

Processeurs

Dès qu'on parle de matériel informatique - clients (pc) ou serveurs -, la description commence dans 95 % des cas par le ou les processeurs. C'est logique car il s'agit du cœur... enfin des unités de calcul.

Tout comme il y a des processeurs (sockets) physiques pour les machines - physiques - les machines virtuelles comportent aussi des processeurs - virtuels.

➤ On considère le socket comme le processeur dans toutes les descriptions de matériel que nous rencontrons. Ceci s'explique finalement assez simplement : au niveau matériel, le socket matériel représente la prise de la carte mère sur laquelle on enclenche le microprocesseur. Donc pour clarifier les choses on parle de socket au lieu de processeur, ce qui évite les confusions avec les cœurs.

1. La notion de vCPU et HEC

Qu'est-ce qu'un vCPU ?

Un vCPU représente la possibilité pour une machine virtuelle de gérer des files d'instructions (des calculs) que l'on appelle « thread » ou « fil ».

➤ En parlant de vCPU, on part en fait du CPU qui signifie Central Processing Unit, ou unité centrale de traitement. C'est ce composant qui exécute les instructions envoyées par des programmes ou les systèmes d'exploitation.

C'est à peu près la définition que l'on peut voir partout, cependant je trouve qu'il y manque un petit quelque chose, une précision qui fait que le doute n'est plus permis. L'idée sous-jacente est qu'un vCPU n'est qu'un CPU (virtuel, certes) et que donc comme la plupart des CPU sont multicœurs, les vCPU peuvent aussi comprendre plusieurs cœurs. C'est d'ailleurs (officiellement) prévu depuis vSphere 5, qui permet via l'interface graphique de configurer le nombre de « vCores » d'un vCPU.

Ainsi donc, quand VMware parle de vCPU, il est supposé en fait : vCPU simple cœur (single core). Afin d'éviter toute confusion, on utilisera la notion de contexte d'exécution matériel (HEC pour *Hardware Execution Context*), qui correspond à la possibilité de gérer **un** thread.

Ainsi, une machine virtuelle disposant de 8 HEC pourra être indifféremment configurée comme suit :

1 vCPU - 8 cœurs

2 vCPU - 4 cœurs chacun

4 vCPU - 2 cœurs chacun

En considérant le même nombre de HEC, quel est donc l'intérêt d'une configuration plutôt qu'une autre ?

La notion de licences applicatives est un élément de réponse évidemment : certaines licences logicielles dépendent du nombre de processeurs (sockets) et il y a donc un bénéfice à multiplier le nombre de cœurs dans une machine virtuelle.

D'ailleurs, ce cas n'est pas réservé aux machines virtuelles : les serveurs ESXi sont licenciés au processeur, sans limite au nombre de cœurs. Il sera donc certainement courant d'ici 2016 de trouver des hyperviseurs installés sur des machines physiques monoprocesseurs, ceci étant lié aux annonces des deux constructeurs majeurs de processeurs d'architecture X86_64, Intel et AMD.



Les fabricants de processeurs ont annoncé des modèles dotés de 20 cœurs physiques. Cette multiplication des cœurs n'a pas que des avantages hors hyperviseurs : Microsoft a annoncé que pour Windows Server 2016, le mode de licence serait au cœur de processeur. La licence au cœur représente un huitième de la licence au processeur... Ce qui implique qu'en utilisant des processeurs dotés de 8 cœurs ou moins, la facture ne s'alourdira pas par rapport à Windows 2012. Cela posera de nouvelles questions de design quant au « scale-in » (peu de serveurs très puissants) ou au « scale-out » (un nombre plus élevé de serveurs, de puissance plus modeste). Pour plus d'informations, consultez les sites suivants : <http://www.zdnet.fr/actualites/windows-server-2016-des-licences-par-coeur-et-non-plus-par-processeur-39829284.htm>, <http://www.silicon.fr/windows-server-2016-microsoft-licensing-coeur-processeur-133222.html>

En tout cas, lors de la configuration des machines virtuelles, il faut garder à l'esprit qu'il ne sera pas possible d'attribuer plus de HEC que ne permet la machine physique.

Par exemple, pour un serveur bihexacore (2 processeurs de 6 cœurs chacun, donc 12 cœurs au total - hors hyperthreading), il ne sera pas possible de dépasser 12 HEC par machine virtuelle. Par contre, plusieurs machines virtuelles pourront utiliser 12 HEC (ou moins).

Comment expliquer cela ?

Ce n'est en fait pas très compliqué : un cœur de processeur est capable d'exécuter un nombre d'instructions en une seconde. Ce nombre est prédéfini (il peut varier, nous verrons cela plus tard). C'est la fréquence, exprimée en (Giga) Hertz. À partir du moment où ce processeur est à disposition du système ESXi, c'est ce dernier qui gère les « connexions » machine virtuelle - processeur physique (cœur). Ces connexions pouvant être modifiées très rapidement (de 2 à 40 millisecondes), les processeurs physiques sont utilisés au mieux et par plusieurs machines virtuelles simultanément. Chaque machine virtuelle consomme en fait ce qu'on appelle du « temps CPU ».

Dans l'exemple du serveur bihexacore, il est donc possible de configurer plusieurs machines virtuelles avec 12 HEC.

2. Le multicœur et l'hyperthreading

a. Les différences

L'hyperthreading consiste à faire voir à un OS deux processeurs physiques alors qu'il n'y en a qu'un seul en réalité.

Les avantages de l'hyperthreading sont les suivants :

- Amélioration du support de code multithreadé (le code doit supporter le multithread).
- Gestion de plusieurs threads en même temps.
- Potentielle augmentation de nombre d'utilisateurs.

Le multicœur consiste à faire voir à un OS plusieurs processeurs (cœurs physiques) qui font partie (ou non) du même processeur physique.

Les avantages du multicœur sont les mêmes que pour l'hyperthreading.

Par contre, il faut distinguer les inconvénients des deux.

Dans le cas de l'hyperthreading, le processeur fait croire à l'OS qu'il y a deux processeurs en créant deux processeurs logiques, cependant, autant les deux processeurs logiques ont des registres séparés, autant le bus et le cache système sont bel et bien partagés.

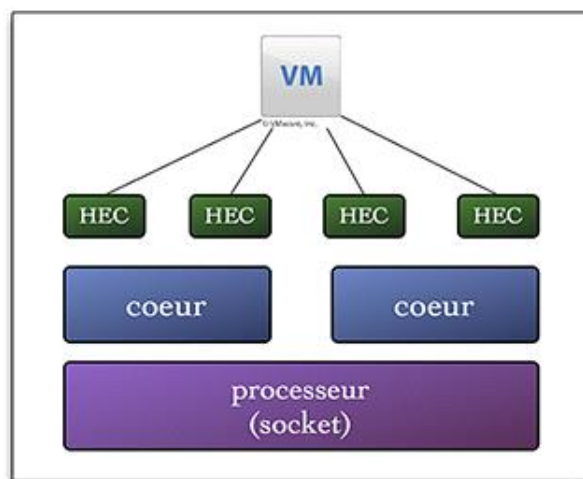
En général, les performances d'une application en multithreading sont meilleures en multicœur qu'en

hyperthreading. Dans le cas où une application gère l'hyperthreading, cela peut devenir très intéressant et le gain de performance peut aller jusqu'à 30 %. Dans ce cas et pour simplifier, l'application va envoyer plusieurs instructions simultanément et en attendre le résultat mais l'astuce réside dans le fait que les instructions sont de nature différente et peuvent être calculées par le même cœur quasiment simultanément.

b. La gestion de l'ESXi

Le serveur ESXi gère l'hyperthreading et il est conseillé de l'activer dans le BIOS ou EFI du serveur. Si l'on ne souhaite pas utiliser l'hyperthreading, il est possible de le désactiver au niveau du serveur ESXi. Sachant que l'hyperviseur prend en compte l'hyperthreading, le désactiver est réservé pour des situations particulières. Ce qu'il faut savoir c'est que pour une machine virtuelle, si celle-ci a plus d'un HEC, ces HEC ne seront pas liés au même cœur physique.

Voici un exemple de machine virtuelle contenant 4 vCPU simple cœur sur un processeur physique bicœur avec hyperthreading activé :



Le HEC utilisé pour un vCPU peut changer régulièrement car c'est le VMkernel qui gère la répartition et l'équilibrage de charge. Le changement de HEC pour un vCPU peut avoir lieu aussi souvent que toutes les 20 millisecondes. Pour toute exécution d'une instruction, l'hyperviseur devra disposer d'un HEC libre. Dans le schéma précédent, la machine virtuelle, qu'elle soit configurée avec un vCPU de quatre cœurs, quatre vCPU simple cœurs ou deux vCPU bicœurs, dispose de 4 HEC. L'hyperviseur « attendra » d'avoir 4 HEC à disposition avant d'exécuter une instruction CPU pour cette VM.

Le fait de configurer la plupart des machines virtuelles avec plusieurs vcores ou vCPU se révèle être une source potentielle de ralentissements et latences au niveau de l'infrastructure virtuelle. Il est donc conseillé d'attribuer un vCPU simple cœur par défaut à chaque machine virtuelle sauf indications ou prérequis contraires (généralement cela concerne l'application qui fonctionnera sur le système invité).

3. L'affinité CPU

L'affinité au processeur permet de bloquer une machine virtuelle sur un ou plusieurs cœurs de processeurs définis. L'affinité se fait en indiquant sur quel(s) cœur(s) la VM fonctionne dans le menu modifier le matériel virtuel, partie CPU :

Planification affinité	État de l'hyperthreading : Inactif
	CPU disponibles : 2 (CPU physiques)
	Choisir l'affinité du processeur physique pour cette machine virtuelle.
	Utiliser « - » pour les plages et « , » pour séparer les valeurs. Par exemple, « 0, 2, 4-7 » désigne les processeurs 0, 2, 4, 5, 6 et 7.
	Effacer la chaîne pour supprimer les paramètres d'affinité.
	<input type="text"/>

C'est un bon moyen de garantir une quantité de ressources, mais à un prix élevé : l'affinité CPU désactive les fonctionnalités suivantes : vMotion, DRS, HA et storage vMotion.

Attention : si la machine virtuelle fait partie d'un cluster DRS, le réglage ci-dessus n'apparaît pas.

L'explication est la suivante : une machine virtuelle peut être assignée au HEC 14 par exemple. Un autre serveur du cluster peut comporter moins de HEC (et donc moins de cœurs ou moins de processeurs), ainsi la situation de la VM ne pourrait être reproduite sur un autre serveur hôte dans le cas d'une éventuelle migration.

De plus, une autre limitation existe : une machine virtuelle a des processus d'affichage et de gestion des périphériques (clavier, souris et lecteur CD-ROM virtuel par exemple). Ces processus sont gérés par les cœurs désignés par le paramètre d'affinité. Cela signifie qu'en cas d'utilisation intense de l'affichage et des périphériques d'entrées et pointeur, les cœurs désignés pourraient être surchargés car ils gèrent aussi les appels du système invité. Il est donc conseillé de systématiquement attribuer un cœur supplémentaire à la VM : pour une VM mono-CPU logique (monocœur) il faut attribuer 2 cœurs, de même pour une VM bi-CPU le réglage d'affinité nécessite 3 cœurs.

➤ Il est fortement déconseillé d'utiliser l'affinité CPU. D'ailleurs, vous ne verrez cette option dans [presque] aucune architecture ou design de référence ! C'est un cas très particulier qui est d'ailleurs le plus souvent utilisé pendant les formations. Le VMkernel gère les changements de HEC au niveau des machines virtuelles : des changements dans la correspondance vCPU-cœur physique se font toutes les 4 à 20 millisecondes, sans perturbation perçue au niveau des machines virtuelles.

4. Les accès mémoire

Les éléments CPU et mémoire sont interdépendants. Si un processus de machine virtuelle (un « world ») utilise des cœurs de processeur physique mais subit des latences au niveau de l'accès à la mémoire, ceci se ressent directement au niveau du système invité, ou pire des applications et donc des utilisateurs.

Les processeurs d'une même carte mère utilisaient précédemment un seul et même contrôleur mémoire. Les accès étaient donc toujours à la même vitesse de bus interne qui devenait le facteur limitant. Le contrôleur mémoire était intégré à ce qu'on appelle le « northbridge ».

➤ Le northbridge est l'un des deux jeux de puces présents sur une carte mère ; il a comme tâche principale la gestion de la mémoire vive et des ports PCI-express et anciennement AGP !

Au fil du temps, les fabricants de processeurs ont délaissé cette architecture dite d'accès uniforme à la mémoire ou UMA (*Uniform Memory Access*), pour des solutions plus rapides (dans certains cas).

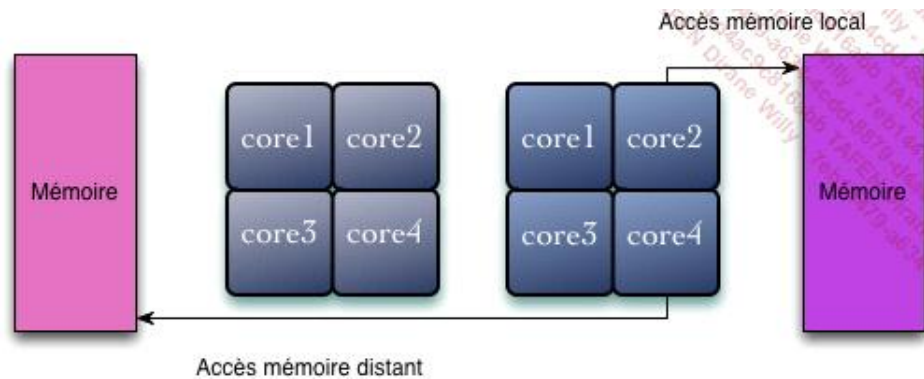
a. NUMA

Les contrôleurs mémoire étant désormais intégrés au processeur, l'accès à cette mémoire est beaucoup plus rapide que si l'on devait passer par un bus commun.

C'est pour cela que la gestion du NUMA ou *Non Uniform Memory Access* est importante : l'hyperviseur tente de maintenir les worlds dans un ensemble processeur + mémoire adressée, qu'on nomme un « nœud NUMA ».

On distingue d'autre part l'accès distant de la mémoire par un processeur qui ne fait pas partie du même nœud.

Voici un schéma résumant la situation :



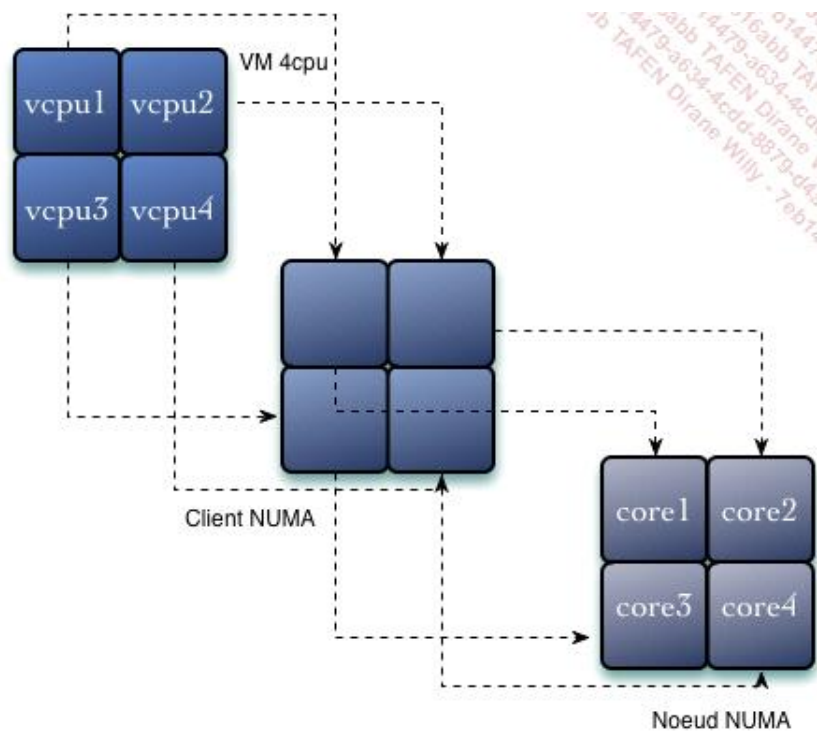
L'accès distant de la mémoire par un processeur est moins rapide car il doit traverser une interconnexion.

Il existe deux prérequis afin de pouvoir utiliser NUMA :

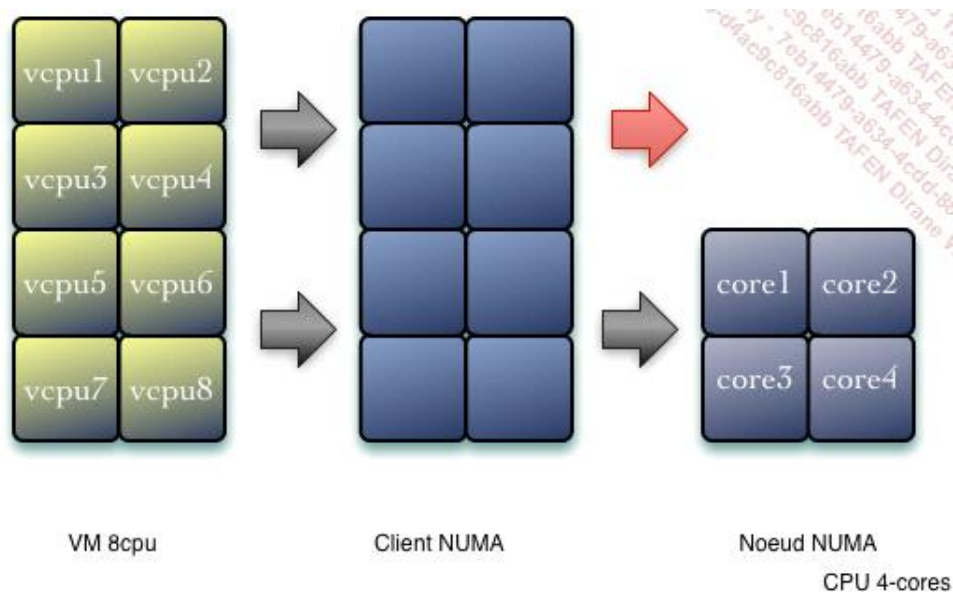
- Posséder un système le supportant (client).
- Avoir des processeurs supportant NUMA.

Or, c'est le cas de l'ESXi et des processeurs INTEL et AMD, mais seulement depuis l'architecture Intel Nehalem pour Intel par exemple...

Imaginons que nous avons tous les prérequis, et que nous souhaitons créer une VM avec 4 HEC sur une machine physique biprocesseur quadricœur. Le nombre de HEC est égal au nombre de cœurs. Il est donc possible au processeur d'avoir uniquement un accès local.



Imaginons désormais que nous souhaitons créer une VM 8 avec HEC sur le même hôte physique. Nous aurons alors un problème.

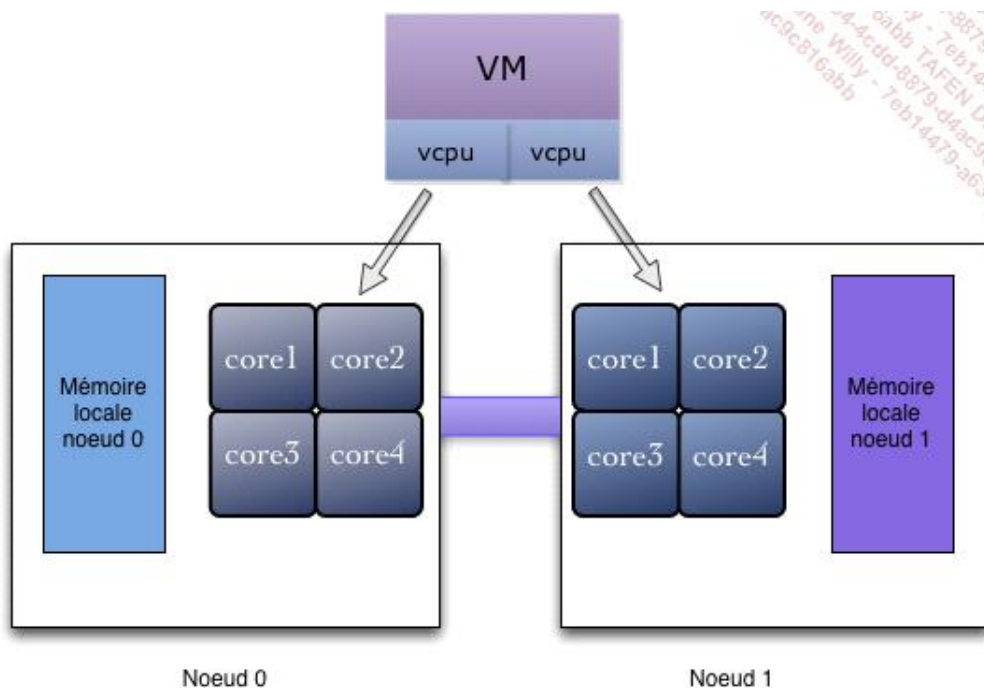


Dans ce cas précis, l'hyperviseur va utiliser l'interconnexion entre les processeurs afin d'utiliser de la mémoire distante.

b. vNUMA

Dans le cas où la configuration de la machine virtuelle excède les caractéristiques d'un nœud NUMA, l'hyperviseur utilise une fonction de NUMA virtuel - virtual NUMA (ou vNUMA).

Cette fonction vNUMA répartit les ressources CPU et RAM au niveau de plusieurs nœuds NUMA physiques. Il s'agit d'utiliser plusieurs nœuds NUMA physiques pour en faire un nœud virtuel.



L'importance de ce fonctionnement est liée aux « monster VMs », ces énormes machines virtuelles avec des configurations matérielles dignes de serveurs physiques. Le fait de garder une machine virtuelle dans un nœud NUMA améliore les performances générales. Dans le cas d'un vNUMA, l'hyperviseur tente d'équilibrer l'utilisation de deux nœuds, mais cette répartition n'est effectuée qu'au démarrage de la machine virtuelle. En cas d'ajout de mémoire à la machine virtuelle, sous vSphere 5 l'hyperviseur ne chargeait que le premier nœud NUMA, ce qui en déséquilibrait la gestion.

Depuis vSphere 6, en cas d'ajout de mémoire à une machine virtuelle, chaque nœud NUMA est utilisé à parts égales.

5. La gestion des licences

Les composants exigeant une licence sont le serveur vCenter et les hyperviseurs ESXi. La licence se présente sous forme d'une suite de 25 caractères à entrer dans l'interface web. Il y en a au moins deux à entrer dans l'interface : une pour vCenter - vCenter Foundation qui peut gérer jusqu'à 3 hyperviseurs et vCenter Server qui n'a pas de limitation (sauf la limitation technique de 1 000 hyperviseurs au maximum dans l'inventaire de vCenter).

En ce qui concerne la licence des hyperviseurs, elle est présentée comme licence « vSphere » et correspond à un nombre de processeurs physiques. Une seule licence peut être utilisée et attribuée à plusieurs serveurs ESXi. Seul compte le nombre de processeurs.

Certaines fonctions comme vSAN utilisent des licences particulières.

Les licences sont disponibles sous forme de packs de licences, et ceux-ci proposent plus ou moins de fonctions. On notera à titre d'exemple les packs « Essential plus » qui comprennent une licence vCenter Foundation, et une licence vSphere pour 6 processeurs. Les fonctions principales sont : cluster haute disponibilité, vMotion (migration de machines virtuelles à chaud), data protection (outil intégré de sauvegarde sur disque et restauration des machines virtuelles).

Les packs Enterprise et Enterprise plus offrent plus de fonctions, le pack Enterprise plus représente la licence « haut de gamme » de vSphere, incluant les commutateurs Ethernet virtuels (vSwitches) de type distribués capables de gérer les VLAN privés (Private VLAN), l'agrégation de lien et le traffic shaping (en entrée et en sortie). C'est à partir de l'édition Enterprise que les clusters d'équilibrage de charge sont inclus.

Voici le récapitulatif des différentes offres de licences vSphere ainsi que leurs fonctionnalités associées : <https://cloud.berton.me/index.php/s/mb23MpgwmE2hZ3i>

Pour plus d'informations sur les tarifs, consultez cette page : <http://www.vmware.com/fr/products/vsphere/pricing>

Concernant les types de licences (au processeur, à la machine virtuelle...), consultez le site suivant (cela concerne tous les produits VMware, vSphere compris) : <http://www.vmware.com/fr/support/support-resources/licensing/product-licenses.html>

Pour plus d'informations, n'hésitez pas à consulter cette page (c'est un document PDF à télécharger) : <http://www.vmware.com/files/fr/pdf/products/vsphere/VMware-vSphere-Pricing-Whitepaper.pdf>