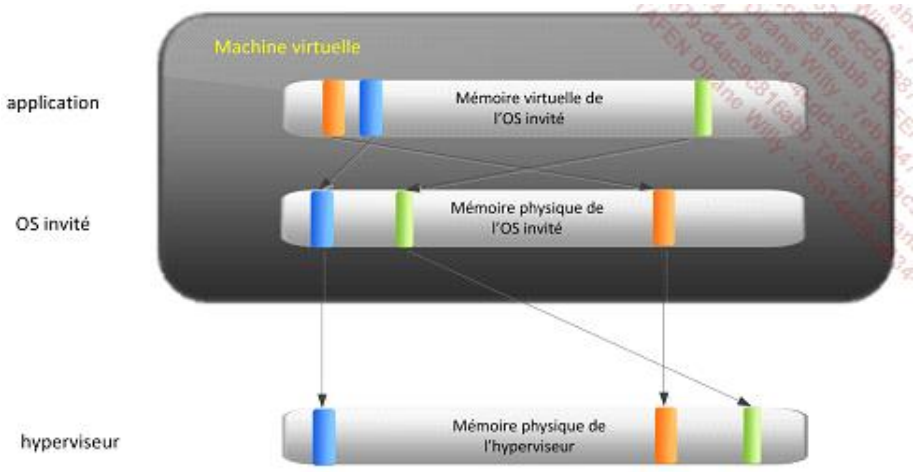
Mémoire

La dotation en mémoire a explosé ces dernières années, et ceci peut être attribué en majorité à la virtualisation. Il n'est pas rare aujourd'hui de voir des serveurs contenant plus de 400 Go de mémoire vive. Évidemment, ces serveurs sont destinés à supporter la couche logicielle ESXi (ou tout autre hyperviseur, en fait). La mémoire est souvent considérée comme le deuxième goulet d'étranglement ou facteur limitant après le stockage.

Lorsqu'on crée des machines virtuelles et qu'on observe au niveau des interfaces d'administration, il convient de distinguer :

- La *Host physical memory* : c'est la mémoire visible par l'hôte physique.
- La *Guest physical memory* : c'est la mémoire que voit l'OS invité dans la machine virtuelle.
- La *Guest virtual memory* : c'est un espace mémoire contigu présenté par l'OS invité aux applications.



1. Allocation

L'allocation de mémoire au niveau d'un système invité paraît simple : on configure la machine virtuelle avec la quantité de mémoire qu'on lui attribue. Par contre, la notion un peu moins simple est la suivante : il n'y a pas de lien direct entre la quantité de mémoire qu'on peut attribuer à une machine virtuelle et la quantité de mémoire physiquement présente au niveau de l'ESXi.

En d'autres termes, on peut attribuer autant de mémoire qu'on veut à une machine virtuelle, même si l'on dépasse la quantité de mémoire totale d'un ESXi.

2. Surallocation

Attribuer plus de mémoire que ce qui est réellement disponible est appelé surallocation ou « overcommitment ». Ce n'est pas un problème, à partir du moment où ce phénomène est contrôlé.

Prenons l'exemple d'un hyperviseur doté de 128 Go de mémoire vive. Dans l'inventaire de ce serveur, il y a 20 machines virtuelles. Ces 20 machines virtuelles sont configurées avec 12 Go de mémoire vive.

Après un rapide calcul, on se rend compte qu'on a attribué 12 x 20 = 240 Go de RAM. Est-ce un problème ? C'est possible.

Si seulement un quart des machines virtuelles est régulièrement en fonctionnement, cela fait 12 x 5 = 60 Go de

mémoire. Aucun souci vu que le serveur a 128 Go de RAM.

Il convient de prendre en compte le fait suivant : la précédente affirmation est valide si on considère que chaque machine virtuelle consomme systématiquement 100 % de la mémoire vive attribuée. Il est donc encore plus tentant de faire de la surallocation quand les machines virtuelles n'utilisent pas la totalité de la mémoire, ce qui est assez souvent le cas.

La surallocation n'est donc absolument pas nocive pour un environnement virtualisé tant qu'elle est **surveillée**.

L'hyperviseur dispose cependant de mécanismes d'économie (optimisation) de la mémoire vive.

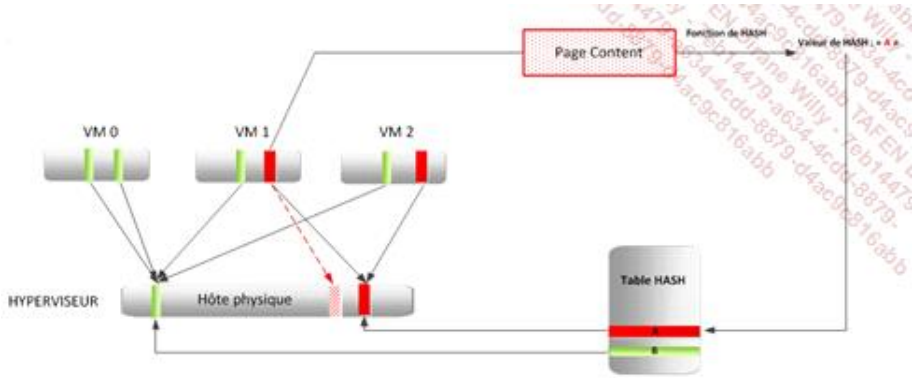
3. Transparent Page Sharing intra et inter VM

Le TPS, ou *Transparent Page Sharing* est le plus cité, et paradoxalement le moins bien connu. C'est le premier processus d'optimisation de la mémoire déclenché par l'hyperviseur.

Le principe est simple : il y a plusieurs machines virtuelles en fonctionnement hébergées sur un hyperviseur. Pour toutes ces machines virtuelles, L'ESXi parcourt les pages mémoires chargées et utilisées. Les pages identiques sont mutualisées et un pointeur est créé afin qu'une seule copie soit chargée et utilisée par plusieurs machines virtuelles ou plusieurs processus dans une machine virtuelle.

Il n'y a aucune incidence au niveau des systèmes invités car à chaque modification d'une page mémoire en particulier, le serveur hôte crée une copie spéciale (copy on write).

Il peut y avoir des pages identiques pour une même machine virtuelle, ou des pages identiques retrouvées au niveau de plusieurs machines virtuelles. On parle donc de TPS intra-VM ou TPS inter-VM :



Plus il y a de machines virtuelles du même type (système d'exploitation et applications) fonctionnant sur un hyperviseur, plus le Transparent Page Sharing se révèle efficace. C'est particulièrement approprié pour les machines de type VDI (*Virtual Desktop Infrastructure* - virtualisation de postes de travail).

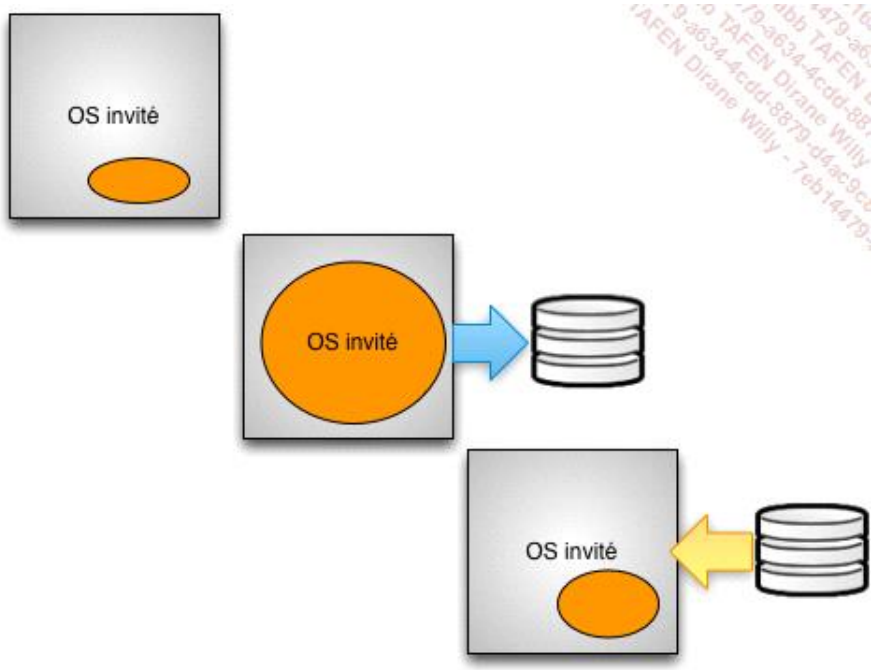
Il y a d'ailleurs une différence de traitement entre le TPS intra-VM et le TPS inter-VM : depuis début 2015, VMware a fourni des patches ayant pour effet de désactiver le TPS inter-VM par défaut. Ceci est lié aux résultats d'une étude ayant prouvé que l'étanchéité entre machines virtuelles pouvait être mise en défaut en passant par les fonctions de TPS.

Le fait de désactiver TPS inter-VM est un principe de précaution de la part de l'éditeur. Il n'est pas prouvé que le résultat de l'étude puisse être reproduit en environnement de production.

4. Ballooning

Le pilote de contrôle de la mémoire des machines virtuelles (vmmemctl) est très connu des administrateurs VMware. Il permet d'évaluer les manques de mémoire vive au niveau des configurations des machines virtuelles. En effet, quand une machine virtuelle doit utiliser une quantité de mémoire vive alors qu'elle en manque, le serveur ESXi utilise le vmmemctl sur les autres VM pour libérer de la mémoire. Le vmmemctl est installé dans le système invité via les VMware Tools. Il n'est pas obligatoirement installé et peut être désinstallé a posteriori dans le cas de restrictions au niveau de l'application invitée. Il force le système invité à utiliser ses mécanismes de swap interne en se comportant comme une application demandant de la mémoire vive physique pour démarrer.

Le comportement de ce pilote peut être assimilé à un ballon qui gonfle (la demande de mémoire va en grossissant). C'est de là que provient le nom couramment utilisé de balloon driver :



Le balloon driver est configuré par défaut pour libérer jusqu'à 65 % de la mémoire attribuée à une machine virtuelle.

Le balloon driver fonctionne en réclamant la mémoire non active, cette mémoire est redistribuée aux systèmes manquant de mémoire vive.

5. Compression mémoire

La compression mémoire a été introduite depuis vSphere 4. C'est l'avant-dernière technique que le serveur hôte utilise avant un impact visible sur le fonctionnement des machines virtuelles. Il s'agit de compresser certaines pages mémoires. Quand la machine virtuelle doit utiliser une page compressée, le serveur hôte décompresse la page pour la fournir à la VM.

Chaque VM a son propre cache de mémoire compressée. Le serveur ESXi compresse des pages de 2 Ko. Chaque décompression génère de l'overhead, mais bien moins que l'utilisation de disque.

6. VMkernel swap

La technique de VMkernel swap existe sous deux formes :

- Utilisation d'un SSD
- Utilisation des datastores

Le VMkernel swap est lié à la notion de surallocation. Afin que les machines virtuelles puissent fonctionner en cas de manque de mémoire physique, un fichier d'échange est créé (par défaut) sur le datastore contenant le fichier de configuration de la machine virtuelle. Ce fichier est créé au démarrage de la machine virtuelle et supprimé après l'arrêt de celle-ci.

Par défaut, la taille du fichier de swap de la machine virtuelle est équivalente à la quantité de mémoire vive configurée pour le système d'exploitation invité.

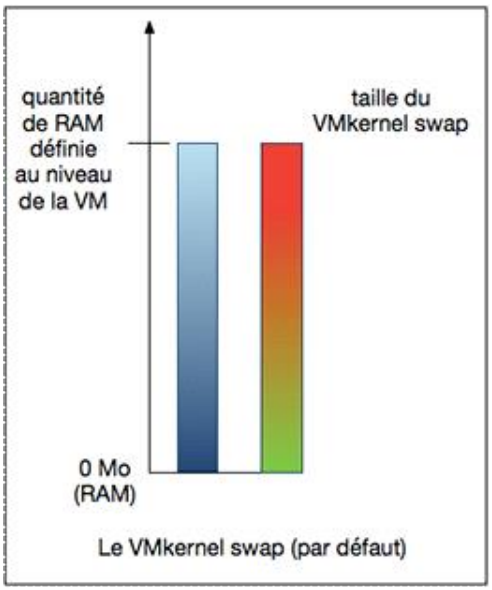
Il est possible de configurer un espace de stockage spécifique pour les fichiers d'échange.

Voici un exemple pour illustrer ces propos :

Prenons une machine virtuelle quelconque. Nous nous concentrerons sur la quantité de mémoire vive. Disons que l'administrateur a attribué 8 Go de RAM à cette VM. Par défaut, il n'y a aucune réservation.

Au démarrage de cette VM, un fichier d'échange (vswp) va être créé : il pèsera 8 Go. Le VMkernel utilisera ce fichier si la VM doit fonctionner alors qu'il a peu ou plus de mémoire physique, ainsi le système conserve sa mémoire, ce qui évite un « plantage » en règle du système invité. Cependant, les conséquences sont désastreuses : au lieu d'utiliser la RAM, on utilise du disque dont les temps d'accès et la bande passante ne sont nullement comparables ! La machine virtuelle utilisant le VMkernel swap est bien plus lente et la différence est aisément ressentie (et en général en tant qu'administrateur, vous avez des nouvelles des utilisateurs des applications concernées par les VM qui swappent).

Les disques durs ont des temps d'accès de l'ordre de la milliseconde alors que la mémoire vive permet des temps d'accès de l'ordre de la nanoseconde.



Le VMkernel swap (par défaut)

La création de fichiers de swap ne doit pas être évitée à tout prix. Il faut prévoir l'espace disque nécessaire. Un message « la machine virtuelle ne peut pas démarrer car l'espace disque est insuffisant » signifie qu'il n'y a pas assez de place pour créer le fichier d'échange.

Si l'hyperviseur dispose d'un SSD, on peut configurer celui-ci pour stocker tous les fichiers d'échange des machines

virtuelles.