Il a été ainsi démontré que le *ballooning* provoque bien moins de dégradation de performance que le *swapping*.

L'idée de *memory compression* est relativement simple. Si les pages sensées être « swappées » sont compressées et stockées dans un cache situé dans la mémoire principale, le prochain accès à une de ces pages provoquera simplement sa décompression. L'accès est donc infiniment plus rapide que lors d'un accès disque. Tant que le cache en question n'est pas plein, peu de pages ont besoin d'être « swappées ». Cette technique permet donc, tout comme le *ballooning*, d'éviter un usage trop intensif du *swapping* par le biais de l'hyperviseur. Les performances des applications sont évidemment bien meilleures lorsque l'hôte est soumis à une intense pression au niveau de la mémoire.

La figure 6-2 illustre le mécanisme de compression mémoire par rapport au *swapping*. Nous constatons qu'aucun accès disque n'est nécessaire dans le cas de la compression mémoire, ce qui la rend beaucoup plus rapide. Nous voyons que les pages A et B sont compressées et stockées dans le cache comme deux demi-pages.

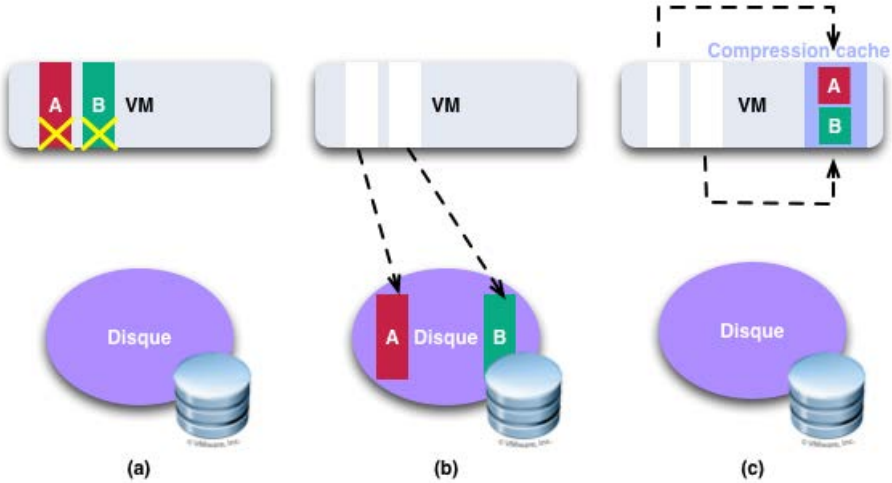


Figure 6-2 : Comparaison entre compression mémoire et *swapping* (Source : http://www.vmware.com/files/pdf/techpaper/vsp_41_perf_memory_mgmt.pdf)

Quant à la technique du ***Transparent Page Sharing***, elle consiste à partager les pages mémoires identiques qu'on peut trouver sur différentes machines virtuelles hébergées. L'hyperviseur utilise le hachage (*hashing*) pour identifier les pages potentiellement candidates au partage. Les clés ainsi obtenues sont stockées dans une table ad hoc. Ces clés sont périodiquement comparées et si certaines sont identiques, une comparaison bit à bit est effectuée. Ainsi, l'hyperviseur est en mesure de détecter les pages identiques et de faire en sorte que ces dernières ne soient enregistrées qu'une fois en mémoire. Aussi, une page mémoire physique peut être adressée dans l'espace mémoire virtuel de plusieurs VM. Dans le cas où l'une de ces dernières tenterait de modifier une page mémoire partagée, l'hyperviseur en créerait une copie privée que seule la machine virtuelle concernée serait en mesure d'adresser.

[94] Cf. section 5.3 du document disponible à l'URL suivante : http://www.vmware.com/files/pdf/techpaper/vsp_41_perf_memory_mgmt.pdf.

Le principe de fonctionnement du TPS est illustré par la figure 6-3.

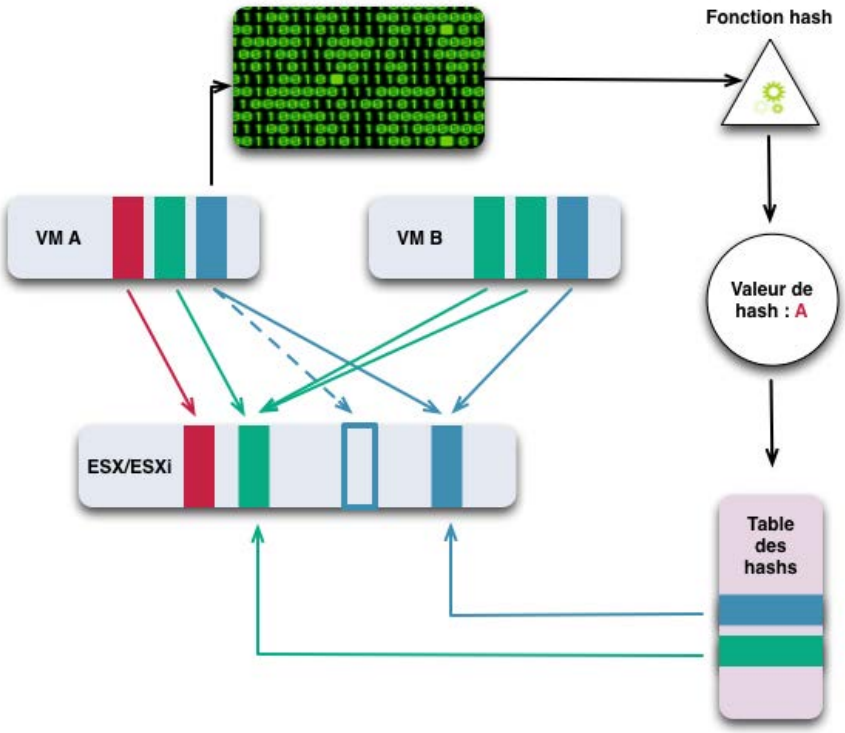


Figure 6-3 : Principe de fonctionnement du *Transparent Page Sharing* (Source : http://wattmil.dyndns.org/vmware/19-gestionmemoiresousesx4?start=1)

Ce mécanisme permet dès lors d'économiser de l'espace mémoire et est, à ce titre, extrêmement important. Si les machines virtuelles sont dotées de la même version de système d'exploitation, de nombreuses pages seront ainsi identiques.

6.1.2.5 Interface réseau

Quoiqu'un hyperviseur puisse fonctionner avec une seule interface réseau, il est fortement recommandé de scinder les services et chercher à mettre en place une redondance. Ceci est d'autant plus vrai, lorsque des technologies de migration à chaud de machines virtuelles en cours d'exécution, telles que Microsoft Live Migration™ ou VMware vMotion™, sont utilisées. Si le nombre de VM exécutées est de surcroit important, l'existence de plusieurs liens est essentielle. Il serait même judicieux d'opter pour des interfaces 10 Gbit/s dans ce cas de figure.

Pour estimer le nombre de machines virtuelles qu'une carte réseau Gigabit est susceptible de supporter, il convient de se baser sur les taux d'utilisation du réseau des serveurs qui doivent être virtualisés. En additionnant ces valeurs, il est possible de mettre en évidence la bande passante qui sera réellement consommée par les VM, une fois ces dernières en production.

Le trafic généré est relativement modeste en moyenne. Les applications lourdes doivent toutefois faire l'objet d'une attention particulière. Nous pouvons tout de même partir du principe qu'il est en général possible d'héberger de 6 à 10 machines virtuelles par carte réseau Gigabit.

Les tableaux 6-3 et 6-4 ci-dessous mettent en évidence les configurations idéales recommandées par certains éditeurs importants (en l'occurrence Microsoft® et son hyperviseur Hyper-V™ et VMware® avec vSphere™). Nous précisons que s'ils ne référencent pas la totalité des cas de figure, ils permettent toutefois de se faire une certaine idée.

Les règles suivantes doivent tout de même être prises en compte pour mettre en place un dimensionnement correct du nombre d'interfaces réseau nécessaires :

- Prévoir un service par interface – soit une ou plusieurs carte(s) pour la production, une carte pour la gestion, une carte pour le service de migration à chaud, etc. ;
- Une ou plusieurs cartes dédiées pour isoler le trafic iSCSI des autres services ;
- Regrouper sur la même interface plusieurs services peut être envisagé, à condition que les services en question ne nécessitent qu'une faible bande passante. Il peut s'agir de la gestion et de la production par exemple ;
- Créer des sous-réseaux par service et mettre en place différents VLAN au niveau du commutateur pour isoler les trafics ;
- Veiller à ce que les pilotes soient à jour.

Microsoft Hyper-V™		
Type de réseau	**Nombre minimum de cartes**	**Nombre maximum de cartes**
Production	2 x 1 Gb	4 x 1 Gb
iSCSI	2 x 1 Gb	4 x 1 Gb
CSV[95] / Heartbeat[96]	1 x 1 Gb	2 x 1 Gb
Live Migration	1 x 1 Gb	2 x 1 Gb
Gestion	1 x 1 Gb	2 x 1 Gb

Tableau 6-3 : Nombre de cartes nécessaire pour Hyper-V™ (Source : Bonnes pratiques, planification et dimensionnement des infrastructures de stockage et de serveur en environnement virtuel par Cédric Georgeot).

VMware vSphere™		
Type de réseau	**Nombre minimum de cartes**	**Nombre maximum de cartes**
Production	2 x 1 Gb	4 x 1 Gb
iSCSI	2 x 1 Gb	4 x 1 Gb
Service Console	1 x 1 Gb	1 x 1 Gb
Fault Tolerance[97] High Availability	1 x 1 Gb	4 x 1 Gb
vMotion™	1 x 1 Gb	4 x 1 Gb
Gestion	1 x 1 Gb	1 x 1 Gb

Tableau 6-4 : Nombre de cartes nécessaire pour vSphere™ (Source : Bonnes pratiques, planification et dimensionnement des infrastructures de stockage et de serveur en environnement virtuel par Cédric Georgeot).

[95] Cluser Shared Volume ou volume de stockage partagé. Ces volumes sont visibles simultanément par tous les nœuds d'un cluster de serveurs hôte. Ces derniers peuvent ainsi accéder à tous les disques virtuels, ce qui autorise la mise en œuvre de Live Migration, la technologie de migration à chaud d'Hyper-V™.

[96] Solution de monitoring permettant à Hyper-V™ de mettre en œuvre la haute disponibilité (Live Migration).

[97] Technologie assurant la disponibilité permanente des applications en cas de panne des serveurs hôte, en créant une instance fantôme active d'une machine virtuelle représentant le pendant virtuel de l'instance principale et en permettant le basculement instantané entre les deux instances.

Les éditeurs accompagnent volontiers leurs clients en les conseillant à ce propos. Microsoft® publie par exemple un guide de configuration du réseau en cas d'utilisation de Live Migration. Ce dernier est disponible à l'URL suivante : http://technet.microsoft.com/en-us/library/ff428137(WS.10).aspx.

6.1.2.6 Conversion P2V

La phase de virtualisation des serveurs physiques va immanquablement entrainer une migration. Cette dernière peut être grandement simplifiée par l'utilisation d'outils prévus à cet effet qui vont réaliser l'opération de **P2V** (*Physical to Virtual*).

Un certain nombre de ces outils sont disponibles sur le marché, s'agissant notamment de :

- VMware **vCenter Converter**™[98] ;
- Microsoft **System Center Virtual Machine Manager**™[99] ;
- SYSInternals **Disk2VHD**™ ;
- Acronis **Backup & Recovery**™[100] ;
- NetIQ **PlateSpin Migrate**™[101].

Cette liste n'est en aucun cas exhaustive. Certains outils sont automatisés, semi-automatisés ou doivent être pilotés à la main. En effet, certains d'entre eux peuvent planifier des opérations de migration la nuit, d'autres vont nécessiter un arrêt du serveur et un démarrage sur un live CD spécialisé, etc.

Il est également possible de réaliser, à l'aide de ces solutions, des opérations **V2V** (Virtual to Virtual) lors du changement d'un hyperviseur pour celui d'un autre éditeur.

Il faut savoir que certains types de serveurs physiques ne peuvent pas être convertis en serveurs virtuels. Le rôle Active Directory ne supporte pas d'être migré à l'aide d'une opération P2V car le système d'exploitation contrôle en permanence l'intégralité de la base. Une corruption majeure peut ainsi survenir. La seule solution serait dans ce cas d'effectuer la conversion avec le serveur éteint, ce qui se prête assez peu à un tel rôle. En conclusion, certains rôles doivent faire l'objet de tests avant la mise en production.

Parmi les bonnes pratiques en matière de conversion P2V, nous suggérons de désinstaller tous les outils des constructeurs installés avec le serveur, tels que les outils de gestion du RAID et d'arrêter temporairement l'anti-virus. Suivant le type de disque, il peut être judicieux d'effectuer une défragmentation et de vérifier le système disque. L'arrêt des services liés au matériel est également préconisé.

Il faut être également conscient qu'après une migration P2V, les pilotes matériel doivent être désinstallés et les outils d'intégration de l'éditeur de la solution de virtualisation installés (par exemple, les VMware Tools™).

[98] http://www.vmware.com/fr/products/datacenter-virtualization/converter/overview.html.
[99] http://www.microsoft.com/france/serveur-cloud/system-center/default.aspx.
[100] http://www.acronis.com/backup-recovery/smallbusiness.html#ABR11-5SW.
[101] https://www.netiq.com/products/migrate/.

En conclusion, une telle opération pouvant être délicate, il convient de la planifier rigoureusement.

6.1.3 Dimensionnement du stockage

Cette section vise principalement à éclairer le lecteur sur le SAN, ce type de stockage étant généralement plébiscité en environnement virtuel, mais également au sein des infrastructures classiques.

Le stockage au sein des serveurs n'a pas été abordé à la section 6.1.2, ces derniers n'étant destiné qu'à héberger l'hyperviseur qui n'exige qu'un espace relativement restreint. D'après une comparaison[102] effectuée par VMware®, portant sur vSphere™[103] et l'offre de la concurrence en la matière d'hyperviseur, l'encombrement du VMM sur le disque se monte à :

- 144 Mo pour vSphere ESX(i)™ ;
- 3 Go pour Hyper-V™ si uniquement l'installation de base de Windows Server™ 2008 R2 SP1 a été faite et 10 Go si l'installation complète du même système d'exploitation a été mise en oeuvre ;
- 1 Go pour XenServer™ 5.6 FP1.

Ainsi, un serveur équipé de deux disques durs de type SAS, cadencés à 15'000 tours/minute (rapides), configurés en RAID 1 et de taille tout à fait standard, s'avèrerait parfaitement suffisant pour accueillir un hyperviseur.

Il est également possible de démarrer un hyperviseur qui aurait été déployé sur le SAN (*Boot on SAN*). Cette manière de faire demeure relativement complexe, en particulier lorsqu'on la compare avec la facilité de mise en œuvre de l'hyperviseur sur les serveurs hôte. Quant au démarrage à partir d'une clé USB ou d'une carte SD, nous le déconseillons. En effet, même s'il est tout à fait possible de procéder de la sorte (les constructeurs proposant des connecteurs USB ou SD disponibles au niveau de la carte mère du serveur hôte), la faible performance au niveau des cycles lecture/écriture et le manque de fiabilité sur le long terme de ce type de média ne plaide pas pour cette manière de faire.

Il peut être judicieux de procéder à une analyse précise des données lors de la phase de migration de ces dernières sur le SAN. Les données pourront être ainsi segmentées en fonction de leur degré de criticité. De telles mesures permettent généralement de réduire les volumes nécessaires, d'ajuster les délais de sauvegarde et de restauration, etc., et de réaliser ainsi des économies.

La mise en œuvre du *tiering* (cf. section 5.4.5, Fonctionnalités avancées) ne manquera pas d'être facilitée par une telle analyse.

6.1.3.1 Espace disque

Même si les baies de stockage actuelles disposent de fonctionnalités les rendant extrêmement flexibles, s'agissant notamment de la déduplication (cf. section 7.1.4, Déduplication) ou du Thin Provisioning (cf. section 5.4.5, Fonctionnalités avancées), il est

[102] http://www.vmware.com/fr/products/datacenter-virtualization/vsphere/esxi-and-esx/compare.html.
[103] L'hyperviseur de VMware®.

judicieux de prévoir un espace disque plus important que celui disponible sur les différents serveurs appelés à être virtualisés. Il est ainsi recommandé de prévoir au moins 20% d'espace disque supplémentaire afin d'anticiper la croissance potentielle des données.

De plus, il s'agit de tenir compte de l'existence de la fonctionnalité permettant de réaliser des instantanés (*snapshot*). Comme nous l'avons précédemment évoqué, cette technique est passablement utilisée en environnement virtualisé. Les solutions de sauvegarde intégrées aux hyperviseurs, comme VMware vSphere Data Protection™ (cf. section 3.1.8, Optimisation de la sauvegarde), la création de modèle de machine virtuelle ou le clonage de machine virtuelle, reposent sur cette technologie.

Il faut donc tenir compte de l'espace nécessaire au stockage, même provisoire, de ces instantanés. Les fabricants sont à même de fournir les renseignements y relatifs, ces derniers pouvant varier en fonction des solutions de virtualisation ou de stockage concernées. Nous recommandons toutefois de créer une réserve correspondant à 20% de la taille des différents volumes, afin de pouvoir utiliser cette fonctionnalité en évitant tout échec potentiel. Il s'agit également de tenir compte du taux de changement *(*ROC pour *Rate of Change* en anglais) des instantanés et de la fréquence à laquelle ces derniers sont mis à jour.

Considérons, à titre d'exemple, le cas suivant :

- 100 Go de données ;
- 10% de ROC ;
- 10 jours de rétention ;
- 1 instantané par jour ;

Nous obtenons donc un total de 240 Go, soit 100 Go + 100 Go comprenant le ROC multiplié par 10 jours à raison d'un instantané par jour + 20 Go correspondant à la réserve prévue pour les instantanés + 20 Go pour la croissance estimée.

6.1.3.2 Contrôleur de stockage

Selon les constructeurs et les modèles de baies choisis, il est possible d'opter pour la fibre optique (Fibre Channel) ou l'iSCSI. La taille de la mémoire cache peut varier considérablement, ainsi que le nombre de contrôleurs.

Même si la baie peut fonctionner avec un seul contrôleur, il n'est pas envisageable de garantir la haute disponibilité de cette dernière sans disposer d'au moins deux contrôleurs. Ces contrôleurs peuvent être configurés en mode **actif/actif** ou **actif/passif**. Les cas de figure où il convient de mettre en œuvre l'une ou l'autre des configurations sont décrits ci-dessous.

6.1.3.2.1 Configuration actif/passif

Dans ce cas de figure, les deux contrôleurs fonctionnent en mode redondant avec leurs caches mis en miroir.

Dans le cas où un contrôleur viendrait à tomber en panne, son cache serait automatiquement commuté.

Chaque contrôleur dispose de ses propres LUN. Un contrôleur bénéficie de l'accès primaire aux LUN, alors que l'autre est défini comme étant le contrôleur secondaire.

Cependant, les LUN disponibles sur la baie ne sont accessibles par le biais de chemins alternatifs qu'en cas de changement d'appartenance entre les contrôleurs, chacun de ces derniers disposant de ses propres LUN. Les performances peuvent ainsi être fortement dégradées en cas de panne d'un des deux contrôleurs.

Dans la figure 6-4 ci-dessous, l'accès aux LUN s'effectue par le biais du contrôleur A. Le contrôleur B est prévu pour entrer en fonction en cas de défaillance du contrôleur A. Si uniquement un port du contrôleur A est défaillant, l'accès se fera par le biais du commutateur B ou via un autre port, mais en aucun cas par le contrôleur B.

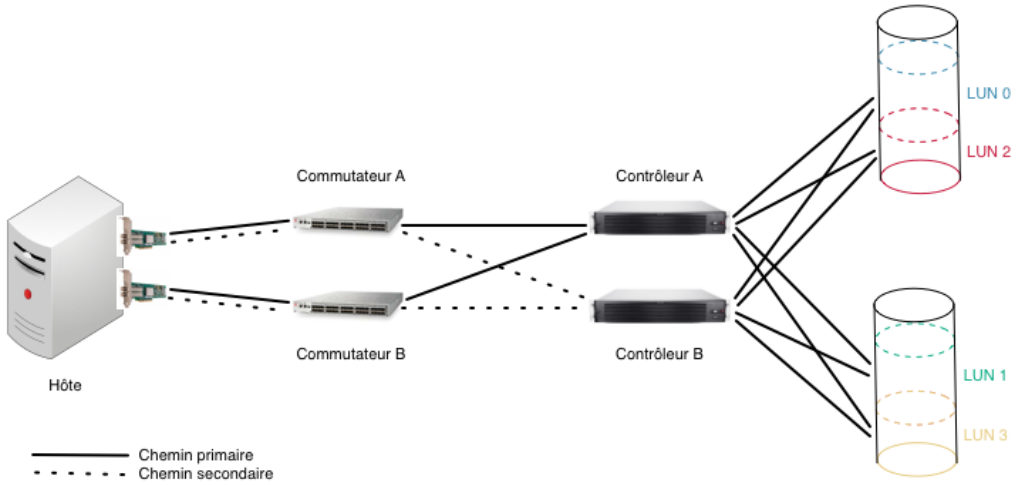


Figure 6-4 : Exemple de configuration actif/passif (Source : Bonnes pratiques, planification et dimensionnement des infrastructures de stockage et de serveur en environnement virtuel de Cédric Georgeot)

6.1.3.2.2 Configuration actif/actif

En mode actif/actif, les deux contrôleurs fonctionnent parallèlement. Les performances sont donc nettement accrues. Il faut tout de même considérer qu'en cas de panne d'un des deux contrôleurs, seul le contrôleur restant est à même de traiter les flux. Les performances seront dès lors fortement impactées.

Des techniques comme le *Multipathing* (existence de plusieurs liens physiques) et le *Load Balancing* (équilibrage des charges) doivent donc être mises en œuvre pour pallier les désagréments causés par une telle situation.

La configuration du cache exerce un impact important au sein d'une telle configuration (cf. section 6.1.3.3, Mémoire cache).

Nous précisons que dans ce cas de figure, les LUN sont visibles par tous les contrôleurs.

Pour pouvoir mettre en œuvre le *Multipathing*, il convient d'équiper le serveur d'une carte FC double munie de deux ports ou de deux cartes FC distinctes.

Dans un stockage de type Fibre Channel, les ports d'interconnexion agissent comme un commutateur interne à des fins de redondance. Si la disponibilité des données prime sur la performance, l'interconnexion de ports peut ainsi être activée afin de faire en sorte que les deux contrôleurs soient « virtuellement » reliés. Le serveur aura alors accès aux volumes par le biais de n'importe lequel des deux contrôleurs. L'acquisition de commutateur FC externe peut être dès lors évitée.

Si l'un des contrôleurs venait à faillir, l'interconnexion de ports autoriserait tout de même l'accès continu aux données, sans qu'une intervention d'urgence doive être prévue. La figure 6-5 illustre d'ailleurs ce cas de figure. Comme nous pouvons le voir, l'interconnexion de ports est nécessaire dans le cas où un attachement direct est préconisé (sans passer par un commutateur FC). Il est évident que si la baie ne dispose que d'un contrôleur, cette dernière ne sera pas compatible avec une telle fonctionnalité.

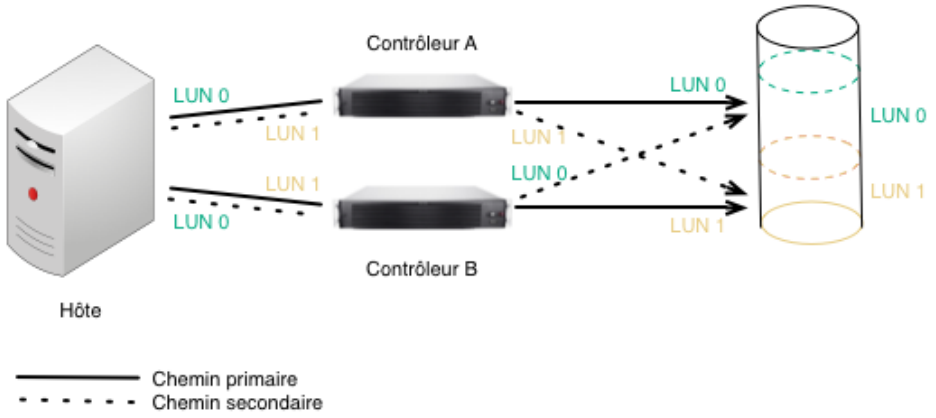


Figure 6-5 : Attachement direct au stockage avec mise en œuvre de l'interconnexion de ports (Source : Bonnes pratiques, planification et dimensionnement des infrastructures de stockage et de serveur en environnement virtuel de Cédric Georgeot)

Lorsque l'interconnexion de ports n'est pas activée, les données appartiennent à l'un des contrôleurs. Il s'agit en quelque sorte d'une connexion point par point. Ce genre de configuration s'apparente à une topologie basée sur deux contrôleurs connectés à un ou plusieurs commutateurs FC, tel qu'illustré par la figure 6-6 ci-dessous.

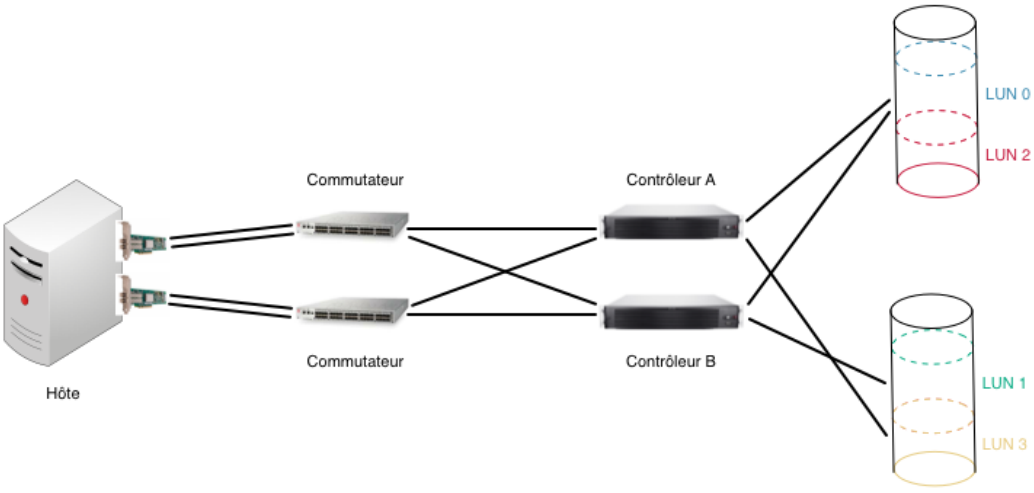


Figure 6-6 : Topologie basée sur deux commutateurs FC (Source : Bonnes pratiques, planification et dimensionnement des infrastructures de stockage et de serveur en environnement virtuel de Cédric Georgeot)

Ici, le serveur connait la totalité des chemins menant à chaque LUN. Un LUN est atteignable depuis n'importe lequel des HBA. Quant aux contrôleurs, ils accèdent simultanément à ce LUN.

6.1.3.3 Mémoire cache

Comme nous l'avons déjà évoqué précédemment, la mémoire cache (antémémoire) revêt une importance toute particulière au sein d'un SAN. Le choix des contrôleurs dépend notamment de cette dernière. Il va sans dire qu'un cache embarquant une quantité importante de mémoire est extrêmement recommandé.

Les caches peuvent fonctionner de manière différente. Voici leurs différents principes de gestion. :

- Antémémoire à **écriture directe** (*Write Through*). Les données présentes dans le cache sont immédiatement écrites vers les disques durs de la baie. Ce type de cache est destiné aux applications nécessitant un état permanent de consistance (par exemple, les bases de données) ;
- Antémémoire à **écriture différée** (*Write Back*). Les données présentes dans le cache sont écrites de manière différée vers les disques durs de la baie. Les données en lecture sont délivrées par le cache alors que les données en écriture sont différées, avec un accroissement de performance à la clé. Ce procédé tient compte du fait que le cache est bien plus rapide qu'une pile RAID et que les données sont écrites bien plus lentement qu'elles sont lues. Il est impératif de disposer d'une batterie à coupler au contrôleur dans ce cas de figure. Toute perte de données sera ainsi évitée en cas de coupure de l'alimentation électrique ;
- Antémémoire à **lecture anticipée** (*Ahead Cache Settings*). Les lectures au niveau du stockage peuvent être anticipées à l'aide de ces technologies. Ce mode de gestion peut être pratique mais il ne peut toutefois pas « prédire » les données qui vont être lues. Ce mode est ainsi adapté pour des accès séquentiels conséquents, tels que le streaming vidéo ou l'accès à des fichiers de taille conséquente.

Le mode *Write Back* améliore les performances tandis que le mode *Write Through* est recommandé par certains éditeurs/constructeurs pour obtenir des instantanés (*snapshots*) consistants. En effet, les données présentes en cache lors de prise de l'instantané peuvent nuire à cette consistance.

Par ailleurs, il est important de considérer l'achat d'onduleurs intelligents et adaptés à l'infrastructure pour sécuriser cette dernière. Les serveurs et les baies de stockage doivent en effet être proprement éteints en cas de panne électrique afin que les caches puissent être entièrement vidés.

En dernier lieu, adapter le type de gestion du cache en fonction du niveau de RAID peut être judicieux. En effet, en cas de pile RAID 0, 1 ou 10, le Write Through est adapté pour les accès séquentiel alors que le Write Back l'est plutôt pour les accès aléatoire. S'il s'agit d'une pile RAID 5 ou 6, le Write Back sera le mieux adapté.

6.1.3.4 Types de disque dur

Il est impératif de mettre en exergue les différentes contraintes auxquelles l'infrastructure doit répondre pour faire le bon choix entre les différentes technologies de disque dur disponibles. La performance n'est pas l'unique contrainte dont il faille tenir compte. La disponibilité du stockage et les exigences relatives aux applications sont également des notions très importantes.

À l'heure actuelle, les types de disques qui font l'unanimité sont les **SAS** (*Serial Attached SCSI*), les **NL-SAS** (Near Line SAS) et les **SATA** (Serial ATA). Les disques **Fibre Channel** sont également une option dont il faut tenir compte, au même titre que les **SSD** (Solid State Drive) et à moindre mesure les disques **hybrides**.

Par ailleurs Il n'est pas rare d'opter pour différents types de disques et de les combiner au sein de la même baie. Comme nous l'avons déjà évoqué à la section 5.4.5, les solutions de *tiering* permettent d'exploiter au mieux les types de disques disponibles sur le marché en fonction des contraintes relatives au métier (criticité des données, exigences en matière d'E/S, etc.).

Les performances offertes par les différents types de disques (à l'exception des disques SSD) peuvent varier en fonction de la capacité de ces derniers. En effet, un disque de petite capacité augmente le nombre d'axes et améliore ainsi les performances, en particulier lors d'accès aléatoires aux données. Ce n'est pas le cas des disques de grande capacité. Lors d'accès séquentiels en lecture ou en écriture, une différence de l'ordre de 6% en moyenne peut être observée.

6.1.3.4.1 SAS

Le SAS a dorénavant supplanté l'ancien bus SCSI pour devenir le standard en qualité de stockage de niveau entreprise. Il dépasse bien évidemment les limites en termes de performances qu'offrait le SCSI grâce à l'apport du mode de transmission de données en série de l'interface SATA.

L'arrivée du SAS a bouleversé le marché des interfaces de disques qui jusqu'alors était partagé entre deux réalités : la norme ATA qui se trouve majoritairement au sein des postes de travail, et ce, en raison de son coût peu élevé, et le SCSI, très présent au sein des centres de données car plus rapide et plus performant (en particulier au niveau des applications multi-utilisateurs ou la gestion des piles de disque RAID).

Ces deux interfaces, nées dans les années 80, reposent sur un mode de transmission parallèle. Alors que les débits observés avec ce mode de transmission n'ont jamais excédé 2.56 Gbit/s (Ultra 320 SCSI), ils atteignent d'ores et déjà 6 Gbit/s avec le mode de transmission série propre au SAS (pour arriver prochainement à 12 Gbit/s). Il ne s'agit d'ailleurs que d'un seul des nombreux avantages offerts par ce mode de transmission.

En outre, les débits fournis par le SAS sont exclusifs. Ainsi, chaque disque dispose d'un débit de 6 Gbit/s alors que dans le cas du SCSI, la bande passante de 2.56 Gbit/s est répartie entre tous les périphériques du contrôleur. Quant à la limitation de 15 disques par

contrôleur en vigueur avec le SCSI, elle est maintenant de 128 disques par connexion pour le SAS.

Les câbles d'interface sont communs entre le SAS et le SATA et le SAS peut cohabiter avec le SATA au sein d'une même grappe de stockage.

Par ailleurs, ces disques sont plus fiables que les disques NL-SAS et SATA et restent performants bien qu'ils soient soumis à des conditions difficiles. Leurs performances sont en outre bien supérieures à celles offertes par les disques NL-SAS et SATA.

En termes de fiabilité, le BER (*Bit Error Rate*) du disque SAS équivaut à 1 sur 10^16 bits, soit 1 bit d'erreur tous les 10'000'000'000'000'000 bits. En comparaison, le BER du disque SATA se monte à 1 sur 10^15 bits, ce qui ne signifie pas pour autant que ce dernier ne soit pas fiable.

De plus, le temps écoulé entre deux erreurs se monte à 16 millions d'heures pour le SAS alors qu'il est de 12 millions d'heures pour le SATA, ce qui correspond à 182 ans pour le SAS contre 136 pour le SATA. Ces chiffres constituent toutefois des moyennes. Ils permettent simplement de mettre en évidence le niveau de qualité des disques SAS.

Les disques/contrôleur SAS offrent également une multitude de commandes additionnelles prévues pour contrôler le disque qui rendent le SAS bien plus efficace que le SATA.

En résumé, seuls les disques Fibre Channel peuvent entrer en comparaison avec les disques SAS, tant en termes de performances – généralement du plus haut niveau si l'on fait exception du SSD dont la technologie est basée sur la mémoire flash – qu'au niveau du coût élevé.

Ces disques atteignent une vitesse de rotation de 15'000 t/m à l'heure actuelle, ce qui permet un accès plus rapide aux données que celui offert par les disques NL-SAS ou SATA. Le fait que les instructions soient pré-triées pour améliorer la performance en cas de nombreux petits accès et l'existence d'une gestion bidirectionnelle des flux sont d'autres raisons qui font de ses disques un choix optimal pour les applications les plus exigeantes.

À moins que l'accès à des disques SSD soit financièrement possible, **les disques SAS ou FC 15'000 t/m doivent être privilégiés pour le *tier* 1** au sein du SAN. Si le *tier 1* est composé de disques SSD, ces derniers composeront dès lors le *tier 2*.

6.1.3.4.2 NL-SAS

Le disque NL-SAS est en quelque sorte une fusion entre un disque SAS et un disque SATA, puisque qu'il s'agit d'un disque SATA de niveau entreprise monté sur un contrôleur SAS. En d'autres termes, il est comparable à un disque SATA, notamment en matière de vitesse de rotation ou de têtes de lecture, mais il dispose du jeu de commandes natives du SAS.

Ce type de disque est donc capable de fournir toutes les fonctionnalités de niveau entreprise nécessaires, telles que les canaux de données simultanés ou le support pour hôtes multiples, sans que sa vitesse de rotation n'atteigne celle du SAS.

À ce titre, il est capable de gérer deux lectures ou deux écritures simultanément ou une lecture/écriture simultanées, contrairement au SATA.

Il ne faut donc pas être trompé par la mention SAS dans l'appellation de ce type de disque, sa fiabilité étant similaire à celle du SATA. Il ne faut donc pas envisager l'achat de disque NL-SAS pour obtenir une fiabilité accrue par rapport au SATA.

Nous précisons à ce propos que là où le NL-SAS atteint généralement 7'200 t/m, le SATA peut parfois atteindre les 10'000 t/m.

L'avantage de NL-SAS est donc sa connectique de type SAS pour un prix plus abordable qu'un disque SAS, tout en offrant une capacité plus importante que ce dernier. Ses performances sont également meilleures que celle offertes par un disque SATA.

Ce type de disque convient parfaitement **pour le *tier 2*, au même titre qu'un disque SAS 10'000 t/m**, à moins que ce niveau soit déjà composé de disques SAS 15'000 t/m. Le cas échéant, les NL-SAS ou les SAS 10'000 t/m formeront le *tier 3*.

6.1.3.4.3 SATA

Le disque SATA ne dispose pas des avantages inhérents au SAS, voire au NL-SAS. Il constitue tout de même un élément vital de tout système de stockage, tout particulièrement pour le **stockage de tier 3** ou le **stockage de masse**.

La métrique principale quant à l'achat de disque SATA est donc le coût par To, qui doit être le plus bas possible alors que celles liées au disque SAS porteront sur la performance (optimale grâce à la vitesse de rotation de 10'000 ou 15'000 t/m qui autorise un nombre significatif d'IOPS par disque physique).

6.1.3.4.4 Les disques hybrides et SSD

Les disques SSD (Solid State Drive) sont composés de mémoire flash en lieu et place des traditionnels plateaux et têtes de lecture. Ils ne contiennent dès lors aucune pièce mécanique et sont par conséquent très peu susceptibles de tomber en panne. Leurs temps d'accès sont extrêmement faibles et ils offrent des débits de très hauts niveaux. Ils gèrent des taux d'IOPS conséquent (3 x plus importants que les disques SAS), consomment peu d'énergie et ne chauffe pratiquement pas.

Ces disques doivent être bannis s'ils sont destinés à être utilisés pour héberger des fichiers journaux. En effet, s'ils sont sollicités par de nombreuses écritures, leur durée de vie sera considérablement amoindrie. C'est d'ailleurs pour cette raison que les constructeurs préconisent d'éviter toute indexation des données situées sur ces disques et toute défragmentation.

Les disques SSD dont parfaitement adaptés au stockage de ***tier* 1**.

Les disques hybrides, quant à eux, sont des disques durs traditionnels embarquant une mémoire flash SSD qui fera office de cache. La partie mécanique du disque n'est sollicitée que lorsque la limite du cache est atteinte.

Les temps d'accès sont dès lors meilleurs que ceux offerts par les disques durs traditionnels. Il est cependant important de veiller à ce que la mémoire flash embarquée soit suffisamment importante. En effet, si tel n'est pas le cas, la vitesse de rotation des plateaux du disque va constamment passer d'un état d'activité faible (*Spin down*) à un état d'activité important (*Spin up*), augmentant ainsi les temps d'accès de manière importante.

Les disques hybrides peuvent être de bons candidats pour des disques NL-SAS.

Ils peuvent également se substituer aux disques SATA pour du stockage de ***tier* 3**.

6.1.3.4.5 Les disques 2.5 pouces

Ces disques durs contiennent des pistes magnétiques plus petites qui réduisent d'une part les temps d'accès et d'autre part les contraintes mécaniques et la puissance absorbée.

Le taux de panne diminue ainsi d'environ 15% et la consommation électrique de près de 50%, avec pour corollaire une diminution de la charge thermique. Les performances E/S augmentent de manière significative non seulement grâce au plus faible rayon de la piste magnétique mais également par le biais de la densité plus importante d'informations contenues dans le même espace.

L'encombrement est également moindre au sein du centre de données, les baies dotées de ce type de disque occupant moins de U que les baies dotées de disque 3.5 pouces.

6.1.3.4.6 Duty cycle

Le *duty cycle* qualifie la durée durant laquelle un composant, un dispositif ou un système est sensé opérer. Il est généralement exprimé sous forme de ratio ou de pourcentage. Supposons, à titre d'exemple, qu'un disque fonctionne durant 1 seconde puis cesse ensuite de fonctionner pendant 99 secondes pour ensuite fonctionner à nouveau pendant 1 seconde et ainsi de suite. Ce disque fonctionne donc 1 seconde sur 100, soit 1/100 du temps. Son *duty cycle* est donc 1/100 ou 1%.

Il faut donc tenir compte du fait que plus un composant est utilisé, plus vite il cessera de fonctionner. En conséquence, plus élevé est le *duty cycle*, plus courte est la durée de vie, toute chose égale par ailleurs.

Or, les disques durs sont conçus pour atteindre certains objectifs opérationnels. Les disques FC ou SAS « entreprise » sont prévus pour offrir de hautes performances et un fonctionnement en continu. Ils sont faits de composants robustes qui leur permettent notamment de supporter une température plus importante. Leur coût est élevé car ils incluent des technologies additionnelles pour parvenir à atteindre un tel niveau de qualité.

Les disques à moindre prix ATA et SATA sont conçus pour une utilisation tout à fait différente. Ils ne vont pas forcément être en usage en permanence (ils ne vont pas tourner constamment) et même si c'était le cas, ils n'auraient pas à supporter un nombre aussi important d'E/S qu'un disque SAS.

En conclusion, le *duty cycle* est une indication précieuse pour connaitre l'usage pour lequel un disque a été conçu et doit dès lors être pris en compte lors du choix des disques durs prévus pour une baie de stockage.

6.1.3.4.7 Hiérarchie du stockage

La notion de *tiering* a déjà été abordée à la section 5.4.5, Fonctionnalités avancées. Nous évoquions alors les fonctionnalités avancées de la virtualisation du stockage. Certaines baies proposent nativement cette fonctionnalité, généralement appelée *Hierarchical Storage Management* (HSM). Certains constructeurs commercialisent des solutions qui automatisent le déplacement des données d'un support de données à un autre (*Automated Tiering Storage)*, s'agissant notamment de **Dell Compellent™**[104], **EMC²®**[105], **HP 3PAR™**[106], **Zarafa®**[107] ou **Quantum StorNext®**[108]. Virtualiser le stockage n'est donc pas forcément nécessaire pour bénéficier de cette technologie.

Nous rappelons que le but est de parvenir à une réduction des coûts relatifs au stockage en scindant les données sur des supports différents, en fonction de certains critères, tels que :

- La performance ;
- La fréquence d'accès ;
- L'importance.

Nous avons déjà mis en relation les types de disques avec les niveaux (*tiers*) auxquels ils correspondent le mieux (cf. section 5.4.5, Fonctionnalités avancées, ainsi que la présente section).

Inventorier les données en fonction des critères mentionnés ci-dessus permet de dimensionner au plus près les besoins en stockage. À titre d'exemple, les données les plus importantes peuvent être stockées sur deux baies répliquées en fibre optique avec des contrôleurs redondants, alors que les fichiers de moindre importance peuvent être archivés sur un NAS doté de disques SATA 7'200 t/m.

L'importance des données stockées sur un support répond à la loi de Pareto. En effet, les utilisateurs estiment que 20% seulement des données ainsi stockées leur sont essentielles. 80% des données devraient dès lors pouvoir être déplacées sur des supports de stockage moins onéreux. Une même baie peut être dotée, à cet effet, de différents types de disques configurés en fonction des besoins. À titre d'exemple, des disques SSD peuvent être configurés en RAID 1 pour héberger les données critiques et des disques SAS en RAID 5 pour des données usuelles.

6.1.4 Choix du réseau de stockage

Les deux types de média, et les réseaux de stockage qui en découlent, sur lesquels baser une infrastructure de stockage à l'heure actuelle sont la fibre optique par le biais du protocole

[104] http://www.compellent.com/.
[105] http://suisse.emc.com/index.htm?fromGlobalSiteSelect.
[106] http://h18006.www1.hp.com/storage/disk_storage/3par/index.html.
[107] http://www.zarafa.com/content/zarafa-archiver.
[108] http://h71028.www7.hp.com/enterprise/cache/486414-0-0-0-121.html.

Fibre Channel et l'Ethernet qui s'appuie sur le protocole **iSCSI** (transport de données en mode bloc en utilisant le protocole SCSI avec une encapsulation à l'aide du protocole IP).

Si un réseau de stockage à base de fibre optique s'avère plus complexe à mettre en œuvre que son homologue iSCSI et est surtout plus couteux, les performances obtenues sont bien meilleures.

Les débits atteints au sein des infrastructures actuelles sont généralement de 4 ou 8 Gbit/s pour la fibre optique et 1 Gbit/s pour l'Ethernet, étant précisé que la norme 10 Gbit/s est en train de faire progressivement sa place sur le marché. Le débit de 16 Gbit/s pour le Fibre Channel est d'ores et déjà atteint (Emulex®, à titre d'exemple, fournit des adaptateurs à Dell®, basés sur la technologie LightPulse calibrée pour ce débit).

Nous précisons que, contrairement à l'Ethernet, la fibre optique est capable de transférer des données sans pertes de paquets, ce qui en fait le meilleur choix pour les applications les plus exigeantes.

La liste des réseaux de stockage possible ne serait pas complète sans que nous évoquions le **Fibre Channel over Ethernet** (FCoE). Cette norme permet une convergence, sur les mêmes réseaux, entre la fibre optique et l'Ethernet en utilisant les éléments déjà en place, ainsi que des cartes réseau CNA (*Converged Network Adapter)*. Nous n'aborderons pas outre mesure cette technologie dans le présent document. Nous souhaitions toutefois que le lecteur ait connaissance de son existence.

6.1.4.1 Fibre optique

La vitesse souhaitée détermine le choix de la carte fibre optique. Il convient toutefois de prendre garde à l'existence de certaines fonctionnalités qui peuvent s'avérer fort utiles :

- Le *Persistent Blinding* : il lie de façon permanente le SCSI Id d'un équipement de stockage avec un WWN (cf. section 5.5.3, Autres éléments du réseau) ;
- L'optimisation de la bande passante pour des périphériques de débit différent (par exemple, la technologie iiDMA de QLogic) ;
- Le mécanisme de réassemblage automatique des trames FC qui permet d'éviter une retransmission complète (par exemple, la technologie OoOFR de QLogic) ;
- La technologie d'activation de ports virtuels NPIV (cf. section 5.5.3, Autres éléments du réseau).

Nous avons abordé en détails la fonctionnalité NPIV à la section 5.5.3. Nous rappelons toutefois que grâce à cette technologie, un serveur physique hôte doté d'une seule carte fibre optique est capable de mettre à disposition de chacune des machines virtuelles qu'il héberge une carte fibre optique virtuelle qui lui sera propre. NPIV apporte donc une flexibilité appréciable au niveau de la connectivité, en permettant la réduction potentielle du nombre de câbles, une facilité accrue dans l'allocation des ressources et dans la mise en place de fonctionnalités de haute disponibilité.

Distance

Les distances potentielles entre les équipements fibre optique qui doivent être desservis par la fibre optique doivent faire l'objet d'un examen attentif. La fibre multimode est

généralement utilisée sur des distances plutôt courtes alors que la fibre monomode est privilégiée pour les longues distances.

Cependant, il convient de tenir compte du fait que les distances maximales possibles peuvent varier en fonction du diamètre de la fibre (50 ou 62.5 microns) et du débit souhaité (1, 2, 4, 8 ou 16 Gbit/s).

La qualité de la fibre doit également être adaptée en conséquence.

Le lecteur est prié de se référer aux recommandations des fabricants à cet égard.

Zoning

Même s'il existe différentes topologies utilisées pour les réseaux Fibre Channel, comme le point à point (*Point-to-Point*) ou la boucle (*Arbitrated Loop*), seule la topologie commutée (*Fabric*) nous intéresse à ce stade. En effet, c'est généralement la topologie la plus à même de faire correspondre le réseau de stockage aux attentes des entreprises.

Le zoning correspond, pour un réseau FC, au VLAN. Il consiste en un partitionnement d'une Fabrique (*Fabric*) FC, soit un commutateur fibre optique, en plusieurs sous-réseaux. Chacune des zones peut ainsi être isolée afin des prévenir les dommages potentiels qui peuvent corrompre les allocations ou déstabiliser l'entier du réseau de stockage.

Un des buts habituels du zoning consiste à faire en sorte que l'homogénéité des systèmes d'exploitation puisse être garantie au sein d'une même zone, ceci afin d'éviter certains scénarios plutôt dangereux. À titre d'exemple, lorsqu'un Windows Serveur accède à des LUN, il écrit systématiquement des blocs sur chacune d'entre elles. Si des serveurs UNIX® accèdent à ces mêmes LUN par le biais de la même fabrique, leurs systèmes de fichiers s'en trouveront dès lors totalement corrompus.

Le zoning permet également de protéger certains périphériques des notifications provenant d'autres périphériques (comme les *Registered State Change Notification*). En effet, ce type de notifications peut provoquer des ruptures de communication susceptibles d'engendrer des pertes de données. Les types de périphériques peuvent dès lors être protégés les uns des autres en étant situés dans des zones différentes.

Lors de la création de zones, la règle consistant à baser la création de zone par rapport à l'initiateur (*Single Initiator Zoning*) est conseillée. De multiples périphériques de stockage peuvent ainsi être ajoutés à une zone sans violer des règles telles que la séparation de la production et du développement.

Le zoning peut être effectué en fonction des différents systèmes d'exploitation en vigueur au sein de la société, des ports, des applications, etc.

En dernier lieu, il convient d'adopter une topologie de fabrique adaptée aux différents types d'accès possibles. Ces derniers sont au nombre de trois :

- **Local** (one-to-one) : les données sont accédées par un serveur et une baie de stockage, tous deux connectés sur la même fabrique ;
- **Centralisé** (many-to-one) : les données sont accédées par plusieurs serveurs et une baie de stockage ;
- **Distribué** (many-to-many) : les données sont accédées par plusieurs serveurs et plusieurs baies de stockage.

Les différents types de topologies sont les suivantes :

- Fabrique **unique** ;
- Fabriques **cascadées** (*cascaded*) ;
- Fabriques **maillées** (*meshed*) ;
- Fabriques en **anneau** (*ring*).

Le tableau ci-dessous indique la topologie à adopter en fonction du type d'accès :

Topologie	Performances en fonction du type d'accès		
	Local	**Centralisé**	**Distribué**
Fabrique unique	Excellentes	Excellentes	Excellentes
Fabrique cascadée	Excellentes	-	-
Fabrique maillées	Bonnes	Bonnes	Excellentes
Fabrique en anneau	Excellentes	Bonnes	-

Tableau 6-5 - Performances des différentes topologies en fonction des types d'accès. (Source : Bonnes pratiques, planification et dimensionnement des infrastructures de stockage et de serveur en environnement virtuel par Cédric Georgeot)

Sauvegarde et zoning

Les dispositifs de sauvegarde ne devraient pas être situés sur la même fabrique que celle du stockage ou, à tout le moins, dans la même zone que le stockage. Il est en effet préconisé de séparer les deux genres de flux afin d'éviter tout problème.

Il vaut mieux, dès lors, implémenter des adaptateurs fibre distincts sur les serveurs (ou alors faire en sorte qu'ils soient doté d'adaptateurs multiports). Si un seul adaptateur est disponible et qu'il ne possède qu'un port, une zone dédiée à la sauvegarde devrait être créée.

6.1.4.2 iSCSI

Les réseaux de stockage sont originellement liés à la fibre optique et au protocole Fibre Channel. L'avènement de la norme Gigabit Ethernet a cependant bouleversé quelque peu la donne. Il est en effet possible, à moindre coût, de mettre en œuvre une architecture SAN, en se basant sur les équipements existants, tout en conservant des performances honorables. L'agrégation des liens Ethernet est cependant conseillée pour y parvenir.

Une infrastructure basée sur la technologie iSCSI est plus simple à mettre en œuvre que son équivalent Fibre Channel.

Les entreprises qui ne disposent pas des ressources financières nécessaires à l'achat des dispositifs Fibre Channel et/ou qui ne possèdent pas les ressources humaines qualifiées pour se faire, peuvent tout de même mettre en place un stockage partagé de qualité.

Les professionnels conseillent tout de même de respecter les points suivants :

- Les différents flux de données doivent être isolés. Le réseau de stockage doit, à titre d'exemple, être isolé de la gestion, de la production, etc. Pour se faire, plusieurs cartes réseaux peuvent être mises en place. L'implémentation de plusieurs VLAN au niveau des commutateurs et l'utilisation de sous-réseau IP différents est également de mise ;
- Plusieurs liens Gigabit Ethernet peuvent être agrégés. Ces derniers peuvent également être déployés de manière redondante. Les flux doivent ensuite être priorisés par le biais de la qualité de service (QoS), en fonction de l'importance de leur accès au stockage ;
- Les cartes embarquant ou prenant en charge des techniques de délestage, telles que TOE (*TCP Offload Engine*) doivent être privilégiées. Des technologies telles que NetQueue ou VMQ[109] (*Virtual Machine Queue*) qui assure le routage des flux réseaux directement vers la mémoire de la machine virtuelle doivent également être prises en compte, à l'instar de Microsoft Chimney Offload Architecture[110] (délestage TCP/IP par la carte réseau). Des optimisations de flux ou des économies de charge processeur sont ainsi possibles ;
- Les trames Jumbo doivent être activées car elles vont augmenter la capacité de transmission maximale de données (MTU) vers un système distant ;
- L'authentification CHAP (pour sécuriser le flux de stockage) doit être utilisée au minimum, même si le flux de stockage est isolé de la production.

Il convient également, tant que faire se peut, d'utiliser des commutateurs optimisés pour le protocole iSCSI. Ces derniers sont mieux à même de supporter les débits y relatifs et les taux importants d'opérations d'E/S. Ils sont de plus capables d'attribuer la plus haute priorité aux flux iSCSI.

Les fonctionnalités de détection des tempêtes Unicast et de boucles (*Spanning Tree*) doivent être désactivées sur les ports par lesquels transitent les flux iSCSI. De plus, les vitesses devraient y être configurées de manière statique et le contrôle de flux (Flow Control) activé.

La puissance actuelle des processeurs évitent généralement l'usage d'HBA iSCSI, même si en utilisation intensive un stockage iSCSI peut consommer jusqu'à 10% des capacités processeur. Si l'usage d'une carte HBA iSCSI pourrait éventuellement être recommandé dans ce cas précis, les mécanismes de délestage suffisent généralement.

Ce type de carte devient indispensable si l'hyperviseur doit être démarré à partir du stockage (Boot-On-San) ou si l'infrastructure doit reposer sur la norme 10 Gigabit Ethernet.

[109] http://technet.microsoft.com/fr-fr/library/gg162704(v=ws.10).aspx.
[110] http://msdn.microsoft.com/en-us/library/windows/hardware/ff569969(v=vs.85).aspx.

6.1.4.3 NFS/CIFS

Jusqu'à il y a peu, les performances offertes par ces protocoles étaient considérées comme insuffisantes. Même si nous ne pouvons toujours pas considérer les performances obtenues avec les solutions basées sur ces protocoles comme étant équivalentes à celles obtenues avec des solutions basées sur le Fibre Channel, ces dernières peuvent maintenant trouver leur place au sein de petites infrastructures. Il est tout de même nécessaire que les exigences en matière d'opérations d'E/S ne soient pas très élevées.

Ces protocoles sont en général implémentés dans les NAS. Le déploiement de ce type de stockage est bien entendu très aisé. Le coût des NAS est faible et ils permettent d'atteindre les mêmes volumes de stockage que les SAN.

Microsoft dispose, par le biais de son système d'exploitation **Windows Storage Server™ 2012**[111], d'une solution permettant la mise en œuvre d'une infrastructure de stockage de type NAS capable de proposer un stockage mutualisé supportant aussi bien les protocoles permettant l'accès aux fichiers (SMB et NFS) qu'aux blocs (iSCSI). Cette solution est capable de mettre en œuvre les technologies de *Thin Provisioning* et de déduplication. Étant donné sa capacité à gérer le protocole iSCSI et à pouvoir créer des disques virtuels, cette solution peut également être apparentée à un SAN.

D'autres solutions, de type *appliance*, fournissant des fonctionnalités similaires à Windows Storage Server existent sur le marché, s'agissant par exemple de **OpenFiler™**[112], **Open-E™**[113] ou **FreeNAS™**[114].

Malheureusement, un stockage basé sur le CIFS ou le NFS est susceptible de restreindre l'accès aux fonctionnalités avancées de certains hyperviseurs. À titre d'exemple, l'usage du NFS avec VMware n'autorise pas la création de partitions brutes (RDM).

6.1.4.4 Redondance et agrégation

Les techniques de redondance et d'agrégation sont essentielles au sein d'un SAN. La première permet la continuité de l'accès au stockage malgré la perte d'un élément du réseau. La seconde permet la multiplication de la vitesse au sein des liens agrégés.

Des protocoles comme **LACP** (*Link Aggregation Control Protocol*), qui est décrit dans la norme IEEE 802.3ad, **EtherChanne**l de Cisco® ou le ***Trunking***, contribuent à sécuriser les environnements iSCSI au niveau des commutateurs. Des techniques comme **MPIO** (*Multi Path I/O*) et **MCS** (*Multiple Connections per Session*) permettent une haute disponibilité du stockage au niveau de l'initiateur.

MPIO permet aux initiateurs iSCSI d'ouvrir plusieurs sessions vers la même cible en la présentant de façon unique. Chaque session peut ainsi être établie en utilisant des cartes réseau ou des commutateurs différents. MCS permet l'agrégat de plusieurs connexions dans

[111] http://www.microsoft.com/en-us/server-cloud/windows-storage-server/default.aspx.
[112] http://www.open-e.com/.
[113] http://www.openfiler.com/.
[114] http://www.freenas.org/.

une seule session, les opérations d'E/S étant ainsi expédiées sur n'importe quelle connexion TCP/IP de la cible.

Ces deux techniques ne permettent pas de partager une commande SCSI sur plusieurs liens. Elles sont utilisées pour mettre en œuvre la redondance de liens.

Quant à l'agrégation de liens, elle permet à plusieurs liens distincts d'être perçus comme un seul et unique lien. La bande passante peut dès lors être démultipliée. Il ne s'agit pas du seul avantage puisque la redondance est également mise en œuvre par ce biais, à l'instar de la répartition de charge.

6.1.4.5 Répartition des charges

Nous l'avons vu à la section précédente, la technique de répartition de charge MPIO est capable d'accroitre la disponibilité des accès au stockage partagé en établissant plusieurs connexions. Ces dernières sont établies au travers d'un DSM (*Device Specific Module*).

Une architecture hautement disponible exige le doublement des équipements à tous les niveaux, qu'il s'agisse des cartes HBA, des commutateurs, les liens réseau ou des contrôleurs actif/actif.

Hormis la haute disponibilité, les performances seront améliorées et un point unique de défaillance (SPOF) peut être éliminé.

Le paramétrage d'une politique de répartition des charges peut varier en fonction de l'hyperviseur, du DSM ou de la baie de stockage. Les plus courantes d'entre elles sont les suivantes :

- ***Fail Over*** *(répartition alternée)* : un chemin d'accès actif est utilisé et tous les autres chemins sont définis comme étant en attente. Dans le cas d'une défaillance du chemin actif, tous les chemins en attente sont tentés à tour de rôle ;
- ***Round Robin*** (répartition alternée) : les flux sont répartis uniformément sur tous les chemins disponibles ;
- ***Round Robin with a subset of paths*** (répartition alternée avec sous-ensemble) : cette technique est similaire au *Round Robin* à la différence que des chemins actifs et passifs peuvent être définis ;
- **Weighted Path** (chemins d'accès mesurés) : une charge de traitement spécifique relative à chaque chemin d'accès est prise en compte. Le nombre obtenu qualifie la qualité du chemin d'accès (si, par exemple, le nombre est élevé, la priorité du chemin d'accès est faible) ;
- **Least Queue Depth** (longueur minimale de la file d'attente) : cette technique est uniquement supportée par MCS. Les charges non uniformes sont compensées par une répartition proportionnelle d'un nombre plus important d'opérations d'E/S au niveau des chemins d'accès les moins sollicités.

D'après certains professionnels, la politique la plus performante, tant en termes de débit que d'opérations d'E/S, est le Round Robin with a subset of paths, suivie de près par le Fail Over.

6.1.5 Paramètres de performances

Au sein des sections précédentes, nous avons fait références aux critères dont il faut être particulièrement attentif lors de l'acquisition du matériel prévu pour la virtualisation des serveurs et le stockage partagé. Un certain nombre de paramètres, en grande partie liés au stockage mutualisé, doivent également faire l'objet d'un examen attentif, une fois le matériel à disposition, pour bénéficier des meilleures performances possibles.

Il convient en particulier d'être vigilant par rapport au **niveau de RAID** à adopter lors de la création des grappes y relatives. La **profondeur de la file d'attente** (*Queue Depth* ou *Execution Throttle*), qui détermine le nombre d'opérations d'E/S pouvant s'exécuter parallèlement sur une unité de stockage, est également un élément décisif dans l'optimisation des IOPS et des temps d'accès. L'**alignement des partitions** exerce également un impact certain sur les performances en matière de nombre d'opérations d'E/S.

Une bonne gestion de ces différents paramètres s'avère d'ailleurs essentielle tant au sein d'un environnement virtuel qu'en environnement classique. Nous n'avons toutefois par prévu de les traiter en détails au sein de ce document, ce dernier portant principalement sur la virtualisation et non sur le stockage. Il nous semble cependant essentiel que le lecteur ait été incité à s'y intéresser. C'est la raison pour laquelle ces paramètres sont abordés succinctement dans cette section, charge au lecteur de parfaire ses connaissances par le biais d'ouvrages (notamment l'excellent livre *Bonnes pratiques, planification et dimensionnement des infrastructures de stockage et de serveur en environnement virtuel*[115]) ou autres publications y relatives.

6.1.5.1 Niveaux de RAID

Les niveaux de RAID font l'objet d'une abondante littérature. Il s'agit principalement d'adapter le niveau de RAID en fonction des applications qui solliciteront le stockage partagé. Si l'adaptation en question est optimale, les performances s'en ressentent immédiatement, et ce, au niveau des opérations d'E/S comme du débit.

Le nombre de disques de rechange (*spare*) doit être intégré au calcul du nombre de disques nécessaires et l'impact en termes de temps de reconstruction doit être pris en compte, en particulier dans le cas où les applications sollicitent énormément et en permanence le stockage. En effet, la durée nécessaire à la reconstruction lors de la défaillance d'un des disques peut varier en fonction du niveau de RAID. Il convient de se documenter à ce propos. La pénalité en écriture, liée aux disques de parité, doit être prise en compte, en particulier en ce qui concerne le RAID 6 (double parité).

Chaque application est caractérisée par un certain modèle au niveau des opérations d'E/S. Les accès peuvent être principalement aléatoires ou plutôt séquentiels. Soit les opérations de lecture prédominent, soit il s'agit des écritures. Le niveau de RAID doit également être adapté à ces divers modèles. Tant le débit que le nombre d'IOPS sont concernés par ce paramètre.

[115] Par Cédric Georgoet (cf. chapitre 10, Bibliographie).

Quant au RAID 6, il offre une sécurité accrue dans le cas où un grand nombre de disques sont contenus dans une pile RAID. Mais il faut être conscient que la double parité (2 disques durs y sont dédiés) a un prix. Du fait du nombre de disques, le coût est plus important. Les performances sont moindres que celles offertes par le RAID 5 et le temps de reconstruction est plus élevé. Il faut savoir que certains contrôleurs RAID embarquent un processeur dotés d'algorithmes spéciaux, destinés à la gestion de la parité propre au RAID 6 (*Application Specific Integrated Circuit*).

6.1.5.2 Profondeur de file d'attente

Pour que des données puissent être lues ou écrites sur les disques d'une baie de stockage, des commandes SCSI sont expédiées vers cette dernière. Par file d'attente, nous entendons en premier lieu la file d'attente de la carte HBA. La baie de stockage possède néanmoins sa propre file d'attente. Dans le cas d'en environnement virtuel, une couche supplémentaire se situe entre le système d'exploitation de la machine invitée (virtuelle) et la carte HBA du serveur hôte. Concernant VMware, par exemple, c'est l'ordonnanceur (*scheduler*), qui est un composant du VMkernel, – nous nous situons ici au niveau de l'hyperviseur – qui est chargé d'empiler les commandes SCSI provenant simultanément des différentes machines virtuelles au sein de la propre file d'attente du VMkernel (la profondeur de cette file d'attente est contrôlée par le paramètre Disk.SchedNumReqOutstanding et est égal à 32 par défaut).

Pour éviter d'engorger du sous-système disque par un nombre trop important de commandes, ces dernières sont placées au sein de la file d'attente de la carte HBA avant d'être expédiées à la baie de stockage. Ces commandes peuvent dès lors être traitées ultérieurement.

Il s'agit, en conséquence, de définir judicieusement une taille pour les files d'attente et de s'assurer que chacune d'entre elles soit paramétrée selon cette taille. Cette dernière ne doit être ni trop faible ni trop importante, afin que les temps d'accès et les IOPS soient parfaitement optimisés.

À titre d'exemple, si le paramètre Disk.SchedNumReqOutstanding est plus petit que la taille de la file d'attente de la carte HBA, il est impossible de tirer le meilleur parti de cette dernière. Dans le cas contraire, des requêtes sont refusées par la même file d'attente et un goulet d'étranglement est ainsi créé.

S'agissant du stockage, ce paramètre peut être défini au niveau de la baie, d'une cible (composée de différents LUN), d'un LUN, d'un port ou éventuellement d'un disque, avec des conséquences différentes en fonction des choix effectués.

Les bonnes pratiques suggèrent que la profondeur soit établie au niveau de la cible ou du LUN. À tire d'exemple, définir une profondeur de file d'attente au niveau d'une cible permet de limiter intentionnellement les IOPS, alors qu'effectuer un tel paramétrage au niveau du LUN permet de définir le nombre d'IOPS qui peuvent être envoyées au maximum par LUN. Une valeur élevée peut ainsi permettre l'accès prioritaire d'un serveur, tandis que l'inverse offre la possibilité de se prémunir d'un engorgement au niveau du stockage.

Il faut également savoir que plus la profondeur d'une file d'attente est importante, plus les données sont accumulées, avec pour corollaire un débit accru mais une augmentation de la

latence. Une petite profondeur va, quant à elle, améliorer la latence par le biais d'un nombre plus important de commandes.

Une grande profondeur de file d'attente est idéale lorsque nous disposons de larges piles RAID et inversement une petite profondeur est recommandée pour les petites piles RAID (dans le cas, par exemple, d'une base de données transactionnelles qui requiert des temps de latence très faibles).

6.1.5.3 Alignement de partition

L'alignement de partition est un élément essentiel au niveau du stockage. Il permet également d'optimiser les opérations d'E/S. En effet, tous les systèmes d'exploitation ne démarrent pas la première partition principale au même secteur (63 au lieu de 64), ce qui provoque un déséquilibrage du *stripe* (peut être assimilé à un bloc ou un *chunk* sur un même disque dur).

Windows Server 2008, ainsi que les versions ultérieures, réalisent automatiquement cette opération et ne sont donc pas concernés par ce paramètre.

Si la partition n'est pas correctement alignée, une opération d'écriture peut être partagée en deux et faire intervenir deux disques au lieu d'un seul. Si, à titre d'exemple, une telle écriture devait être effectuée sur une pile RAID composée de six disques, de nombreuses écritures supplémentaires seraient provoquées par ce décalage, en lieu et place des six écritures qu'une partition alignée aurait générées.

Pour une seule commande, les opérations d'E/S sont donc démultipliées. Quant aux IOPS et aux débits, ils chuteront au fur et à mesure du remplissage de la partition, les données étant du plus en plus décalées.

VMware annonce d'ailleurs des gains de performance de l'ordre de 10% à 40% avec des partitions correctement alignées.

Pour vérifier si une partition est alignée sous Windows, la commande suivante doit être entrée :

```
Vmnic partition get blocksize, name, index, startingoffset
```

La commande similaire à exécuter au sein d'un environnement VMware est :

```
fdisk -lu
```

Elle doit être exécutée avec les privilèges ROOT.

Il convient de préciser que l'alignement de partitions provoque généralement la destruction des données qui y étaient stockées.

6.2 Choix des fournisseurs

Cette section est destinée à produire un inventaire non exhaustif des **principaux éditeurs de solutions de virtualisation**, ainsi que de certains **fabricants qui disposent de partenariats avec ces derniers**.

Nous avons d'ores et déjà cité les éditeurs de solutions de virtualisation au sein de la section 4.4, Mise en œuvre, au sein de laquelle sont notamment décrites les applications commerciales issues des différentes techniques de virtualisation.

Les principaux éditeurs, et les solutions qu'ils proposent, y sont spécifiés en fonction du type de mise en œuvre (Hyperviseur de type 1 ou 2, émulation, virtualisation au niveau noyau, etc.). Ces mêmes éditeurs sont à nouveau cités, à l'instar de certains fabricants au cours du chapitre 5, Domaines d'application, cette fois-ci en fonction précisément du domaine d'application de la virtualisation au sein duquel ils sont actifs. Enfin, certains éditeurs, et les technologies qu'ils proposent, sont mentionnés tout au long du présent chapitre. Nous avons en effet jugé opportun d'évoquer les éditeurs et les constructeurs alors que nous évoquions les solutions et autres technologies qu'ils proposent, et ce, dans un souci de cohérence.

C'est la raison pour laquelle nous n'allons pas les citer à nouveau dans la présente section.

Nous précisons toutefois que, le présent document portant sur la virtualisation du système d'information, les fabricants qui nous intéressent en premier lieu, sont ceux qui sont à même de fournir des serveurs destinés à des hyperviseurs. Il est à noter que ces derniers fournissent bien souvent des solutions de stockage ou disposent de partenariats avec les sociétés qui sont susceptibles de les fournir. Ces fabricants disposent également de partenariat avec les principaux éditeurs de solution de virtualisation, ce qui fait d'eux des interlocuteurs privilégiés pour la mise en place d'une infrastructure virtuelle. Il est même souhaitable de prendre contact en premier lieu avec ces fabricants, plutôt qu'avec les éditeurs de solutions de virtualisation.

À ce titre, les fabricants qui demeurent incontournables sur ce marché sont **Dell**®, **HP**® et **IBM**®.

7 Gestion de l'infrastructure virtuelle

Ce chapitre est destiné à mettre l'accent sur certains points essentiels qui doivent être pris en considération une fois l'infrastructure virtuelle fonctionnelle.

Nous rappelons, à la section 7.1, Sécurisation, les grands principes de la sécurité des systèmes d'information que nous mettons en relation avec les particularités d'une infrastructure virtuelle.

Certains mécanismes permettant d'assurer au mieux la disponibilité de l'infrastructure et l'intégrité des données y sont décrits. Nous faisons également référence à certaines notions qui s'avèrent fort utiles lorsque les conséquences résultant d'un sinistre et les plans de reprise d'activité sont évoqués au sein de l'entreprise. Ces dernières sont destinées à aider le lecteur à évaluer le plus correctement possible les niveaux de protection du système d'information et les mécanismes qui doivent être mis en place pour parvenir à les atteindre.

La section 7.2, Postproduction, traite des méthodes à mettre en œuvre pour conserver l'infrastructure fonctionnelle.

La planification de la reprise d'activité en cas de sinistre y est également abordée et, en conséquence, nous apportons des précisions quant à la notion de basculement vers un site de secours. Nous y évoquons également certains logiciels susceptibles de faciliter l'élaboration d'un plan de reprise d'activité, ainsi que certaines technologies ayant pour but d'en assurer une mise en œuvre efficace, le cas échéant.

Cette section porte également sur la planification budgétaire relative au fonctionnement de l'infrastructure virtuelle, une fois cette dernière fonctionnelle. Nous faisons notamment référence aux coûts d'exploitation, aux frais d'amortissements relatifs au matériel et aux systèmes de licences tels que pratiqués par les éditeurs de solutions de virtualisation.

De plus, nous apportons quelques précisions quant à la manière d'évaluer les coûts réels relatifs aux machines virtuelles et la raison pour laquelle il peut être avisé d'être capable de les produire.

Enfin, nous terminons ce chapitre en évoquant les perspectives qui nous sont offertes par le développement conséquent de la virtualisation au sein des centres de données, ainsi que le changement de paradigme que ce dernier pourrait impliquer, tant dans l'élaboration des systèmes d'information que dans leur gestion.

7.1 Sécurisation

Quoiqu'un système d'information partiellement ou intégralement virtualisé réponde à plusieurs critères qui lui sont propres, sa sécurisation repose sur les mêmes principes que ceux auxquels n'importe quel système d'information classique est d'ores et déjà soumis.

Ce document s'adresse à des professionnels qui sont déjà censés faire face à cette problématique. En effet, gérer un système d'information revient à s'assurer que les actifs matériels ne subissent aucun dégât susceptible d'en altérer l'usage. Nous savons toutefois

que la vraie richesse d'une société réside dans ses actifs immatériels, autrement dit le capital informationnel. Nous entendons par capital informationnel, le savoir-faire, matérialisé par les brevets, dont découlent les méthodes de production. Nous faisons référence également à la clientèle, à la concurrence, etc. Ces informations sont contenues sous des formes diverses au sein du système d'information, ce dernier revêtant dès lors une importance toute particulière.

Aussi, le lecteur est-il certainement au fait des grands principes inhérents à la sécurité des systèmes d'information que sont :

- La garantie que le système d'information soit accessible lorsque désiré, moyennant un temps de réponse adéquat (Disponibilité) ;
- La garantie que seules les personnes autorisées ont bel et bien accès au système d'information, les accès indésirable étant prohibés (Confidentialité) ;
- La garantie que les éléments considérés du système d'information n'ont pas été altérés fortuitement ou volontairement, et sont dès lors exacts et complets (Intégrité) ;
- La garantie que les accès et tentatives d'accès aux éléments considérés du système d'information sont journalisés puis conservés, et qu'ils sont, à ce titre, exploitables (Traçabilité).

L'existence d'une infrastructure virtuelle n'affranchit en rien l'administrateur système d'avoir à évaluer les risques inhérents à l'existence d'un système d'information. Les technologies relatives à la virtualisation évoluent très rapidement. Les failles potentielles sont généralement corrigées par les éditeurs avec un degré de célérité que n'est pas différents de celui que nous connaissons déjà lorsque des solutions classiques sont concernées. Aussi, nous ne procéderons pas à une analyse exhaustive des risques liés à la virtualisation dans le présent document. Nous souhaitons toutefois attirer l'attention du lecteur sur quelques concepts que nous jugeons particulièrement importants.

La confidentialité de l'infrastructure virtualisée, et en particulier des serveurs hôte de machines virtuelles, doit être garantie en évitant toute intrusion. Assurer la confidentialité du système d'information est un défi auquel les administrateurs des infrastructures classiques sont déjà soumis. Cependant, des paramètres propres à la virtualisation doivent être pris en compte dans ce cas précis, la complexité de l'infrastructure s'étant amplifiée. Il convient dès lors d'intégrer les particularités de ce type d'infrastructure et d'agir en conséquence.

De plus, il n'est pas forcément évident d'évaluer précisément la criticité des serveurs hôte de machines virtuelles. En effet, il faut considérer qu'un tel serveur est susceptible d'héberger un nombre potentiellement important de machines virtuelles. La perte d'un tel dispositif peut dès lors prendre des proportions conséquentes. Il convient également de considérer que certaines machines virtuelles sont plus critiques que d'autres et que ces dernières peuvent être déplacées d'un serveur physique à un autre. Il faut donc caractériser la criticité de l'hôte régulièrement.

Il est nécessaire, en conséquence, de considérer que, si la virtualisation permet de consolider un nombre important de machines virtuelles sur un seul serveur physique, ce dernier doit faire l'objet d'une supervision irréprochable. Au même titre, toutes les mesures nécessaires à sa haute disponibilité doivent être prises. Nous entendons, au niveau purement matériel, la redondance des alimentations, des ventilateurs, des cartes réseau,

voire des processeurs ou de la mémoire. Mais il est également primordial de créer des fermes (*cluster*) de serveurs physiques. Les machines virtuelles qui dépendent d'un serveur défectueux peuvent ainsi être exploitées à partir d'un autre serveur du même *cluster* de basculement, et ce, grâce aux différentes technologies de haute disponibilité.

Ceci nous amène à traiter du stockage. Nous avons vu à la section 2.1, Évolution du modèle de centre de données, que ce dernier devient la pièce maîtresse de l'environnement virtuel. En effet, ne contient-il pas, en plus des données, les machines virtuelles elles-mêmes, qui ne sont ni plus ni moins que des fichiers. Au même titre que les serveurs physiques, le stockage doit donc disposer des technologies nécessaires à sa haute disponibilité. Il s'agit également d'assurer la redondance des alimentations et autres ventilateurs mais également des cartes HBA. Il est également primordial de garantir la perte d'un ou plusieurs disques par le biais de la création de pile RAID (cf. section 6.1.5.1, Niveaux de RAID).

La sécurisation des données est donc capitale en environnement virtuel, et ce, tant pour garantir la disponibilité du système d'information que pour en assurer son intégrité.

Sont abordées, dans les sous-sections ci-dessous, un certain nombre de mécanismes qui permettent d'ajouter la redondance des données à la sécurisation offerte par les piles RAID, les clusters de basculement ou la redondance des éléments matériels. Les mécanismes dont il est question sont souvent mis en œuvre simultanément afin d'assurer au mieux la redondance désirée et la sauvegarde des données.

Enfin, nous faisons souvent référence, lorsque nous évoquons les conséquences d'un arrêt de la production et les coûts qui en résultent, aux notions de Plan de reprise d'activité (PRA) ou de Plan de continuité d'activité (PCA).

Ces dernières sont intimement liées aux concepts de RTO (*Recovery Time Objective*) et RPO (*Recovery Point Objective*). Le premier fait référence à une durée maximale tolérée avant la reprise d'activité. Le second qualifie l'état minimal dans lequel le système doit se trouver au moment de cette reprise (l'entreprise peut-elle fonctionner un certain temps en mode dégradé ou la totalité des services informatiques doit-elle être disponible, pouvons-nous nous passer des données de la dernière journée d'activité ou ces dernières doivent-elles être restaurées, etc.).

Il est bien entendu souhaitable que les valeurs respectivement du RTO et du RPO soient le plus proches possible de zéro. Cependant, les coûts y relatifs n'en seront que plus élevés.

Les types de sinistres potentiels étant malheureusement nombreux, il est impératif de définir correctement la disponibilité désirée par type de données, le budget nécessaire étant immanquablement conditionné par les décisions y relatives (un serveur de développement n'a pas forcément à être protégé comme celui hébergeant une solution ERP).

La redondance des éléments physiques (comme les alimentations ou deux contrôleurs par baie) d'un serveur ou d'une baie de stockage ou les licences autorisant l'usage des solutions de haute disponibilité permettent uniquement de renforcer la disponibilité du système d'information (PCA). Un virus ou une corruption des données peut survenir, malgré la mise en œuvre de telles mesures.

À titre d'exemple, la mise en place d'un *cluster* de basculement contribue fortement à approcher une valeur RTO proche de zéro. Si, *a contrario*, ladite valeur s'en éloigne, nous nous situons plutôt dans une stratégique de PRA (existence de procédures de restauration d'un serveur à partir d'une sauvegarde, etc.).

Si la perte de données n'est pas envisageable, la valeur RPO doit être proche de zéro. Il convient dès lors d'opter pour une réplication synchrone (cf. section 7.1.1.1, Réplication synchrone). Si nous nous éloignons de cette valeur, la réplication asynchrone (cf. section 7.1.1.2, Réplication asynchrone) est probablement la solution adéquate.

7.1.1 Réplication

Une condition *sine qua non* à la mise en place d'une synchronisation adéquate des données consiste à connaître parfaitement ces dernières, en particulier leur volume et leur taux de changement.

Il convient également de sélectionner convenablement de système de stockage. Les solutions de réplication peuvent en effet être intégrées au contrôleur d'une baie mais peuvent également se présenter sous forme logicielle. La technologie en question doit être capable d'optimiser le stockage par l'usage d'algorithme de compression, de *tiering* (cf. section 5.4.5, Fonctionnalités avancées) ou de déduplication (cf. section 7.1.4, Déduplication).

7.1.1.1 Réplication synchrone

Ce type de réplication est prévu pour faire en sorte que la correspondance entre les données présentes sur la première baie (source) et la seconde (cible) soient parfaite, et ce, à tout moment. Il s'agit donc d'une réplication « en temps réel ».

Un débit important, et disponible en tout temps, est indispensable car le processus de réplication va systématiquement attendre de recevoir l'acquittement de la seconde baie avant que la requête d'E/S suivante puisse être traitée. Des liens dédiés sont ainsi recommandés et certains constructeurs n'hésitent pas à recommander de la fibre optique 8 Gbit/s au minimum pour ces derniers. Si les liens en question n'offrent pas un débit suffisant, les performances du système de stockage peuvent ainsi être impactées. C'est également pour cette raison que les distances entre les centres de données qui contiennent les baies répliquées ne peuvent pas être aussi importantes qu'elles pourraient l'être en cas de réplication asynchrone.

Il est bien entendu possible de ne répliquer de manière synchrone qu'une partie des données, cette technologie étant coûteuse à déployer. Considérons le cas d'un SGBD, le système étant installé sur un premier volume et les données sur un second. Il est dès lors tout à fait envisageable de répliquer le système de manière asynchrone et les données de manière synchrone.

7.1.1.2 Réplication asynchrone

Ce mode de réplication ne transmet pas les données en temps réel mais à intervalle régulier, se contentant dès lors d'une bande passante moins importante. Il s'agit d'une solution idéale pour un contexte de PRA par le biais d'un site distant.

L'augmentation du trafic sur la ligne dévolue à la réplication ou la chute du débit de cette dernière n'influe pas sur les performances, ce type de réplication n'étant jamais en attente de quelconque acquittement.

7.1.2 Protocole de réplication

Les protocoles évoqués ci-dessous sont liés au réseau de stockage basé sur la fibre optique (cf. section 6.1.4.1, Fibre optique). En effet, la mise en œuvre de la réplication entre deux baies de stockage fonctionnant sur la base du protocole iSCSI est relativement simple. Les commandes SCSI étant transportées, dans ce cas de figure, sur un réseau TCP/IP, il suffit de configurer un VPN entre les deux baies pour qu'elles puissent communiquer.

7.1.2.1 FCIP

Le Fiber Channel over IP, également appelé Fibre Channel Tunneling ou Storage Tunneling, permet de relier deux SAN distants. Ce protocole est destiné à la mise en place de la réplication sur des distances importantes.

7.1.2.2 iFCP

Internet Fiber Channel Protocol se trouve être un protocole routable. Deux réseaux peuvent ainsi être connectés entre eux. Il est également possible de connecter des périphériques dotés d'une connectique FC à un SAN IP, en passant par un réseau TCP/IP.

7.1.3 Protection continue

Le CDP (*Continuous Data Protection*) est une sauvegarde en continu dont le mécanisme s'apparente à la réplication synchrone, à ceci près qu'une nouvelle version comportant les changements est écrite vers le stockage distant. Le bloc ou le fichier modifié ne remplace donc pas l'ancienne version comme c'est le cas avec la réplication synchrone. La restauration n'est donc pas limitée à la dernière version en date du document mais à n'importe laquelle des versions disponibles.

Dans ce contexte, le RPO est donc quasiment nul, puisque les versions peuvent être restaurées instantanément, sans perte de données. Dans un environnement protégé par des sauvegardes traditionnelles, toutes les données écrites entre la dernière sauvegarde et le désastre auraient été définitivement perdues.

Contrairement à la réplication, le CDP permet de protéger les données de toute forme de changement ou de corruption accidentelle (éventuellement provoquée par l'humain) ou causée par un *malware* quelconque, puisqu'il est possible de restaurer une version précédente des données en question.

Deux variantes de ces solutions sont disponibles sur le marché, s'agissant du True CDP et du Near Continuous DP. La première journalise chaque écriture en mode bloc, permettant dès lors la restauration d'une version correspondant au moment choisi dans le temps, et ce, pratiquement à l'infini. La deuxième est planifiée en fonction d'un intervalle défini (par exemple, toutes les 10 minutes). Ici également, l'importance des données à sauvegarder détermine le type de technologie à adopter.

Le True CDP est destiné à effectuer des sauvegardes dites consistantes. Il convient de considérer que les données présentes dans le cache de la baie doivent être vidées (*flush*) vers les piles RAID pour que True CDP puisse en avoir connaissance. Or, il n'est pas possible de vider le cache en permanence. De plus, True CDP capture chaque écriture sans se soucier de savoir si cette dernière présente un état cohérent, contrairement à Near Continuous DP qui fonctionne sur la base des *snapshots* (cf. section 7.1.5, Snapshot) obtenus lors de chaque intervalle. Il faut donc être particulièrement vigilant à ce genre de détail lors de l'acquisition d'une solution basée sur ce principe.

7.1.4 Déduplication

La déduplication est incontournable lorsqu'il s'agit de limiter l'espace toujours plus conséquent occupé par les données.

Cette technique consiste à factoriser à l'aide d'un algorithme *ad hoc* des séquences de données identiques afin d'économiser l'espace disque utilisé. Chaque fichier est découpé en une multitude de tronçons. À chacun de ces tronçons est associé un identifiant unique, ces derniers étant stockés dans un index. Ainsi, un même tronçon n'est stocké qu'une seule fois. Une nouvelle occurrence d'un tronçon déjà présent n'est pas sauvegardé à nouveau mais remplacé par un pointeur vers l'identifiant correspondant.

La déduplication est très efficace en environnement virtuel. En effet, les machines virtuelles déployées sont souvent pilotée par le biais de la même version d'un système d'exploitation. Si l'infrastructure comprend une cinquantaine de serveurs virtuels Windows Server 2012, une quantité prodigieuse de fichiers (tels que des DLL) seront identiques. Ainsi, déployer les serveurs sur la base d'un modèle peut s'avérer très utile en matière déduplication, puisque les nombreux fichiers similaires seront parfaitement pris en compte par l'algorithme de déduplication.

Si la déduplication permet des économies en matière d'espace disque, le stockage des données, notamment les sauvegardes, nécessitera moins de disques durs. Par extension, la consommation électrique et les besoins en climatisation sont réduits, la quantité de U nécessaire au sein du centre de données diminue et la fenêtre de sauvegarde est restreinte.

7.1.4.1 Bloc fixe

Cette méthode consiste à :

- Couper les données en blocs de taille fixe, cette dernière ayant été prédéfinie (au moins 4 Ko) ;
- Créer un *hash* correspondant à chaque bloc ;
- Stocker le bloc y relatif et ajouter l'identifiant ainsi créé dans l'index prévu à cet effet ;
- Si l'identifiant en question y figure déjà, le bloc est remplacé par un simple pointeur faisant référence au bloc de données original.

7.1.4.2 Bloc variable

La déduplication à bloc variable vise à améliorer le processus décrit à la section précédente. En effet, dans le cas de la déduplication à bloc fixe, la recherche d'un bloc similaire est effectuée à partir du début de chaque bloc, soit, dans notre exemple, tous les 4 Ko. Dans le cas de la déduplication à bloc variable, le découpage des données effectué par l'algorithme

produit des blocs de taille variable. Puisque ces derniers ne sont plus alignés, la recherche est effectuée octet par octet, avec pour corollaire un taux de déduplication nettement plus important.

Considérons, à titre d'exemple, qu'une légère modification affecte le début d'un fichier. Ce dernier n'est pas traité dans sa totalité, comme il l'aurait été dans le cas d'une déduplication à bloc fixe.

Il convient de préciser que cette méthode requiert une puissance de calcul supérieure à celle nécessaire pour la déduplication à bloc de taille fixe. C'est la raison pour laquelle la comparaison est généralement effectuée après la sauvegarde afin de préserver les performances (post-processus). Ce n'est pas le cas lors d'une déduplication à bloc de taille fixe dont la norme est *in-line*. Dans ce cas, la comparaison est effectuée durant la sauvegarde, ce qui nécessite des capacités accrues en matière de cache.

7.1.5 Snapshot

Les *snapshots,* ou instantanés, sont des copies en lecture seule correspondant à l'état d'un système (des données y relatives) à un instant précis. Par état, nous entendons un état consistant des données à un instant T.

Il ne s'agit pas d'une sauvegarde à part entière puisque les données initiales sont nécessaires pour restaurer un système à partir d'un instantané. Ce cliché peut toutefois être restauré à un autre emplacement que celui d'origine, ce qui peut s'avérer intéressant pour créer un clone d'une machine virtuelle ou pour effectuer des tests applicatifs.

L'instantané est un excellent moyen de sécuriser les données et constitue d'ailleurs la base de certaines stratégies de sauvegarde. En effet, il est tout à fait possible de prendre des instantanés à intervalle régulier durant la journée et de les externaliser sur un support de stockage distant (un NAS, par exemple). Cette méthode permet d'approcher un RPO d'une valeur de zéro.

Une telle méthode s'apparente à de la pseudo-réplication. Un instantané de l'ensemble d'un volume peut en effet être déclenché, ce dernier étant copié sur un autre stockage. L'espace disque ainsi consommé ne représente que la différence entre l'état des données au moment du déclenchement du *snapshot* et l'instantané précédent, puisque nous nous situons dans un contexte de sauvegarde incrémentale. La bande passante requise demeure dès lors minime. En conséquence, cette manière de faire ressemble à de la réplication asynchrone.

Cette technologie peut être mise en œuvre au niveau logiciel (notamment par les solutions de virtualisation) ou au niveau des baies de stockage.

Il convient également de préciser que les instantanés, exploités conjointement par les API et autres agents commercialisés par les éditeurs, permettent de mettre en œuvre des procédures de sauvegarde des données alors même que le modèle d'affaire n'autorise aucune fenêtre de sauvegarde (par exemple, dans le cas où la production n'est jamais interrompue). Il est en effet possible de déclencher un instantané en cours de production, même si des fichiers concernés par la sauvegarde sont ouverts. La machine virtuelle peut

être rapidement et intégralement restaurée à son emplacement d'origine, avec pour corollaire la diminution de la valeur du RTO.

7.2 Postproduction

7.2.1 Préservation d'une infrastructure fonctionnelle

La gestion quotidienne d'une infrastructure virtuelle exige des administrateurs systèmes qu'ils soient parfaitement conscients de l'état de ses divers éléments, et ce, en permanence. Les applications métier sont hébergées par des serveurs virtuels, la couche de virtualisation s'immisçant entre eux et le matériel sous-jacent. Ce matériel, à savoir les serveurs hôte de machines virtuelles fonctionnent désormais en *cluster*. Les ressources physiques nécessaires aux VM étant elles-mêmes issues de l'entier de ce *cluster*, identifier les serveurs (virtuels) devient sensiblement plus difficile, à tout le moins visuellement. Il en va de même pour les stations de travail, lorsque ces dernières sont également virtuelles, et pour les systèmes de stockage, en particulier lorsque le *pool* est composé de différentes solutions commercialisées par différents fabricants.

Ainsi, certaines fonctionnalités inhérentes à la virtualisation, telles que la migration de machines virtuelles actives entre les différentes hôtes ou l'obtention d'un *pool* unique de stockage à partir de systèmes hétérogènes, peuvent être perçues comme des sources additionnelles de complexité. Or, il n'en est rien, pour autant que les administrateurs systèmes concernés prennent la mesure des outils de **supervision** qui deviennent primordiaux dans un environnement virtuel. Dès lors que le matériel s'efface quelque peu derrière la couche de virtualisation, ces derniers s'avèrent être des outils privilégiés pour vérifier que toute l'infrastructure fonctionne parfaitement.

Une fois les outils de supervision maitrisés, des fonctionnalités comme vMotion™[116] de VMware ou comme Distributed Resource Scheduler[117] (DRS) de la même société, contribueront à rendre l'infrastructure agile. Il s'agit tout de même de l'un des buts que nous recherchons dans la virtualisation.

En effet, vMotion™ permettant le déplacement instantané de machines virtuelles en cours d'exécution, il devient possible d'effectuer des migrations « à chaud », sans interruption pour les utilisateurs. Les opérations de maintenance matérielle d'un serveur hôte ne dérangeront guère plus lesdits utilisateurs. Le déplacement des VM d'un hôte à un autre peut même être automatisé par DRS, et ce, dans un souci d'équilibrage de la puissance de calcul. La consommation d'énergie inhérente au centre de données peut enfin être réduite par le biais de solutions telles que VMware vSphere Distributed Power Management™ (DPM) qui est capable d'optimiser cette consommation en continu, en fonction des besoins des VM (qui sont moindres la nuit ou le week-end, par exemple).

Grâce aux outils de supervision, il est possible de connaître en permanence l'état du matériel (température, alimentation, composants) ou vérifier que les performances offertes soient en conformité avec les attentes (utilisation des processeurs, espace disque et mémoire

[116] http://www.vmware.com/fr/products/datacenter-virtualization/vsphere/vmotion.html.
[117] http://www.vmware.com/fr/products/datacenter-virtualization/vsphere/drs-dpm.html.

disponibles, comportement des VM à ce niveau). Il est bien souvent possible de savoir si tous les services, ainsi que le réseau, sont opérationnels.

Ces outils sont généralement fournis par les éditeurs ou les constructeurs. Seule leur prise en main est requise de la part des administrateurs systèmes. Toutefois, si l'infrastructure est particulièrement hétérogène, une combinaison de solutions attend ces derniers. Il convient de préciser, à ce propos, que l'outil d'**orchestration** semble être la solution préconisée pour pallier, entre autres buts, ce problème (cf. section 7.2.4, Perspectives).

7.2.2 Récupération après catastrophe

Lorsqu'une reprise après sinistre doit être planifiée, le pire des scénarios est généralement retenu, à savoir la perte complète d'un centre de données. Les notions de PRA et de RTO sont donc intimement liées à un tel événement.

Le plan de reprise d'activité (*Disaster Recovery Plan* en anglais) est donc un plan d'urgence avalisé par les instances dirigeantes, résultant d'une réflexion dépassant le cadre strict du système d'information. Il s'agit de faire en sorte que l'activité principale de l'entreprise, autrement dit celle qui génère des revenus, puisse reprendre au plus vite. Or, nous pouvons considérer qu'à l'heure actuelle, une telle activité est souvent indissociable du système d'information.

Un plan de reprise d'activité consiste habituellement à mettre en place un système de relève sur lequel il est possible de basculer.

Dans le cadre d'un environnement virtualisé, il s'agit généralement de mettre en place une réplication asynchrone entre le système de stockage présent dans le premier centre de données et son pendant, situé sur un site distant. En cas de sinistre, la deuxième baie peut être configurée manuellement en tant que nouvelle baie principale. Les données ayant été répliquées sur cette dernière, l'activité est en mesure de reprendre. De plus, un nombre adéquat de serveurs hôte de machines virtuelles doit également être disponible au sein du deuxième centre de données. Ces derniers doivent être en mesure de supporter seuls la charge imposée par la totalité des machines virtuelles et doivent faire partie du même cluster que les serveurs du centre de données principal. Une technologie telle que VMware vSphere High Availability[118] (HA) doit avec été mise en œuvre. Ainsi, les machines virtuelles peuvent être redémarrées sur les serveurs hôte du site distant.

Il existe cependant plusieurs niveaux possibles de basculement. Ces niveaux dépendent des besoins exprimés par les instances dirigeantes et ces derniers sont établis en fonction du RTO et du RPO. Un système de basculement permettant d'atteindre des valeurs de RTO et de RPO quasiment nulles peut coûter passablement cher. Il s'agit dès lors d'évaluer les besoins avec sagacité.

En effet, la mise en œuvre d'un PCA s'avère être encore plus onéreuse que celle d'un PRA. Il ne s'agit plus, dans ce cas, de la remise en activité d'une infrastructure mais de faire en sorte que les utilisateurs puissent fonctionner en toute transparence en cas de sinistre.

[118] http://www.vmware.com/fr/products/datacenter-virtualization/vsphere/high-availability.html.

Si nous faisons référence au scénario décrit ci-dessus en le transposant dans un contexte de PCA, quelques mesures doivent d'ores et déjà être prises. En l'occurrence, les baies devraient être configurées de telle sorte qu'une réplication synchrone soit en vigueur. De plus, le basculement d'une baie à l'autre devrait être automatisé. Il en irait d'ailleurs de même pour la solution chargée d'assurer la disponibilité des serveurs. En effet, vSphere High Availability devraient être abandonné au profit de vSphere Fault Tolerance[119] (FT). Ainsi, des instances fantômes des machines virtuelles seraient déjà disponibles sur les serveurs du centre de données distant, autorisant dès lors un basculement automatique des instances principales des VM vers ces dernières.

Nous précisons que certains outils destinés à faciliter la mise en place de PRA sont disponibles sur le marché. Le basculement des applications critiques vers un site de secours peut être facilité, voire automatisé par le biais de ces derniers. Il s'agit notamment de **VMware vCenter Site Recovery Manager**[120] (SRM) et de **Double-Take®**.[121].

VMware SRM permet également de réaliser des tests relatifs au PRA sans impacter la production, afin de pouvoir s'assurer de la conformité des processus. Il permet en outre de définir avec précision les machines virtuelles qui doivent être maintenues à tout prix et de définir l'ordre de leur démarrage, d'exécuter des actions basées sur des scripts, de modifier automatiquement la configuration de certaines machines virtuelles en fonction des exigences relatives au site de secours, etc.

7.2.3 Planification budgétaire

Plus que jamais dans l'histoire des entreprises, les aspects financiers font l'objet d'examens attentifs. Les opportunités de retour sur investissement offertes par la virtualisation ont déjà été évoquées au sein de ce document, en particulier au chapitre 3, Bénéfices de la virtualisation, la hiérarchie s'intéressant tout particulièrement à cet aspect pour débloquer le budget y relatif.

Une fois l'infrastructure disponible, il convient de tenir compte des frais d'exploitation de cette dernière. Nous faisons notamment référence aux frais engendrés par la consommation électrique (qui devraient par ailleurs notoirement décroître), qu'elle soit générée par les dispositifs ou par leur refroidissement.

Le matériel doit, quant à lui, être amorti, au même titre qu'il le serait dans un environnement classique. Pour les serveurs hôte, cet amortissement est habituellement basé sur leur durée de vie qui est estimée à cinq ans au maximum. Il peut être toutefois judicieux de se baser sur le laps de temps durant lequel le constructeur garantit le matériel.

Les licences correspondant aux solutions de virtualisation fournies par les éditeurs doivent également être prises en compte. Bien que généralement non limitées dans le temps, il peut s'avérer obligatoire de les faire évoluer en cas de changement du matériel. Le prix de la licence d'un hyperviseur est en effet calculé, la plupart du temps, en fonction du nombre de

[119] http://www.vmware.com/fr/products/datacenter-virtualization/vsphere/fault-tolerance.html.
[120] http://www.vmware.com/products/site-recovery-manager/overview.html.
[121] http://www.visionsolutions.com/world/francaiseu/francaiseu.aspx.

processeurs. Si les capacités des serveurs hôte de machines virtuelles nouvellement acquis augmentent, le prix des licences pourraient dès lors également s'avérer plus important.

Si tous ces coûts peuvent aisément être estimés, ce n'est pas forcément évident de les répartir par département. De plus, s'il est possible d'estimer le coût du matériel, il n'est pas évident d'estimer le coût des machines virtuelles elles-mêmes. Or, ces dernières ne sont pas gratuites, même si leur coût est infiniment moins important que celui d'une machine physique. De plus, les besoins en matière de machines virtuelles par département sont parfois nécessaires pour répartir avec précision les budgets informatiques sur l'ensemble de l'entreprise.

VMware® propose d'ailleurs une solution qui permet de pallier ce manque de transparence, s'agissant en l'espèce de **Chargeback Manager**[122]. Cette dernière permet de modéliser les coûts réels associés aux machines virtuelles en donnant la possibilité aux administrateurs de configurer les coûts de base, les coûts fixes, les coûts de mise en œuvre, ainsi que plusieurs facteurs de taux. Elle offre également la possibilité d'assurer le suivi des coûts en matière d'alimentation, de refroidissement ou de licences logicielles notamment. Les gestionnaires des différents départements peuvent bénéficier de rapports périodiques comprenant les données relatives à l'utilisation et aux coûts.

7.2.4 Perspectives

Si nous nous référons aux achats récents de Xsigo®[123] par Oracle®, le 30 juillet 2012, et de son concurrent Nicira®[124] par VMware® sept jours plus tôt, la virtualisation du réseau semble faire office de pierre angulaire dans la stratégie de développement des acteurs les plus importants du monde de la virtualisation.

En effet, tant les solutions proposées par Nicira® que celles commercialisées par Xsigo® sont reposent sur le concept de ***Software Defined Networking*** (SDN), soit un nouveau paradigme d'architecture réseau ayant pour particularité de découpler le plan de contrôle du plan de données. Une fois virtualisé, le réseau physique est utilisé uniquement pour la transmission de paquets, s'agissant dès lors simplement d'un fond de panier IP. Les réseaux virtuels sont créés par le biais d'un programme et fonctionnent de manière totalement découplée du matériel sous-jacent, en offrant les mêmes caractéristiques et les mêmes garanties qu'un réseau physique. Un contrôleur externe peut ainsi diriger toute l'infrastructure de commutation et/ou de routage, le réseau physique se transformant alors en un pool de ressources réseau qui peuvent être dynamiquement livrées aux clients en fonction des besoins.
Après la virtualisation des serveurs et du stockage, le SDN ouvre la porte à la virtualisation complète du centre de données ou Software Defined Datacenter (SDDC), avec pour corollaire la simplification des centres de calculs basée sur la virtualisation.

VWware® fait d'ailleurs référence au Virtual Datacenter (VDC) et matérialise sa vision par le biais de **vCloud® Suite**[125], solution permettant la création d'infrastructure de type nuage,

[122] http://www.vmware.com/fr/products/datacenter-virtualization/vcenter-chargeback/overview.html.
[123] http://www.xsigo.com/index.php.
[124] http://nicira.com/.
[125] http://www.vmware.com/products/datacenter-virtualization/vcloud-suite/overview.html.

complète et intégrée. Cette suite s'appuie évidemment sur l'hyperviseur vSphere™, mais également sur **vCloud™ Director**, véritable solution d'orchestration permettant de créer et de fournir des centres de données virtuels.

Ainsi, les concepts de **SDDC** et de solution d'**orchestration** constituent manifestement le socle d'un nouveau paradigme en matière de mise en place et de gestion des centres de données et devraient *a priori* faire parler d'eux à l'avenir, en particulier dans l'élaboration d'un nouveau modèle de Cloud, plus en phase avec les attentes formulées par les entreprises.

En effet, la virtualisation a initialement été adoptée par de nombreuses entreprises pour réduire leurs dépenses grâce à la consolidation des serveurs et pour diminuer les charges d'exploitation grâce à l'automatisation. La diminution des pertes engendrée par une meilleure gestion des interruptions de service planifiées comme non planifiées a également fortement contribué à cette adoption.

Aujourd'hui, à l'heure où les différents départements formant une entreprise sollicitent les équipes informatiques pour obtenir un accès toujours plus rapide aux ressources informatiques et aux applications, la virtualisation paraît être la seule technologie capable d'apporter aux administrateurs systèmes la réactivité attendue. Or, la mise en place de solutions Cloud privées ou hybrides offrent des possibilités intéressantes pour relever le défi.

Il devient en effet possible de provisionner des machines virtuelles et des applications multiniveaux avec célérité, en augmentant ainsi la capacité mise à disposition des divisions de l'entreprise tant sur site que hors site. Le déploiement des ressources aura dès lors de plus en plus tendance à se faire à la demande. Nous invitons d'ailleurs le lecteur à s'intéresser de près aux notions de SaaS (*Software as a Service*) ou de PaaS (*Platform as a Service*) dont l'évolution est intimement liée à la virtualisation.

La virtualisation des applications permet d'ores et déjà d'encapsuler ces dernières dans leur propre « bulle » applicative. À l'avenir, il s'agira d'être capable de les distribuer par le biais du Cloud, via des plateformes ad hoc, afin de répondre à la forte demande en matière d'accès mobile aux applications. VMware répond d'ores et déjà à cette demande en proposant une solution permettant de créer un catalogue applicatif et gérer les espaces de travail sécurisés sur les terminaux mobiles (Horizon Application Manager™ et Horizon Mobile™). Cette solution a d'ailleurs fait l'objet d'une annonce au VMworld 2012.

De plus, les utilisateurs, hormis le fait qu'ils soient de plus en plus mobiles, ont tendance à vouloir utiliser leurs dispositifs personnels dans le cadre de leur travail. Cette tendance est connue sous l'appellation BYOD (*Bring your own device*) ou de « consumérisation » de l'IT. VMware® propose, à ce titre, de coupler sa solution View™ à l'offre Mirage™ de Wanova®[126], une société rachetée en mai 2012. Ainsi, la synchronisation et la gestion des applications sur différents types de postes de travail physiques ou virtuels deviennent possibles.

[126] http://www.wanova.com/.

Les éditeurs de solutions de virtualisation confirment donc cette tendance consistant à virtualiser le poste de travail et manifeste un intérêt certain pour la gestion de la mobilité.

Quant au modèle PaaS, il suggère que le fournisseur maintienne la plateforme d'exécution des applications (au niveau du Cloud) tandis que l'entreprise cliente conserve la possibilité de maintenir ses propres applications. Il s'agit donc bien d'un nouveau modèle d'exploitation des ressources informatiques, basé sur le Cloud et indirectement sur l'existence de la virtualisation.

Ces concepts constituent toutefois une tendance et ne sauraient être considérés comme pleinement fonctionnels à l'heure où nous rédigeons ces lignes. De plus, le degré d'acceptation de ces technologies par les professionnels de la branche est encore mesuré.

8 Conclusion

À l'origine de l'informatique, nous disposions d'ordinateurs capables de faire fonctionner un unique processus. Il nous a alors semblé opportun de faire en sorte que plus d'un seul programme puissent y être exécutés. Les premiers systèmes multitâches virent ainsi le jour. Ces systèmes étant passablement coûteux, nous nous sommes dès lors mis en tête de pouvoir partager leur puissance de calcul entre plusieurs utilisateurs, désir qui donna naissance au pseudo-parallélisme. En effet, les ordinateurs étaient souvent inactifs, puisque l'utilisateur était régulièrement occupé à d'autres tâches. Or, il était tentant d'utiliser les centaines de milliers cycles disponibles à ce moment-là pour en faire bénéficier un autre utilisateur et rentabiliser ainsi au mieux l'investissement que représentait cet ordinateur.

Nous l'avons évoqué au sein de ce document, la virtualisation est née du désir de partager la puissance de calcul des ordinateurs. Elle a par la suite permis de réaliser de sérieuses économies en consolidant un nombre important de machines virtuelles sur un nombre restreint de serveurs physiques. Tout comme les environnements à temps partagé avant elle, la virtualisation nous a permis d'améliorer drastiquement la rentabilisation du matériel informatique. En définitive, cette technologie a donné l'opportunité aux ingénieurs de pouvoir franchir l'étape qui devait logiquement succéder au temps partagé, à savoir le fait de pouvoir exécuter plusieurs systèmes d'exploitation sur une seule et même machine.

À l'heure actuelle, nous franchissons une nouvelle étape : faire abstraction de l'infrastructure pour permettre la mise à disposition des ressources informatiques aux utilisateurs comme autant de services. La virtualisation nous autorise à envisager un tel concept et tout semble indiquer que le système d'information est en train d'évoluer dans ce sens.

Le concept peut sembler séduisant car il suppose une simplification de la mise à disposition des ressources informatiques à l'attention des utilisateurs, que ces derniers se trouvent à l'interne ou en déplacement. Nous sommes cependant face à un véritable changement de paradigme. Il s'avère donc essentiel que les DSI prennent la mesure des mutations à venir et adaptent en conséquence leurs objectifs stratégiques, faute de quoi certaines entreprises pourraient passer à côté des bénéfices potentiellement réalisables ou éventuellement perdre la maîtrise de leur système d'information.

9 Glossaire

- ***Chunk*** : équivalent d'un bloc dans le monde du RAID. Le boc étant un ensemble de secteurs ;
- **Cluster** : grappe de serveurs (ou « ferme de calcul ») constituée de deux serveurs au minimum (appelé aussi nœud) et partageant une baie de disques commune, pour assurer une continuité de service et/ou repartir la charge de calcul et/ou la charge réseau ;
- **DSI** : Direction (ou Directeur) des systèmes d'information ;
- **Dispositif** : toute machine ou composant qui fait partie d'un ordinateur. Ces dispositifs requièrent, pour la plupart, un programme appelé *pilote* pour pouvoir fonctionner en adéquation avec l'ordinateur. Ce pilote agit comme un traducteur en convertissant les commandes générales d'une application en commandes spécifiques au dispositif en question ;
- ***Framework*** : kit de composants logiciels structurels qui sert à créer les fondations, ainsi que les grandes lignes de tout ou partie d'un logiciel. Ces outils et composants logiciels sont organisés conformément à un plan d'architecture et de patrons, l'ensemble formant un squelette de programme ;
- **Hachage** : Une fonction de hachage permet, à partir d'une donnée fournie en entrée, de calculer une empreinte servant à identifier rapidement, bien qu'incomplètement, la donnée initiale. Elle sert à rendre plus rapide l'identification des données ;
- **Hyperviseur** : plateforme de virtualisation permettant à plusieurs systèmes d'exploitation de travailler sur une machine physique en même temps ;
- **Interruption** : en informatique, une interruption est un arrêt temporaire de l'exécution normale d'un programme par le microprocesseur afin d'exécuter un autre programme ;
- **IOPS** : de l'anglais Input/Output Operations Per Second pour opérations d'entrées/sorties par seconde qui est une mesure commune de performances ;
- **ISA** : pour *Instruction Set Architecture*, s'agissant de la spécification fonctionnelle d'un processeur, du point de vue du programmeur, en langage machine. Cet ensemble d'instructions est codé au niveau du processeur et n'est pas modifiable. Il définit les capacités d'un processeur dont l'architecture matérielle est ensuite optimisée pour exécuter les instructions de l'ISA le plus efficacement possible ;
- **ITIL** : *Information Technology Infrastructure Library* pour Bibliothèque pour l'infrastructure des technologies de l'information qui est un ensembles d'ouvrages recensant les bonnes pratiques de la gestion des systèmes d'information. Ce référentiel aborde différents sujets, tels que l'organisation d'un système d'information, l'amélioration de son efficacité, la réduction des risques qui lui sont inhérents, ainsi que l'amélioration de la qualité des services informatiques ;
- **LAN** : de l'anglais *Local Area Network* ou réseau local. Ce réseau désigne par abus un réseau d'entreprise. Il s'agit plus précisément d'un réseau comprenant des dispositifs qui communiquent entre eux au niveau 2 du modèle OSI, soit par le biais de trames. Le protocole IP (Internet Transport) n'est pas nécessaire à cet échelon ;
- **LUN** : pour *Logical Unit Number*. Dans un réseau SAN, un LUN est le numéro d'identification d'un espace de stockage présenté à un ou plusieurs serveurs ;
- ***Mainframe*** : terme trouvant son origine dans l'informatique des années soixante, s'agissant d'un ordinateur ultra-performant, créé pour effectuer des traitements basés

sur un volume conséquent de données et exigeant de grandes capacités de calcul. Ces systèmes étaient capables de traiter des requêtes provenant de dizaines, voire de centaines de terminaux simultanément. Ils fonctionnent dès lors selon un modèle centralisé et sont à même de faire fonctionner de façon simultanée plusieurs sessions d'un système d'exploitation ou éventuellement de systèmes d'exploitation différents ;

- **Mémoire vive** : appelée également RAM (de l'anglais *Random Access Memory*), est la mémoire dans laquelle un ordinateur place les données lors de leur traitement. Cette mémoire est rapide d'accès et volatile (les données qui y sont contenues sont perdues une fois que l'ordinateur n'est plus alimenté en électricité) ;
- ***Monitoring*** : anglicisme qualifiant une activité de surveillance et de mesure d'une activité informatique ;
- **Mot** : en informatique, le mot est l'unité de base que manipule un processeur. Sa taille est exprimée en bits ou en octets et est souvent utilisée pour classifier les microprocesseurs (32 bits, 64 bits, etc.). Un microprocesseur est d'autant plus rapide que ses mots sont longs, car les quantités de données qu'il traite à chaque cycle sont plus importantes ;
- ***Multipathing*** : technique permettant de mettre en place plusieurs routes d'accès entre un serveur et ses disques situés au sein du SAN, en diminuant ainsi les risques de perte d'accès ;
- **Multiprogrammation** : possibilité offerte, pour un système d'exploitation, de faire coexister plusieurs programmes simultanément en mémoire et de mettre à profit les temps d'inactivité relatifs à un programme donné, pendant les opérations d'entrées-sorties notamment, et ce, afin de faire progresser l'exécution d'autres programmes ;
- **NAS** : de l'anglais *Network Attached Storage*, pour serveur de stockage en réseau, s'agissant d'un serveur de fichiers autonome, relié à un réseau et dont la principale fonction est le stockage de données en un volume centralisé pour des clients réseau hétérogènes ;
- **NIC** : de l'anglais *Network Interface Card*, soit une carte réseau en français ;
- **Portabilité** : désigne, pour un programme informatique, sa capacité à être porté pour fonctionner plus ou moins facilement dans différents environnements d'exécution ;
- ***Provisioning :*** terme utilisé en informatique pour désigner l'affectation plus ou moins automatisée de ressources aux utilisateurs. Pour parvenir à cette allocation automatique de ressources, des outils de gestion de la configuration sont utilisés. Ces derniers permettent notamment d'installer et de configurer des logiciels à distance, d'allouer de l'espace disque, de la puissance ou de la mémoire ;
- **SAN** : de l'anglais *Storage Area Network* et qui implique une mutualisation des ressources de stockage au travers d'un réseau spécialisé permettant un accès bas niveau aux disques utilisés ;
- ***Serialization*** : processus visant à coder l'état d'une information présente en mémoire sous la forme d'une suite d'informations plus petites (ou atomique) qui se traduit le plus souvent sous forme d'octets, voire de bits ;
- **SMP** : de l'anglais *Symmetric Multiprocessing*, s'agissant d'une architecture informatique dite parallèle qui consiste à multiplier des processeurs au sein d'un ordinateur, de telle manière à augmenter la puissance de calcul ;
- **SPOF** : de l'anglais *Single Point of Failure*, ou point unique de défaillance ;

- **Temps partagé** : appelé également pseudo-parallélisme. Il s'agit d'une approche permettant de simuler le partage pour plusieurs utilisateurs du temps processeur ;
- **U** : Le U, pour unité, est une unité de mesure employée pour décrire la hauteur d'un dispositif au sein d'un rack. Un U correspond à 1 ¾ pouce, soit 44,45 millimètres ;
- **WAN** : de l'anglais Wide Area Network ou réseau étendu. Ce réseau couvre une vaste zone géographique et utilise l'Internet et le protocole IP (Internet Protocol), situé au niveau 3 de modèle OSI. Nous pouvons considérer que plusieurs LAN sont interconnectés par le biais du WAN ;
- **VM** : de l'anglais, *Virtual Machine* pour machine virtuelle ;
- **VMM** : de l'anglais, *Virtual Machine Manager* pour Gestionnaire de machine virtuelle ou plus généralement un hyperviseur (cf. Section 4.4.1.1, Hyperviseur).

10 Bibliographie

1. **Wikipedia.** Full system simulator. *en.wikipedia.org.* [En ligne] 22 avril 2012. [Citation : 28 juillet 2012.] http://en.wikipedia.org/wiki/Full_system_simulator.
2. **Wikipédia.** Hyperviseur. *fr.wikipedia.org.* [En ligne] 19 juillet 2012. [Citation : 28 juillet 2012.] http://fr.wikipedia.org/wiki/Hyperviseur.
3. **Wikipedia.** SimOS. *en.wikipedia.org.* [En ligne] 16 décembre 2011. [Citation : 28 juillet 2012.] http://en.wikipedia.org/wiki/SimOS.
4. —. VM (operating system). *en.wikipedia.org.* [En ligne] 12 juillet 2012. [Citation : 28 juillet 2012.] http://en.wikipedia.org/wiki/VM_(operating_system).
5. **La rédaction du magazine l'Informaticien.** VMware : Une réussite loin d'être virtuelle. *www.linformaticien.com.* [En ligne] 1 décembre 2010. [Citation : 28 juillet 2012.] http://www.linformaticien.com/dossiers/les-saga-de-lit/id/20086/vmware-une-reussite-loin-d-etre-virtuelle.aspx.
6. **Wikipédia.** Conversation Monitor System. *fr.wikipedia.org.* [En ligne] 8 octobre 2010. [Citation : 23 juillet 2012.] http://fr.wikipedia.org/wiki/Conversation_Monitor_System.
7. **Wikipedia.** CP/CMS. *en.wikipedia.org.* [En ligne] 6 février 2012. [Citation : 23 juillet 2012.] http://en.wikipedia.org/wiki/CP/CMS.
8. **The Computer Language Company Inc.** Definition of CP/CMS. *www.pcmag.com.* [En ligne] 2012. [Citation : 23 juillet 2012.] http://www.pcmag.com/encyclopedia_term/0,1237,t=CPCMS&i=58807,00.asp.
9. **TechTerms.com.** Hardware Terms, Mainframe. *site Web TechTerms.com.* [En ligne] 2012. [Citation : 23 juillet 2012.] http://www.techterms.com/definition/mainframe.
10. **Wikipedia.** History of CP/CMS. *en.wikipedia.org.* [En ligne] 21 juillet 2012. [Citation : 23 juillet 2012.] http://en.wikipedia.org/wiki/History_of_CP/CMS.
11. **Wikipédia.** Ordinateur central. *fr.wikipedia.org.* [En ligne] 23 juillet 2012. [Citation : 23 juillet 2012.] http://fr.wikipedia.org/wiki/Ordinateur_central.
12. **VMware, Inc.** Principes de base de la virtualisation. *Site Web VMware, Inc.* [En ligne] 2012. [Citation : 23 juillet 2012.] http://www.vmware.com/fr/virtualization/virtualization-basics/history.html.
13. **Le Journal du Net.** Solution > Mainframe. *Le Journal du Net.* [En ligne] 2012. [Citation : 23 juillet 2012.] http://www.journaldunet.com/solutions/mainframe/.
14. **Wikipédia.** Temps partagé. *fr.wikipedia.org.* [En ligne] 30 mai 2012. [Citation : 25 juillet 2012.] http://fr.wikipedia.org/wiki/Temps_partag%C3%A9.
15. **Multics.** The IBM 360/67 and CP/CMS. *www.multicians.org.* [En ligne] 20 juillet 2012. [Citation : 23 juillet 2012.] http://www.multicians.org/thvv/360-67.html.
16. **IBM.** z/VM. *vm.ibm.com.* [En ligne] 13 avril 2012. [Citation : 25 juillet 2012.] http://www.vm.ibm.com/.
17. **Wikipédia.** Architecture de processeur. *fr.wikipedia.org.* [En ligne] 7 avril 2012. [Citation : 28 juillet 2012.] http://fr.wikipedia.org/wiki/Architecture_de_processeur.
18. **Mathieu Lamelot, Pierre Dandumont.** Les défis de la virtualisation. *tom's hardware, THE AUTHORITY ON TECH.* [En ligne] 15 février 2010. [Citation : 28 juillet 2012.] http://www.presence-pc.com/tests/virtualisation-Intel-AMD-512/5/.
19. **TechTarget.** SearchServerVirtualization, Definition, Virtualization. *searchservervirtualization.techtarget.com.* [En ligne] décembre 2010. [Citation : 29 juillet 2012.] http://searchservervirtualization.techtarget.com/definition/virtualization.
20. **Webopedia.** Virtualization. *www.webopedia.com.* [En ligne] 2012. [Citation : 29 juillet 2012.] http://www.webopedia.com/TERM/V/virtualization.html.
21. **Hess, Kenneth et Newman, Amy.** *Virtualisation en pratique.* Paris : Pearson Education France, 2010.
22. **Maillé, Eric et Mennecier, René-François.** *VMware vSphere 5 au sein du Datacenter.* St Herblain : Éditions ENI, 2012.

23. **Georgeot, Cédric.** *Bonnes pratiques, planification et dimensionnement des infrastructures de stockage et de serveur en environnement virtuel.* Paris : Books on Demand, 2011.
24. **Wikipédia.** Information Technology Infrastructure Library. *fr.wikipedia.org.* [En ligne] 30 juillet 2012. [Citation : 13 août 2012.] http://fr.wikipedia.org/wiki/Information_Technology_Infrastructure_Library.
25. —. RAID (informatique). *fr.wikipedia.org.* [En ligne] 3 août 2012. [Citation : 13 août 2012.] http://fr.wikipedia.org/wiki/RAID_(informatique).
26. —. Cluster. *fr.wikipedia.org.* [En ligne] 6 août 2012. [Citation : 14 août 2012.] http://fr.wikipedia.org/wiki/Cluster.
27. —. Déduplication. *fr.wikipedia.org.* [En ligne] 28 juin 2012. [Citation : 14 août 2012.] http://fr.wikipedia.org/wiki/D%C3%A9duplication.
28. —. Portabilité (informatique). *fr.wikipedia.org.* [En ligne] 13 juin 2012. [Citation : 15 août 2012.] http://fr.wikipedia.org/wiki/Portabilit%C3%A9_(informatique).
29. **Steve Gribble, Associate Professor, CSE.** CSE490H: Virtualization. *www.cs.washington.edu.* [En ligne] 2008. [Citation : 19 août 2012.] http://www.cs.washington.edu/education/courses/cse490h/08au/lectures/cse490_virtualization.pdf.
30. **Wikipédia.** Small Computer System Interface. *fr.wikipedia.org.* [En ligne] 2 août 2012. [Citation : 6 août 2012.] Small Computer System Interface.
31. —. Serveur de stockage en réseau. *fr.wikipedia.org.* [En ligne] 26 juillet 2012. [Citation : 6 août 2012.] http://fr.wikipedia.org/wiki/Serveur_de_stockage_en_r%C3%A9seau.
32. —. Logical Unit Number. *fr.wikipedia.org.* [En ligne] 25 juin 2012. [Citation : 6 août 2012.] http://fr.wikipedia.org/wiki/Logical_Unit_Number.
33. —. Unité de gestion mémoire. *fr.wikipedia.org.* [En ligne] 30 mai 2012. [Citation : 19 août 2012.] http://fr.wikipedia.org/wiki/Unit%C3%A9_de_gestion_m%C3%A9moire.
34. **R. Beuchat LAMI-EPFL, LMP-EIG.** DMA, Accès Direct en Mémoire Direct Memory Access. *lapwww.epfl.ch.* [En ligne] mai 1993. [Citation : 19 août 2012.] http://lapwww.epfl.ch/courses/embsys/docs/DMA.pdf.
35. **Wikipédia.** Algorithmes de remplacement des lignes de cache. *fr.wikipedia.org.* [En ligne] 27 juillet 2012. [Citation : 19 août 2012.] http://fr.wikipedia.org/wiki/Algorithmes_de_remplacement_des_lignes_de_cache#LRU_.28Least_Recently_Used.29.
36. —. Taux de multiprogrammation. *fr.wikipedia.org.* [En ligne] 16 mai 2012. [Citation : 20 août 2012.] http://fr.wikipedia.org/wiki/Taux_de_multiprogrammation.
37. —. Segmentation (informatique). *fr.wikipedia.org.* [En ligne] 26 mai 2012. [Citation : 20 août 2012.] http://fr.wikipedia.org/wiki/Segmentation_(informatique).
38. —. Mémoire paginée. *fr.wikipedia.org.* [En ligne] 9 juin 2012. [Citation : 20 août 2012.] http://fr.wikipedia.org/wiki/M%C3%A9moire_pagin%C3%A9e.
39. —. Mémoire virtuelle. *fr.wikipedia.org.* [En ligne] 28 juillet 2012. [Citation : 20 août 2012.] http://fr.wikipedia.org/wiki/M%C3%A9moire_virtuelle.
40. —. Mot (informatique). *fr.wikipedia.org.* [En ligne] 31 juillet 2012. [Citation : 20 août 2012.] http://fr.wikipedia.org/wiki/Mot_(informatique).
41. —. Mémoire vive. *fr.wikipedia.org.* [En ligne] 28 juillet 2012. [Citation : 20 août 2012.] http://fr.wikipedia.org/wiki/M%C3%A9moire_vive.
42. **Mignot, Gaël Le.** Systèmes d'exploitation, Gestion de la mémoire. [En ligne] 2007. [Citation : 20 août 2012.] http://cours.pilotsystems.net/cours-insia/cours-de-systemes-dexploitation-ing2-srt/memory.pdf.
43. **Wikipédia.** Commutation de contexte. *fr.wikipedia.org.* [En ligne] 30 mai 2012. [Citation : 20 août 2012.] http://fr.wikipedia.org/wiki/Commutation_de_contexte.
44. **Institut national des sciences appliquées de Toulouse.** Mode protégé. *http://www.insa-toulouse.fr.* [En ligne] 26 décembre 2009. [Citation : 20 août 2012.] http://etud.insa-toulouse.fr/~projet_tut_OS/w/Mode_prot%C3%A9g%C3%A9.
45. **Wikipédia.** RFLAGS. *fr.wikipedia.org.* [En ligne] 8 août 2012. [Citation : 21 août 2012.] http://fr.wikipedia.org/wiki/RFLAGS.

46. —. Anneau de protection. *fr.wikipedia.org.* [En ligne] 20 juin 2012. [Citation : 21 août 2012.] http://fr.wikipedia.org/wiki/Anneau_de_protection.
47. **Wikipedia.** Ring (computer security). *en.wikipedia.org.* [En ligne] 14 août 2012. [Citation : 21 août 2012.] http://en.wikipedia.org/wiki/Ring_(computer_security).
48. **PC Magazine.** Definition of : VM/386. *www.pcmag.com.* [En ligne] [Citation : 22 août 2012.] http://www.pcmag.com/encyclopedia_term/0,1237,t=VM386&i=54031,00.asp.
49. **Sandoz, Alain.** Concept de machine virtuelle. *www.epfl.ch.* [En ligne] 2007. [Citation : 22 août 2012.] http://lsrwww.epfl.ch/webdav/site/lsrwww/shared/Enseignement/SysExp07/chapitre%2005%20machine%20virtuelle.pdf.
50. **Wikipédia.** Histoire des ordinateurs. *fr.wikipedia.org.* [En ligne] 8 août 2012. [Citation : 22 août 2012.] http://fr.wikipedia.org/wiki/Histoire_des_ordinateurs#Troisi.C3.A8me_g.C3.A9n.C3.A9ration_.281963-1971.29.
51. **Wikipedia.** Popek and Goldberg virtualization requirements. *en.wikipedia.org.* [En ligne] 2 avril 2012. [Citation : 22 août 2012.] http://en.wikipedia.org/wiki/Popek_and_Goldberg_virtualization_requirements.
52. **Benkemoun, Antoine.** Virtualisation. *http://www.antoinebenkemoun.fr.* [En ligne] [Citation : 22 août 2012.] http://www.antoinebenkemoun.fr/virtualisation/.
53. **Wikipedia.** Binary translation. *en.wikipedia.org.* [En ligne] 7 avril 2012. [Citation : 23 août 2012.] http://en.wikipedia.org/wiki/Binary_translation.
54. —. x86 virtualization. *en.wikipedia.org.* [En ligne] 5 août 2012. [Citation : 23 août 2012.] http://en.wikipedia.org/wiki/X86_virtualization.
55. **Intel.** Intel Virtualization Technology. *www.intel.com.* [En ligne] 10 août 2006. [Citation : 27 août 2012.] http://www.intel.com/technology/itj/2006/v10i3/1-hardware/5-architecture.htm.
56. —. PCI-SIG SR-IOV Primer. *www.intel.com.* [En ligne] janvier 2011. [Citation : 1 septembre 2012.] http://www.google.ch/url?sa=t&rct=j&q=pci-sig%20i%2Fo&source=web&cd=10&cad=rja&ved=0CHcQFjAJ&url=http%3A%2F%2Fwww.intel.com%2Fcontent%2Fwww%2Fus%2Fen%2Fpci-express%2Fpci-sig-sr-iov-primer-sr-iov-technology-paper.html&ei=eilCUMn1MIjasgbfkYGoCQ&usg=AFQjCNGW.
57. **Wikipedia.** PCI-SIG. *en.wikipedia.org.* [En ligne] 8 août 2012. [Citation : 1 septembre 2012.] http://en.wikipedia.org/wiki/PCI-SIG.
58. **Wikipédia.** NX Bit. *fr.wikipedia.org.* [En ligne] 24 juin 2012. [Citation : 8 septembre 2012.] http://fr.wikipedia.org/wiki/NX_Bit.
59. **SearchServerVirtualization.** Blades v. rack servers for virtualization. *searchservervirtualization.techtarget.com.* [En ligne] octobre 2009. [Citation : 8 septembre 2012.] http://searchservervirtualization.techtarget.com/feature/Blades-vs-rack-servers-for-virtualization.
60. **Wikipédia.** Chemin de données. *fr.wikipedia.org.* [En ligne] 12 octobre 2011. [Citation : 19 août 2012.] http://fr.wikipedia.org/wiki/Chemin_de_donn%C3%A9es.
61. **Thierry Lévy-Abégnoli, indexel.net.** Commutateur virtuel : six questions sur un concept en pleine révolution. *www.indexel.net.* [En ligne] 10 mars 2010. [Citation : 5 septembre 2012.] http://www.indexel.net/materiels/commutateur-virtuel-six-questions-sur-un-concept-en-pleine-revolution-3052.html.
62. **Wikipedia.** Converged network adapter. *en.wikipedia.org.* [En ligne] 5 mars 2012. [Citation : 6 septembre 2012.] http://en.wikipedia.org/wiki/Converged_network_adapter.
63. **Wikipédia.** Framework. *fr.wikipedia.org.* [En ligne] 27 août 2012. [Citation : 2 septembre 2012.] http://fr.wikipedia.org/wiki/Framework.
64. —. Interruption (informatique). *fr.wikipedia.org.* [En ligne] 30 mai 2012. [Citation : 19 août 2012.] http://fr.wikipedia.org/wiki/Interruption_(informatique).
65. —. Jeu d'instructions. *fr.wikipedia.org.* [En ligne] 4 juillet 2012. [Citation : 19 août 2012.] http://fr.wikipedia.org/wiki/Jeu_d%27instructions.

66. **Anicet Mbida, 01net Entreprises.** La virtualisation d'applications. *pro.01net.com.* [En ligne] 24 août 2006. [Citation : 2 septembre 2012.] http://pro.01net.com/editorial/324015/la-virtualisation-dapplications/.
67. **Wikipédia.** Langage machine. *fr.wikipedia.org.* [En ligne] 6 novembre 2011. [Citation : 19 août 2012.] http://fr.wikipedia.org/wiki/Code_machine.
68. —. Processeurs. *fr.wikipedia.org.* [En ligne] 4 août 2012. [Citation : 19 août 2012.] http://fr.wikipedia.org/wiki/Processeur#Les_op.C3.A9rations_du_processeur.
69. —. Registre du processeur. *fr.wikipedia.org.* [En ligne] 4 juillet 2012. [Citation : 19 août 2012.] http://fr.wikipedia.org/wiki/Registre_de_processeur.
70. —. Réseau local virtuel. *fr.wikipedia.org.* [En ligne] 30 juillet 2012. [Citation : 5 septembre 2012.] http://fr.wikipedia.org/wiki/R%C3%A9seau_local_virtuel.
71. —. Serveur lame. *fr.wikipedia.org.* [En ligne] 6 février 2012. [Citation : 8 septembre 2012.] http://fr.wikipedia.org/wiki/Serveur_lame.
72. —. Surveillance (informatique). *fr.wikipedia.fr.* [En ligne] 21 juin 2012. [Citation : 5 septembre 2012.] http://fr.wikipedia.org/wiki/Surveillance_(informatique).
73. —. Unité arithmétique et logique. *fr.wikipedia.org.* [En ligne] 4 juillet 2012. [Citation : 19 août 2012.] http://fr.wikipedia.org/wiki/Unit%C3%A9_arithm%C3%A9tique_et_logique.
74. —. Unité de contrôle. *fr.wikipedia.org.* [En ligne] 4 juillet 2012. [Citation : 19 juillet 2012.] http://fr.wikipedia.org/wiki/Unit%C3%A9_de_contr%C3%B4le.
75. **Bonnin, Aurélien.** Virtulalisation d'application : App-V 4.6 SP1. *www.itpro.fr.* [En ligne] 11 juin 2012. [Citation : 2 septembre 2012.] http://www.itpro.fr/a/virtualisation-application-app-v-4-6-sp1/.
76. **Jeff.** VLAN - Réseaux virtuels. *www.commentcamarche.net.* [En ligne] 20 avril 2011. [Citation : 5 septembre 2012.] http://www.commentcamarche.net/contents/internet/vlan.php3.
77. **Wikipedia.** Fibre Channel over IP. *en.wikipedia.org.* [En ligne] 19 avril 2012. [Citation : 8 septembre 2012.] http://en.wikipedia.org/wiki/Fibre_Channel_over_IP.
78. —. Internet Fibre Channel Protocol. *en.wikipedia.org.* [En ligne] 20 juin 2011. [Citation : 8 septembre 2012.] http://en.wikipedia.org/wiki/Internet_Fibre_Channel_Protocol.
79. **Alberto Farronato, Group Product Marketing Manager for VMware Storage and Availability Products.** Memory Overcommit - Real life examples from VMware customers. *blogs.vmware.com.* [En ligne] 8 octobre 2008. [Citation : 9 septembre 2012.] http://blogs.vmware.com/virtualreality/2008/10/memory-overcomm.html.
80. **Emeriau, Jean-Baptiste.** La gestion de la mémoire dans ESX 4. *wattmil.dyndns.org.* [En ligne] 30 janvier 2010. [Citation : 9 septembre 2012.] http://wattmil.dyndns.org/vmware/19-gestionmemoiresousesx4?start=2.
81. **VMware.** Understanding Memory Resource Management in VMware ESX 4.1. *www.vmware.com.* [En ligne] 2010. [Citation : 9 septembre 2012.] http://www.vmware.com/files/pdf/techpaper/vsp_41_perf_memory_mgmt.pdf.
82. **Shieds, Gred et Siebert, Eric.** Hyper-V dynamic memory allocation vs. VMware memory overcommit. *searchservervirtualization.techtarget.com.* [En ligne] mai 2011. [Citation : 9 septembre 2012.] http://searchservervirtualization.techtarget.com/tip/Hyper-V-dynamic-memory-allocation-vs-VMware-memory-overcommit.
83. **Wikipédia.** Fonction de hachage. *fr.wikipedia.org.* [En ligne] 4 septembre 2012. [Citation : 9 septembre 2012.] http://fr.wikipedia.org/wiki/Fonction_de_hachage.
84. —. Appel système. *fr.wikipedia.org.* [En ligne] 30 mai 2012. [Citation : 12 septembre 2012.] http://fr.wikipedia.org/wiki/Appel_syst%C3%A8me.
85. **DataCoreTM.** SANsymphonyTM-V Hyperviseur de stockage, Fonctionnalités - Récapitulatif. *www.datacore.com.* [En ligne] 2012. [Citation : 12 septembre 2012.] http://www.datacore.com/Libraries/Language_-_French/SANsymphony-V_Abbreviated_Feature_Description_DataSheet_-_Fr.sflb.ashx.
86. **Wikipedia.** NPIV. *en.wikipedia.org.* [En ligne] 14 septembre 2012. [Citation : 14 septembre 2012.] http://en.wikipedia.org/wiki/NPIV.
87. **GuVirt.** Le NPIV. *www.guvirt.org.* [En ligne] [Citation : 14 septembre 2012.] http://www.guvirt.org/serveurs/2-esx/78-npiv-la-virtualisation-des-cartes-hba.

88. **CoArSys Conseils, Architecture et Systèmes.** Une avancée de plus dans le monde de la virtualisation. *www.coarsys.fr.* [En ligne] 2009 avril 2009. [Citation : 14 septembre 2012.] http://www.coarsys.fr/index.php?option=com_content&view=article&id=105:une-avancee-de-plus-dans-le-monde-de-la-virtualisation&catid=57:virtualisation&Itemid=79.
89. **Loïc Thobois, Direction filière technologies Microsoft.** Présentation des Cluster Shared Volume pour Hyper-V 2.0. *www.espace-microsoft.com.* [En ligne] 20 mai 2009. [Citation : 15 septembre 2012.] http://www.espace-microsoft.com/fr/articles/18914-presentation-cluster-shared-volume-hyper-v-2-0.html.
90. **Seng, Lai Yoong.** Virtual Machine Heartbeat in A Cluster Hyper-V Host. *www.ms4u.info.* [En ligne] 15 juillet 2011. [Citation : 15 septembre 2012.] http://www.ms4u.info/2011/07/virtual-machine-heartbeat-in-cluster.html.
91. **Josephes, Chris.** Understanding Snapshot Consumption. *broadcast.oreilly.com.* [En ligne] 5 février 2009. [Citation : 17 septembre 2012.] http://broadcast.oreilly.com/2009/02/understanding-snapshot-consump.html.