VMware – Les mécanismes de gestion du CPU

12 juin 2019 (https://www.virtualease.fr/vmware-mecanismes-de-gestion-cpu/) Mathieu (https://www.virtualease.fr/author/matteo/) ESXi (https://www.virtualease.fr/category/vmware/esxi/), VMware (https://www.virtualease.fr/category/vmware/), vSphere (https://www.virtualease.fr/category/vmware/vsphere/)

Synopsis

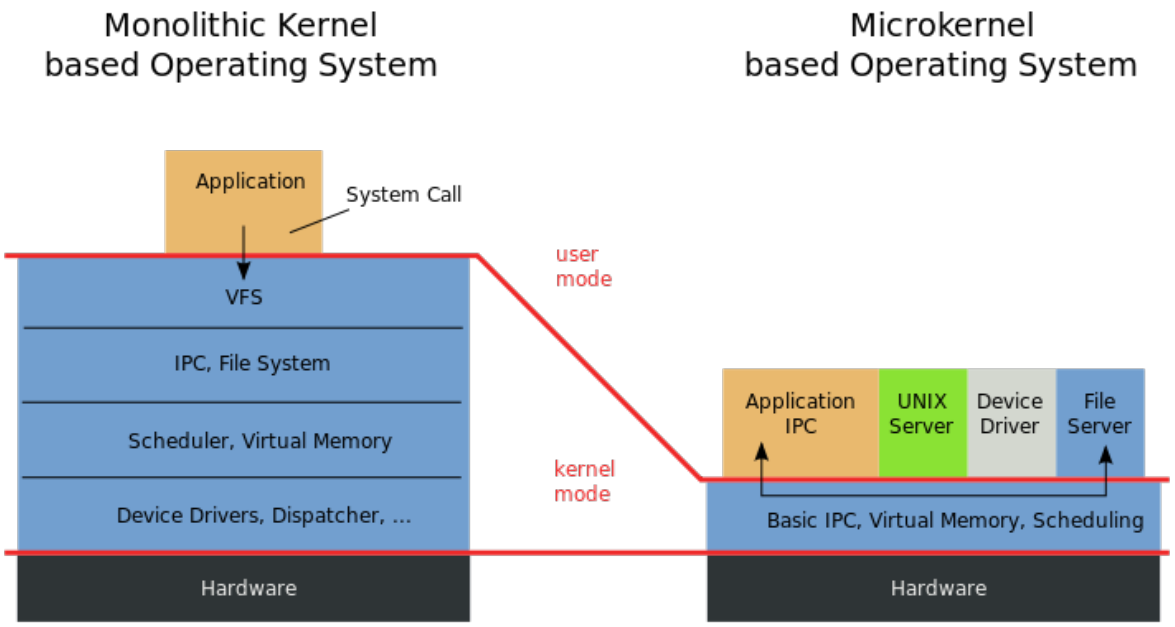
VMware utilise des mécanismes de gestion intelligents du **CPU**, de la **Mémoire**, du **Storage** et du **Réseau** pour permettre la cohabitation d'un grand nombre de VMs et l'**overcommitting**. Après avoir parlé de la **RAM** dans l'article *VMware – Les mécanismes de gestion de la RAM (/vmware-mecanismes-de-gestion-de-ram/)*, je vais me pencher aujourd'hui sur la gestion du **CPU** avec des explications claires sur l'utilisation et le fonctionnement des mécanismes.

Je vous conseille de jeter un oeil au post *VMware – ESXTOP (https://www.virtualease.fr/vmware-esxtop/)*

9

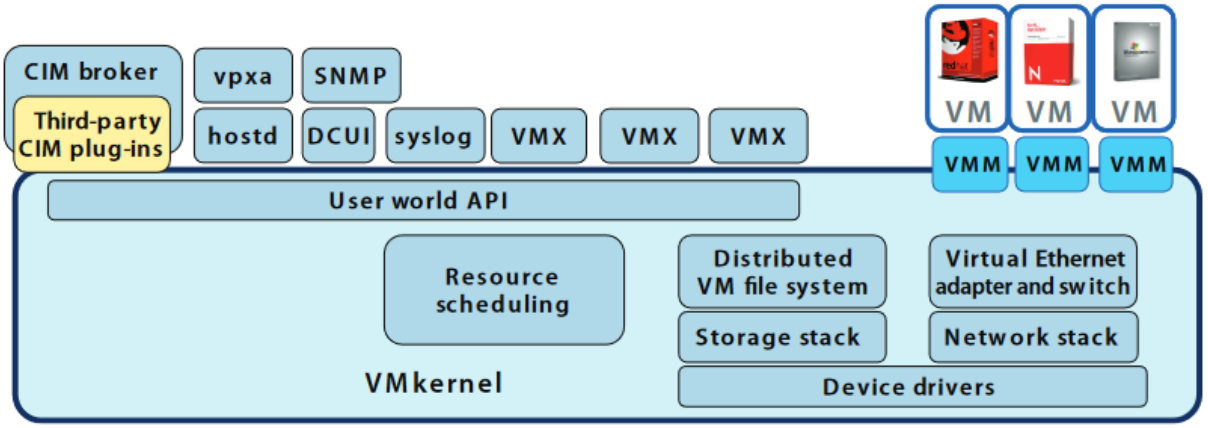
VMkernel

Quand l'hyperviseur de **VMware** s'appelait encore **ESX** (3.5) il était basé sur un kernel Linux, avec l'évolution vers **ESXi** ils l'ont remplacé par un microkernel. Il permet d'être plus léger et plus sécurisé avec moins de code à patcher en ne gardant que l'essentiel dans la couche basse.



Wikipedia.org

L'architecture **VMware ESXi** comprend le système d'exploitation appelé **VMkernel**, ainsi que des processus qui s'exécutent au-dessus. **VMkernel** fournit des moyens pour exécuter tous les processus sur le système, y compris les applications, les agents de gestion mais aussi les machines virtuelles. C'est le cœur de l'ESXi, il contrôle tous les périphériques matériels du serveur et gère les ressources des applications.



World

Un **world** est à considérer comme un process ou un thread, c'est un terminologie du **VMkernel**.

~~**VMkernel** utilise des **world** de gestion, séparés en 3 groupes/interfaces.~~

- ~~Guest System : Il est nécessaires pour effectuer divers services qui comprennent :~~
  - ~~Idle World : Un par **CPU physique**, qui fonctionne quand il n'y a rien d'autre à exécuter sur ce CPU.~~
  - ~~Helper Worlds : Pour effectuer des **taches asynchrones**.~~
  - ~~Driver Worlds : **Gestion** souris et clavier, snapshots et périphériques I/O.~~
- ~~Service Console : Utilisé pour l'exécution du **service console**, qui était utile dans les anciennes version pour le support VMware, à l'heure actuelle il est remplacé par le DCUI/Shell sur ESXi.~~
- ~~Virtual Machine (Hardware) : Prend en charge la gestion des instructions hardware **CPU**~~

~~et **RAM.**~~

Depuis vSphere 6.5 il y a une séparation en 4 Worlds :

- vCPU
- VM Mouse Keyboard and Screen (MKS)
- CD-ROM
- VMX File

pCPU

Un **pCPU** désigne soit un **CPU Logique** soit un **Cœur Physique** suivant le contexte.

- Cœur Physique dans le cas ou l'Hyperthreading est indisponible ou désactivé.
- CPU Logique si l'Hyperthreading est activé.

On peut ainsi assimiler pCPU au nombre de Cores Logiques (Logical Processors / lCPU).

Ex. ESXi – 2 Sockets – Xeon Quad Core :

- Sans Hyperthreading = 2 Sockets x 4 Cores = 8 lCPU = 8 pCPU
- Avec Hyperthreading = 2 Sockets x (2 x 4 Cores) = 16 pCPU

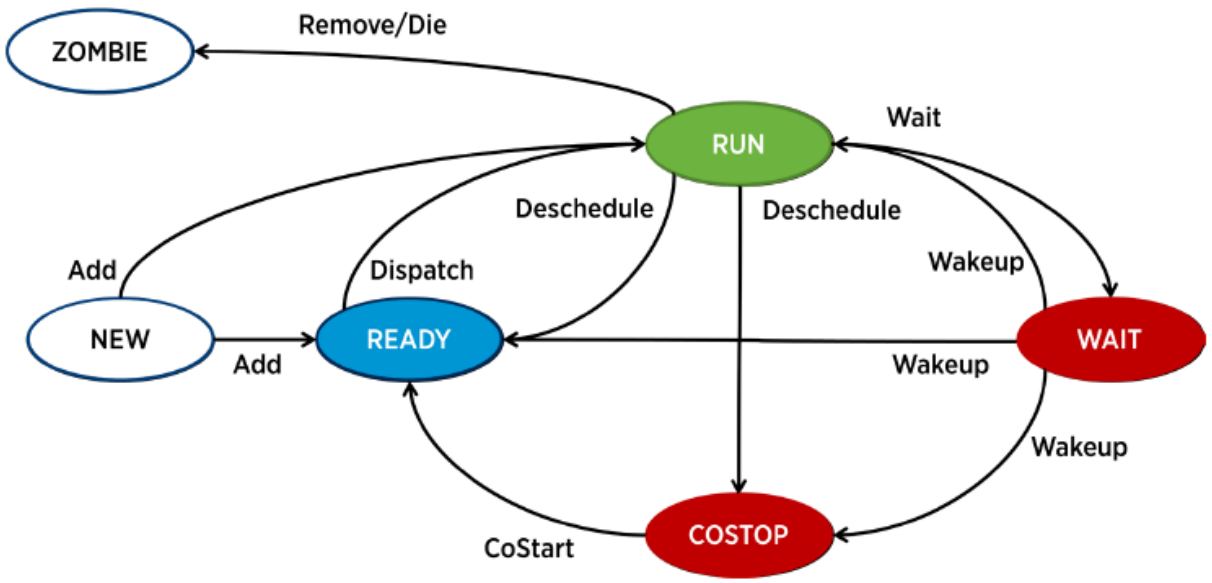
vCPU

vCPU désigne un core **CPU virtuel** utilisé par une Machine Virtuelle, il est assimilé à un **World**, c'est donc un processus qui tourne sur un **pCPU**.

Une VM de 4 vCPU utilise donc 4 vCPU Worlds.

CPU States

Les demandes CPU du **Guest OS** passent par ce processus, lui-même géré par le CPU **Scheduler**.



9

Un **World** est associé à un état d'exécution, au départ il est soit en **RUN** soit en **READY** State en fonction de la disponibilité d'un **pCPU**. Le temps passé par les différents états est disponible avec un **ESXTOP**.

En **READY** State il est envoyé pour exécution par le Scheduler et passe en **RUN** State. Il peut être plus tard de-programmé pour redevenir **READY** ou **COSTOP** (vSMP).

Un World en **RUN** passe en **WAIT** en réservant une ressource, il est réveillé une fois la ressource disponible.

Un World qui n'a rien à exécuter passe en **WAIT_IDLE**, il n'attend et ne bloque donc pas de ressource.

Overcommitting CPU

Tout d'abord qu'est-ce que l'**Overcommitting**, en faisant simple c'est le fait d'utiliser des ressources en excès, **assigner** plus de **vCPU** aux machines virtuelles que l'ESXi n'en **possède**, on dit alors qu'il y a **Contention**.

Comme vous aller le voir ci-dessous il y a des mécanismes de gestion dans VMware qui permettent cette **sur-allocation**, le tout est de ne pas en abuser sous peine de perdre en performance.

Ratio Overcommit

Il désigne le taux de **sur-allocation** de l'host, c'est mathématiquement la division du nombre de pCPU par le nombre de vCPU total des VMs.

Ex. Imaginons que j'ai un ESXi sans hyperthreading avec 2 Sockets de 10 Cores, dessus je fais tourner 10 VMs avec 4 vCPU chacune, j'aurais au total 20 pCPU (2*10) et 40 vCPU (10*4) donc un ratio de 2 (40/20).

J'utilise donc deux fois plus de cores que mon ESXi n'en possède. Ce ratio c'est à vous de le choisir suivant la **qualité de service** que vous souhaitez délivrer, personnellement je conseille un ratio de maximum 3 pour de la **prodution** et proche de 1 pour des services vraiment **critiques**.

Il faut bien séparer contention et % d'utilisation CPU, ils restent liés mais il est possible d'avoir un pourcentage non excessif avec un gros overcommit, il en resulte de mauvaises performances avec un CPU Ready qui s'envole.

CPU Scheduler

Le Scheduler est un système VMware vSphere de gestion du CPU.
Il va gérer les demandes CPU des machines virtuelles via leurs World et **ordonner** les exécutions pour maintenir les performances avec certaines règles de **partage** des ressources.

Le Scheduler utilise par défaut un **Quantum** de 50ms (modifiable via cpu.quantum), qui est le temps maximum de monopolisation d'un pCPU par un vCPU (dans le cas d'une priorité égale).

Comment le Scheduler arrive à savoir quel world doit il mettre en RUN et lesquels sont en Ready State ? Il détermine ceci via une priorisation ainsi qu'une forme de « justice » pour être équitable.

Lorsqu'il prend une décision d'équilibrage de charge, la charge CPU seule n'est pas un facteur suffisant pour une bonne performance. Par exemple, la migration d'un World sur un pCPU différent pour maximiser l'utilisation processeur peux induire une perte de performance significative pour l'application, un peu comme DRS qui ne souhaite pas équilibrer malgré la différence de charge, le gain ne surpasse pas la perte.

La notion de **réactivité** est importante, qu'il ne soit pas trop longtemps en attente en Ready

State, il y a une prise en compte de la **priorité**, un World souvent bloqué pendant contention sera à un moment **forcé** à s'exécuter.

CPU Scheduler Accounting

Un des élément fondamental du **Scheduler** est l'**Accounting Algorithm**, il permet de lui afficher une vue holistique des workloads présents sur l'ESXi.

Ainsi il permet de calculer le temps exact de **monopolisation** d'un CPU par un **world**, mais aussi de prendre en compte le temps « volé » par certains worlds du fait de l'**overcommitment** et d'être ensuite **équitable**, il est au final le **juge** du CPU Scheduler.

CPU Ready Time

Le CPU Ready Time contrairement à ce qu'on pourrait penser est une valeur qui doit rester relativement **faible**, elle n'est pas comme on pourrait le croire le temps pendant lequel le CPU est prêt à être utilisé.
Cette valeur propre à l'environnement VMware vSphere informe du temps pendant laquelle la VM est **prête** à utiliser le CPU mais qu'elle ne peut pas l'utiliser car les ressources ne sont **pas disponibles** sur le CPU physique.
Quand je dis prête il faut entendre par là, prêtre à travailler et donc en demande de ressources processeur.

> Le CPU Ready est une valeur de VM, ne pas se fier à la valeur de l'ESXi ...

En cas de valeur élevée, il faut décharger l'ESXi hôte ou réserver des ressources CPU.

CPU Co-Stop Time

Le CPU Co-Stop est quant à lui à prendre en compte sur les VMs **SMP**, c'est-à-dire multi vCPU, en effet il sera à zéro sur une VM mono vCPU.
Cette valeur est similaire/liée au Ready Time, si elle est élevée c'est qu'un vCPU est en **attente** de schedule d'un autre vCPU.
Si par exemple le scheduler est sous contention, il aura plus facilement les ressources pCPU necessaires pour une petite VM d'1vCPU qu'une autre plus grosse de 4vCPU, il fournira donc moins souvent ou qu'une partie de la demande à ma grosse VM.

En cas de valeur élevée, il est conseillé de supprimer les snapshots (KB2000058 (https://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=2000058)) et diminuer le nombre de vCPU de la VM pour augmenter les opportunités de recevoir tout les pCPU necessaires.

Calcul CPU Ready/Co-stop

9

La KB2002181 (https://kb.vmware.com/kb/2002181) permet de connaître le calcul qui permet de convertir la valeur **Summation** en Millisecondes vers le Ready Time en %.

> *(CPU summation value / (<chart default update interval in seconds> * 1000)) * 100 = CPU ready %*

Dans le cas d'un chart en **Real Time**, nous avons 20 secondes d'intervalle, pour faire simple il suffit de diviser par 200 la valeur Summation (ms).

> *(1000 / (20s * 1000)) * 100 = 5%*
> *1000 / 200 = 5%*

Les valeurs de **% best practices** sont à prendre par vCPU.

<2.5%
Pas de soucis à se faire !

2.5%-5%
Contention minimale, à monitorer durant les pics.

5%-10%
Contention à prendre en compte, investiguer pour améliorer rapidement.

10%
Grosse contention, à résoudre d'urgence !

Avec une VM de 2 vCPU, si le **Ready Summation Total** en moyenne est de 1000ms, le CPU Ready Total moyen est de 5% donc de 2.5% par vCPU, ce qui reste **correct**.

Le CPU Co-stop time se calcule de la même façon, par contre la valeur **best practice** est de ne pas dépasser les 3%.

CPU Share

Les CPU Shares permettent d'attribuer une priorité à une VM ou à un Pool de VM en cas de contention.
En période de **contention**, le nombre de Shares va determiner le temps pendant laquelle le/les vCPU vont être schedulés.

La production aura par exemple une valeur High et la non-prod une valeur Low. Les valeurs de Share peuvent être modifiés en manuel pour **prioriser** par exemple ¾ du temps (75%) la production avec 75 Shares « Production » et 25 Shares « Non-Production »

Chaque High CPU Share équivaut à deux Normal CPU Share qui lui-même équivaux à deux Low CPU Share. En période de **contention** CPU, une VM de production (High) avec 1 vCPU sera donc schedulé 4 fois plus de temps qu'une VM de non-prod (Low)

Conseils

- N'ajouter qu'en cas de réelle nécessité des vCPU à une VM
- Monitorer l'utilisation CPU des Nodes (~ +50%)
- Vérifiez le Ready/Co-Stop Time des VMs gourmandes
- Configurer les Shares, Limites et surtout les Réservations avec grand soin, un trop grand nombre de VMs par pool peux les brider sévèrement.

Problème	Indicateur	Solution

Contention CPU d'une VM	Le CPU Ready Time est fréquemment au-dessus de 2000ms	Migrer la VM vers un ESXi moins chargé. Augmenter le CPU Share ou Reservation de la VM ou diminuer celui des VM concurrentes
Les ressources CPU sont insuffisantes pour la demande de la VM	L'utilisation CPU de la VM est constamment au-dessus de 80%	Réduire l'utilisation CPU du Guest OS, ajouter des vCPU ou migrer sur une ESXi plus véloce
Les ressources CPU sont insuffisantes pour la demande de l'ESXi	L'utilisation CPU de l'ESXi est constamment au-dessus de 80%	Réduire le nombre de VMs sur l'ESXi via vMotion ou ajouter des CPU physiques

Sources :

http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=1005362 (https://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=1005362)

http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=2000058 (https://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=2000058)

http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=1017926 (https://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=1017926)

http://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/techpaper/vmware-vsphere-cpu-sched-performance-white-paper.pdf (https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/techpaper/vmware-vsphere-cpu-sched-performance-white-paper.pdf)

Contention (https://www.virtualease.fr/tag/contention/) CPU (https://www.virtualease.fr/tag/cpu/) ESXTOP (https://www.virtualease.fr/tag/esxtop/) pCPU (https://www.virtualease.fr/tag/pcpu/) Ram (https://www.virtualease.fr/tag/ram/) Ready (https://www.virtualease.fr/tag/ready/) vCenter (https://www.virtualease.fr/tag/vcenter/) vCPU (https://www.virtualease.fr/tag/vcpu/) VMware (https://www.virtualease.fr/tag/vmware/) vSphere (https://www.virtualease.fr/tag/vsphere/) World (https://www.virtualease.fr/tag/world/)

MATHIEU (HTTPS://WWW.VIRTUALEASE.FR/AUTHOR/MATTEO/)

Je suis actuellement ingénieur spécialisé dans le design d'environnements cloud

virtualisés. Adepte des technologies de VMware, Nutanix, Citrix et Microsoft je propose à travers ce blog diverses astuces de troubleshooting.

‹ Check Point SmartConsole R80.20 – Erreur fatale à l'ouverture de Topology (https://www.virtualease.fr/check-point-smartconsole-r80-20-erreur-fatale-a-louverture-de-topology/)

Citrix Virtual Apps & Desktops – Analyser des traces CDF sans erreurs (Failed to find matching TMF for GUID) › (https://www.virtualease.fr/citrix-virtual-apps-desktops-analyser-des-traces-cdf-sans-erreurs-failed-to-find-matching-tmf-for-guid/)

vmware TECHNICAL SOLUTIONS PROFESSIONAL

vmware CERTIFIED ASSOCIATE 6 NETWORK VIRTUALIZATION

vmware CERTIFIED PROFESSIONAL 6 DATA CENTER VIRTUALIZATION

veeam TECHNICAL SALES PROFESSIONAL VMTSP

#TAGS

Azure (https://www.virtualease.fr/tag/azure/)

Backup (https://www.virtualease.fr/tag/backup/)

Certificat (https://www.virtualease.fr/tag/certificat/)

Citrix (https://www.virtualease.fr/tag/citrix/)

Cloud (https://www.virtualease.fr/tag/cloud/)

Command Prompt (https://www.virtualease.fr/tag/command-prompt/)

Console (https://www.virtualease.fr/tag/console/)

Contention (https://www.virtualease.fr/tag/contention/)

CPU (https://www.virtualease.fr/tag/cpu/)

Debian (https://www.virtualease.fr/tag/debian/)

ESXi (https://www.virtualease.fr/tag/esxi/)

Exchange (https://www.virtualease.fr/tag/exchange/)

GPO (https://www.virtualease.fr/tag/gpo/)

Hardware (https://www.virtualease.fr/tag/hardware/)

HP (https://www.virtualease.fr/tag/hp/)

vmware vEXPERT (https://communities.vmware.com/docs/DOC-34135?src=vmw_so_vex)

Microsoft Specialist
Implementing Microsoft Azure Infrastructure Solutions

Microsoft CERTIFIED Professional