VMware vSphere et le stockage

1. L'I/O Path

Avant toute chose, nous devons comprendre le cycle de vie d'un I/O depuis l'application jusqu'au stockage. Comme nous pouvons le voir dans le schéma ci-dessous, un I/O applicatif nécessite beaucoup d'étapes et de temps (à l'échelle de la machine bien sûr).

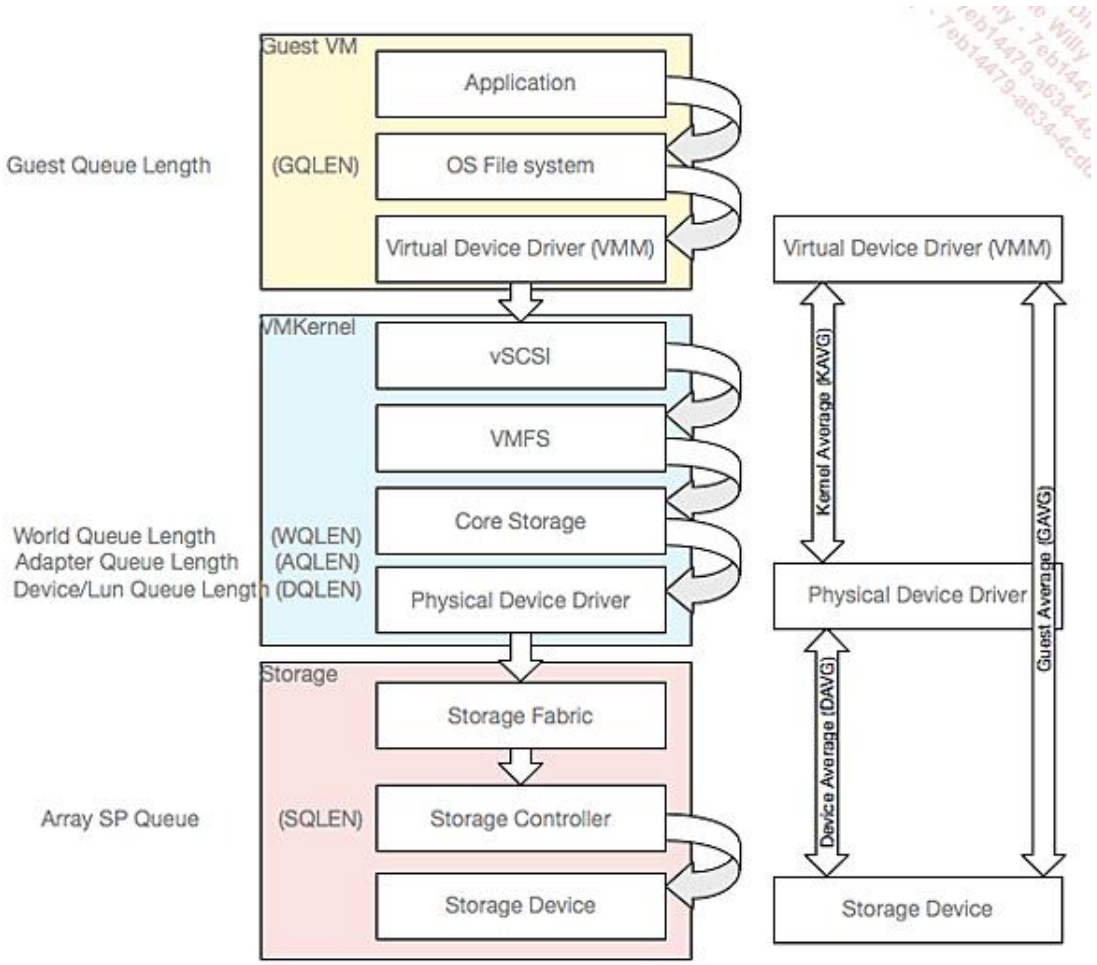
À chacun de ces niveaux, il existe différentes profondeurs dans la file de gestion des I/O :

OS (pilote de périphérique virtuel) : vNIC (LSI :32, PVSCSI : 64) ou vHBA (LSI : 128, PVSCSI : 255) si nous utilisons le NPIV.

VMkernel FC : 32, iSCSI : 128, NFS : 256, disque local : 256

Au niveau du stockage nous avons exactement les mêmes choses sur plusieurs niveaux (le chemin de l'I/O au niveau du Storage Processor, du tampon d'I/O du port, des disques...).

L'utilitaire ESXTOP et ses variantes permettent d'avoir une mesure de la latence à chacun de ces niveaux.



La partie droite concerne les statistiques de latence que l'on retrouve dans ESXTop.

La valeur GAVG correspond à la latence pour l'écriture et sa confirmation du point de vue de la VM (nous parlons ici d'un aller-retour - *roundtrip*). Elle est l'addition des valeurs DAVG et KAVG.

Le KAVG correspond à la latence liée au nombre de commandes au niveau du VMKernel.

Le DAVG correspond à la latence vue entre la carte de l'ESXi (HBA) et le stockage (ici nous parlons de la valeur du round trip).

Au centre se trouve l'I/O Path, ou le chemin d'une écriture sur le stockage.

La partie gauche concerne la longueur des files d'attente (buffer) ou queues, les valeurs sont disponibles via ESXTop. Donc nous ne parlons ici que des buffers accessibles via l'ESXtop. Il faut aussi prendre en compte les buffers au niveau de la HBA et de la Fabric.

Les API (VAAI, VASA, VDAP), le PSA, SIOC, les filtres I/O, etc. sont des éléments s'intégrant dans l'I/O Path.

2. API liées aux stockages

Les API pour application programming interfaces sont des commandes/primitives permettant de décharger les hyperviseurs d'une partie de la gestion du stockage. L'intérêt est d'éviter la gestion purement logicielle et de la remplacer par une gestion matérielle en envoyant les bonnes commandes aux serveurs de stockage, ce qui est bien plus efficace et rapide.

a. vSphere Storage API Array Integration - VAAI

Il existe deux manières de créer les VMDK ou disques durs virtuels des machines virtuelles :

Thick Provisioning, c'est-à-dire que l'on fournit l'ensemble de l'espace demandé ce qui entraîne une sous-utilisation de l'espace disque.

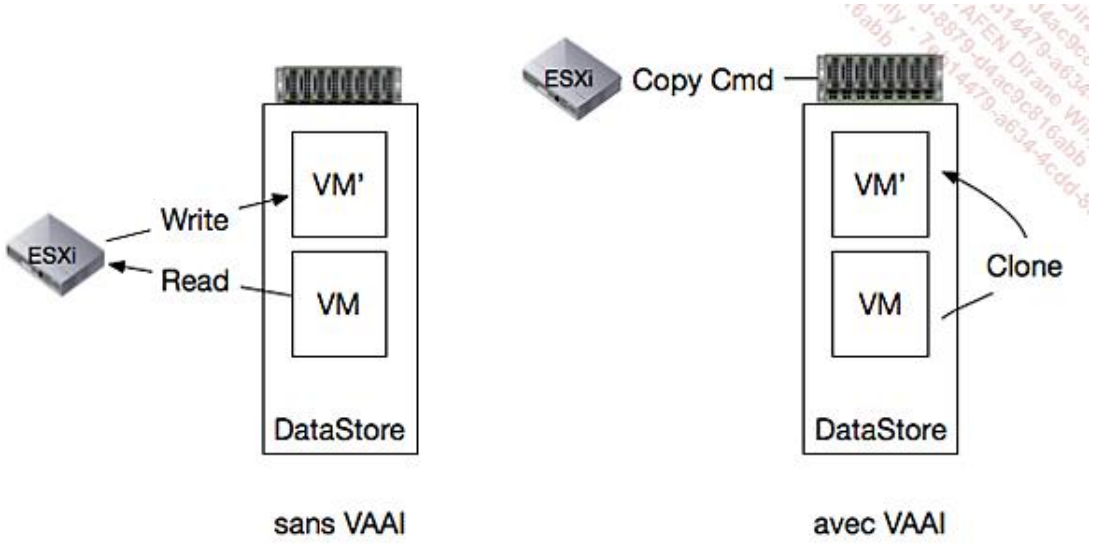
Thin Provisioning, c'est-à-dire que l'on fournit à la machine virtuelle uniquement l'espace qui lui est nécessaire. Par exemple, une machine virtuelle en Thin Provisionning à qui on alloue 20 Go d'espace disque au niveau de la machine virtuelle verra bien les 20 Go, tandis qu'au niveau des fichiers de la machine virtuelle nous ne verrons qu'un fichier VMDK de 8 Go. Cela peut créer une surallocation de l'espace disque ou over provisioning. Il est important de surveiller cette surallocation afin de ne pas être à court d'espace disque. Pour cela nous pouvons utiliser le plug-in VAAI (https://kb.vmware.com/kb/1021976).

Le VAAI est une API, se trouvant entre le VMKernel et les baies de stockages. Le but de cette API est de déléguer certaines fonctionnalités très génératrices d'I/O aux baies de stockage. Il est nécessaire d'implémenter le logiciel du constructeur de baies (sous forme de fichier VIB) sur les ESXi afin d'avoir accès aux fonctionnalités du VAAI.

Nous trouvons dans le VAAI :

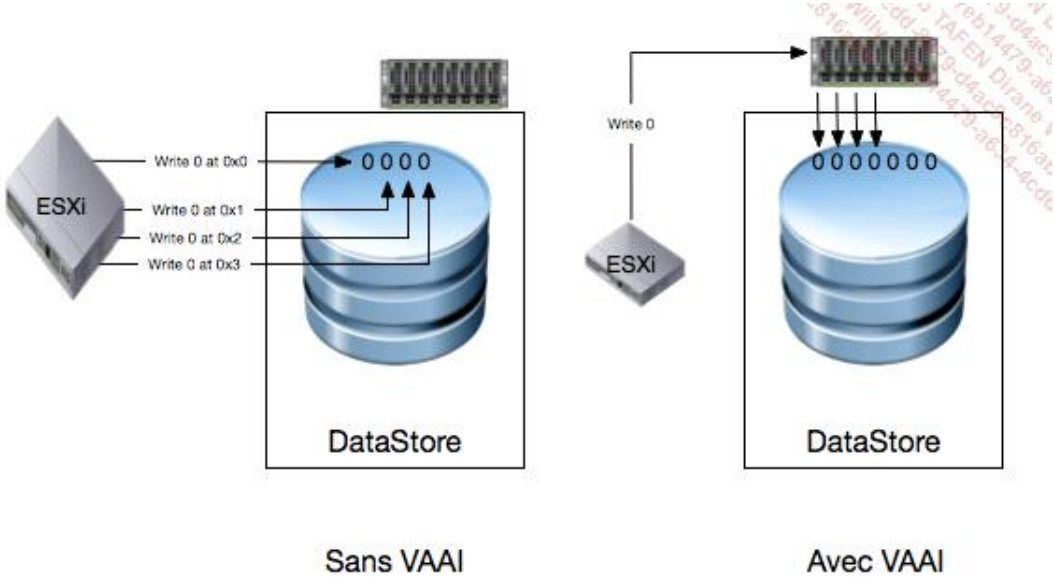
a. Le hardware assistant fournissant :

- Pour le stockage en mode bloc :
  - Full Copy/Clone Block qui autorise les baies de stockage à faire la copie des données sans aucune intervention (lecture ou écriture) de la part des serveurs ESXi. Cela permet de réduire le temps et l'impact sur le réseau lors de clonage, de mise à disposition ou de migration via vMotion.

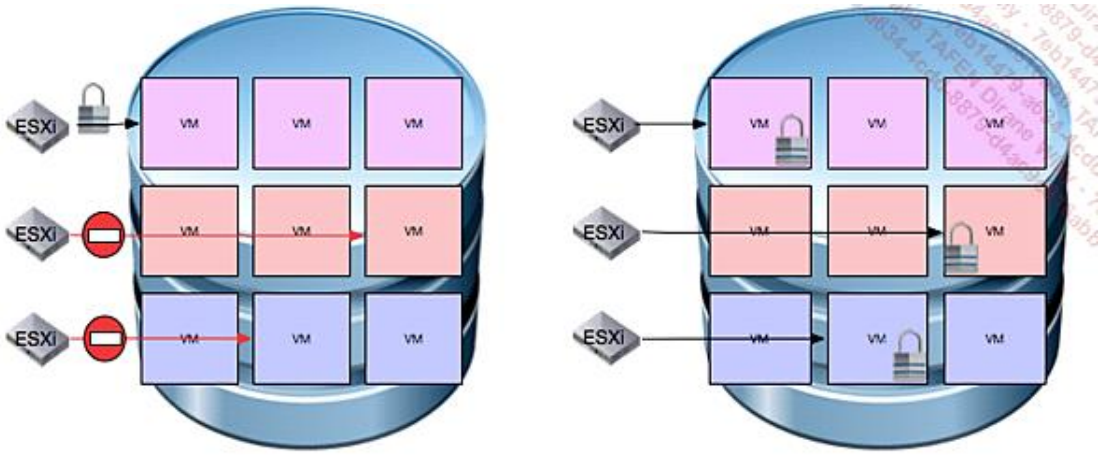


- Block Zeroing : une fonctionnalité permettant d'utiliser les capacités de la baie afin de remettre à zéro une partie du datastore suite à une suppression de VM, avant de le mettre à disposition pour la création de nouvelles VM.

  Cela permet de réduire le temps lors de la création de VM et du formatage des disques virtuels (VMDK). Cette fonctionnalité est utilisée par le Thick Provision Eager Zero.



- Hardware assisted locking / Atomic Test and Set (ATS) qui permet un verrouillage au niveau du secteur du disque (block) contre le verrouillage de la LUN par le verrouillage SCSI. Cela permet à chaque ESXi de gérer de manière granulaire (au niveau de la VM) la mise à jour des métadonnées. Auparavant, la mise à jour des métadonnées via le verrouillage SCSI ne pouvait être faite qu'unitairement (un ESXi à la fois)



Il est possible de désactiver (https://kb.vmware.com/kb/1033665) ces options en modifiant les valeurs des paramètres avancés suivants :

- HardwareAcceleratedMove

  Ce paramètre correspond au Full Copy, pour le désactiver, il faut configurer DataMover.HardwareAcceleratedMove à 0.

- HardwareAcceleratedInit

  Ce paramètre correspond au Block Zeroing, pour le désactiver, il faut configurer DataMover.HardwareAcceleratedInit à 0.

- HardwareAcceleratedLocking

  Ce paramètre correspond à l'ATS, pour le désactiver, il faut configurer /VMFS3/HardwareAcceleratedLocking à 0.

Attention, il est possible de désactiver l'ATS à plusieurs niveaux (https://kb.vmware.com/KB/2146451).

- Pour le NAS
  - Full File Clone, est l'équivalent de la fonctionnalité Full Copy/Clone Block, qui permet de cloner des fichiers à la place des blocs.
  - La réservation d'espace qui autorise et permet l'allocation d'espace pour les disques de format thick où avec l'ensemble de l'espace demandé.

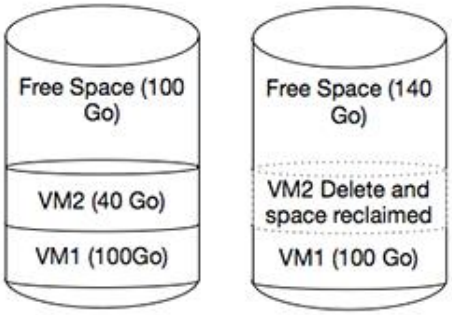
    Par défaut, sur un datastore NFS (NAS), l'allocation d'espace par défaut est en mode Thin provisioning ou l'ensemble de l'espace demandé n'est pas alloué.
  - Lazy File Clone, qui permet à Horizon View de créer des clones liés sur une baie NAS.
  - Extended file statistics qui autorise la baie à fournir un rapport détaillé sur l'utilisation de l'espace de stockage par rapport à l'espace de stockage alloué (allocation par rapport à l'usage).

b. L'intégration du Thin Provisionning avec les LUNs et la gestion de l'espace sur ces dernières :

- Le Thin provisionning, par défaut sur un LUN, l'espace demandé est alloué à la VM. Le Thin provisionning autorise à faire de la surallocation d'espace en ne fournissant à la VM que l'espace qui lui est nécessaire

- Une interaction plus fine entre l'ESXi et la baie de stockage permettant :
  - La supervision de l'espace alloué en mode Thin provisionning (permettant la surallocation) afin de prévenir les problèmes d'espace disque.
  - La récuperation d'espace non alloué suite à des Storage vMotion.

c. UNMAP (https://kb.vmware.com/KB/2057513) est une (commande) primitive permettant la récupération des blocs sur des LUNs en mode Thin Provisioning. Le mécanisme d'UNMAP a été amélioré afin de limiter son impact par l'intégration de récupération incrémentale. Son utilisation était jusqu'à présent manuelle. Depuis la version 6.5 elle est automatisable via l'interface utilisateur. La granularité d'UNMAP est d'un bloc de 1MB. UNMAP depuis la version 6.0 permettait de réclamer l'espace non alloué des partitions sous Windows 2012 R2 et RedHat 6.5. Avec vsphere 6.5, UNMAP permet de faire cette même opération sous Linux (http://www.codyhosterman.com/2015/04/direct-guest-os-unmap-in-vsphere-6-0/ et https://storagehub.vmware.com/#!/vsphere-core-storage/vsphere-6-5-core-storage/unmap-4).



La récupération d'espace

Le plug-in VAAI s'installe (https://kb.vmware.com/kb/2008939) de la manière suivante :

```
Esxcli software vib update -v [URL]
Esxcli software vib update -d [path_to_Vib]
```

Où [URL] est l'URL d'accès au fichier VIB et `[path_to_Vib]` est le chemin d'accès local au fichier VIB.

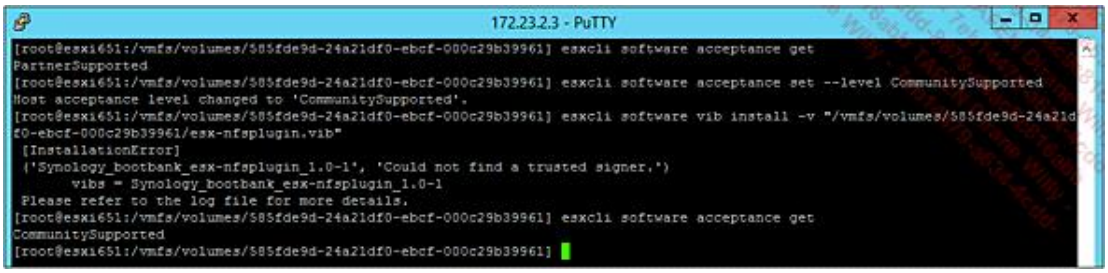
Attention au niveau d'acceptation. Le niveau d'acceptation (https://pubs.vmware.com/vsphere-65/index.jsp#com.vmware.vsphere.upgrade.doc/GUID-27BBBAB8-01EA-4238-8140-1C3C3EFC0AA6.html) est une configuration de l'ESXi qui détermine quel niveau de validation des packages VIB nous pouvons installer.

Il existe quatre niveaux d'acceptation :

- **VMwareCertified** : les VIB avec ce niveau sont soumis à des tests minutieux équivalents aux tests d'assurance qualité réalisés en interne chez VMware pour la même technologie.
- **VMwareAccepted** : les VIB avec ce niveau d'acceptation sont soumis à des tests de vérification minutieux, mais ces tests ne concernent pas toutes les fonctions du logiciel. Le partenaire exécute les tests et VMware vérifie le résultat. Actuellement, les fournisseurs CIM et les plug-ins PSA font partie des VIB publiés à ce niveau.
- **PartnerSupported** : les VIB avec le niveau d'acceptation PartnerSupported sont publiés par un partenaire en qui VMware a confiance. Le partenaire effectue tous les tests. VMware ne vérifie pas les résultats.
- **CommunitySupported** : le niveau d'acceptation CommunitySupported est destiné aux VIB créés par des individus ou des entreprises en dehors des programmes de partenariat de VMware.

Le plus restrictif est le niveau VMwareCertified.

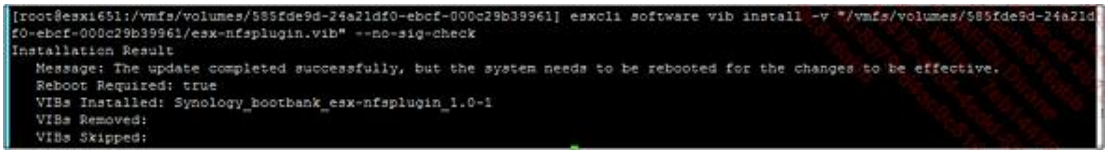
Dans notre cas, nous avons installé la VIB VAAI de Synology sur chacun de nos serveurs ESXi. Nous avons reçu le message d'erreur suivant : « Could not find a trusted signer. »



Ceci est un problème de signature de la VIB.

Afin de forcer l'installation nous avons ajouté à la commande le paramètre `-no-sig-check` (permettant de désactiver la vérification de la signature du package VIB). L'installation s'est déroulée sans problème.

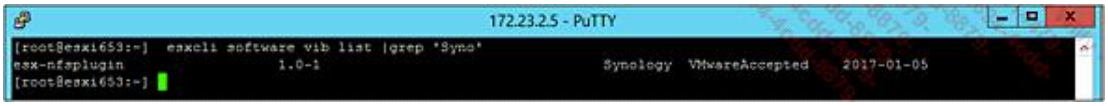
Attention, cette opération est déconseillée en environnement de production pour des questions de support technique.



Après le reboot de l'ESXi, vérification s'impose à l'aide de la commande :

```
Escli software vib list|grep 'Syno'
```

Pour être sûr que le paquetage VIB est bien présent et avec la bonne version.



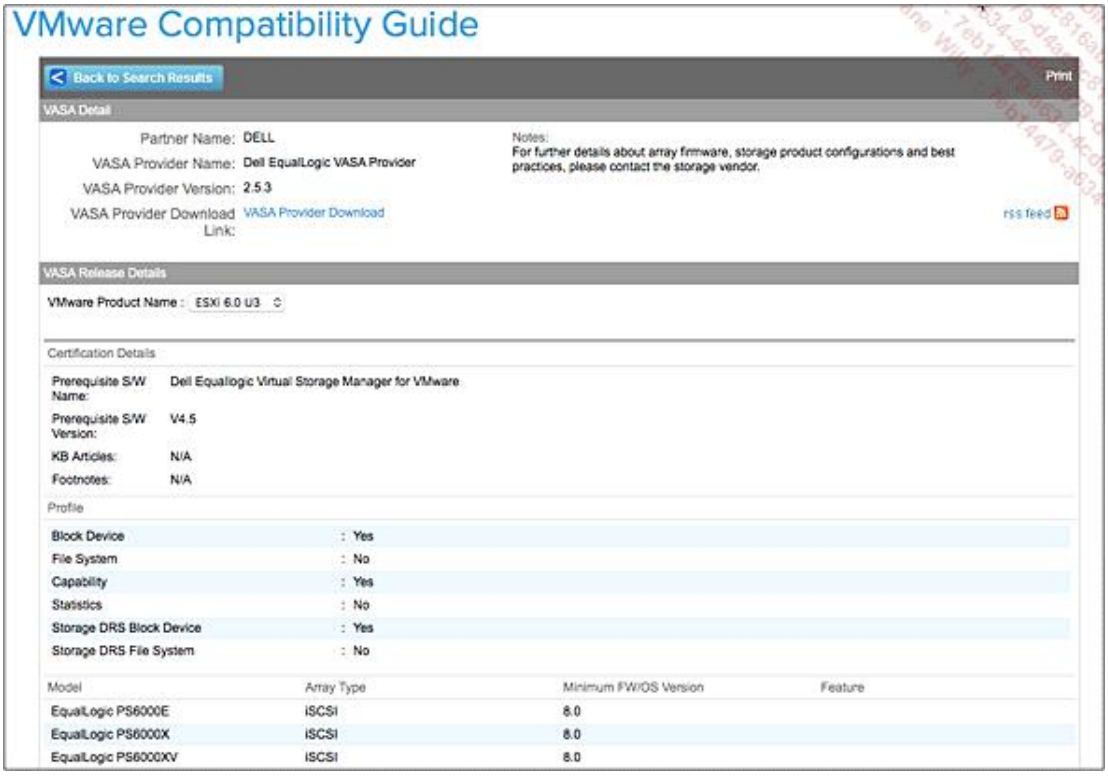
David STAMEN propose un script en Powershell pour effectuer cette opération : http://davidstamen.com/powercli/using-powercli-to-install-host-vibs/

La méthode la plus simple et la plus efficace reste l'Update Manager. Cela évite d'oublier un ESXi, permet de programmer et d'automatiser les installations et mise à jour des plug-ins.

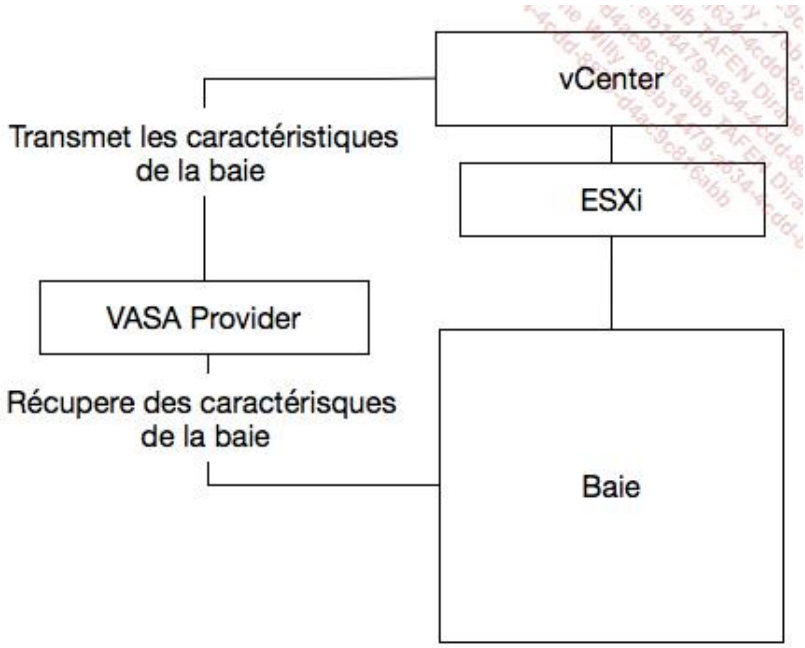
b. vSphere API for Storage Awareness - VASA

Le VASA est un plug-in permettant au vCenter d'avoir connaissance des aptitudes (I/O, type de périphérique de stockage, compression...) et capacités de la baie (RAID, type de disque, SLA...), sur laquelle se trouvent les VMs. VMware fournit directement un Storage provider pour le vSAN. Dans le cadre du stockage d'autres constructeurs, VMware fournit une liste de compatibilité : http://www.vmware.com/resources/compatibility/search.php?deviceCategory=vasa

Exemple pour le plug-in Dell Equalogic Virtual Storage Manager for VMware :



Les plug-ins sont généralement installés sur un serveur auxiliaire.



VASA est employé dans le cadre du Storage DRS, de la politique de gestion de stockage, des vVOLs... Les plug-ins VASA sont fournis par les constructeurs de baies de stockages.

c. Politique de stockage ou Storage Policy Based Management (SPBM)

Les politiques de stockage sont la déclinaison, dans le monde VMware du SDS (*Software Define Storage*). Nous pourrions les comparer aux GPO Windows. Elle se base sur 4 éléments :

- Les fournisseurs de stockage (VASA).
- Les Tags qui sont définis par l'administrateur et permettent de qualifier les capacités du stockage avec des informations non fournies par VASA.
- Le vCenter qui centralise et permet de créer les règles de validité propre à chaque politique de stockages.
- Les VMs sur lesquelles on applique les politiques de stockage. Les politiques de stockage s'appliquent sur les fichiers constituant la VM, il est donc possible par ce biais d'avoir les fichiers VMDK sur plusieurs stockages différents, tout comme les fichiers de log, d'échange et de configuration (vmx).

Il ne faut pas confondre SPBM qui permet de choisir le placement des fichiers VMDK en fonction des qualités de la baie de stockage avec les affinités entre les fichiers VMDK que propose le Storage DRS.

Il est possible de faire une utilisation conjointe de SPBM et de Storage DRS. Pour cela il faut créer l'option avancée du cluster DRS : EnforceStorageProfiles et utiliser un cluster de datastores ayant différentes caractéristiques (I/O, type de périphérique de stockage, etc.). Cette option permettra de renforcer les règles d'affinités entre les VMDK, en se basant sur les caractéristiques du stockage.

Le paramètre `EnforceStorageProfile` peut avoir trois valeurs :

- Valeur 0 : pas d'utilisation conjointe entre le SPBM et le storage DRS.
- Valeur 1 : cela équivaut aux règles DRS de type « devrait », aussi appelées soft rules. Cela signifie que le storage DRS peut violer les contraintes de conformités imposées par SPBM. Les règles définies au niveau du storage DRS sont prioritaires par rapport aux règles de SPBM.
- Valeur 2 : cela équivaut aux règles DRS de type « doit », aussi appelées hard rules. Cela signifie que le storage DRS ne peut violer les contraintes de conformités imposées par SPBM. Les règles définies au niveau de SPBM sont prioritaires sur les règles du storage DRS. Cela peut générer l'erreur suivante : ne peut pas corriger la violation de la règle d'anti-affinité (*could not fix anti-affinity rule violation*).

d. Multipathing

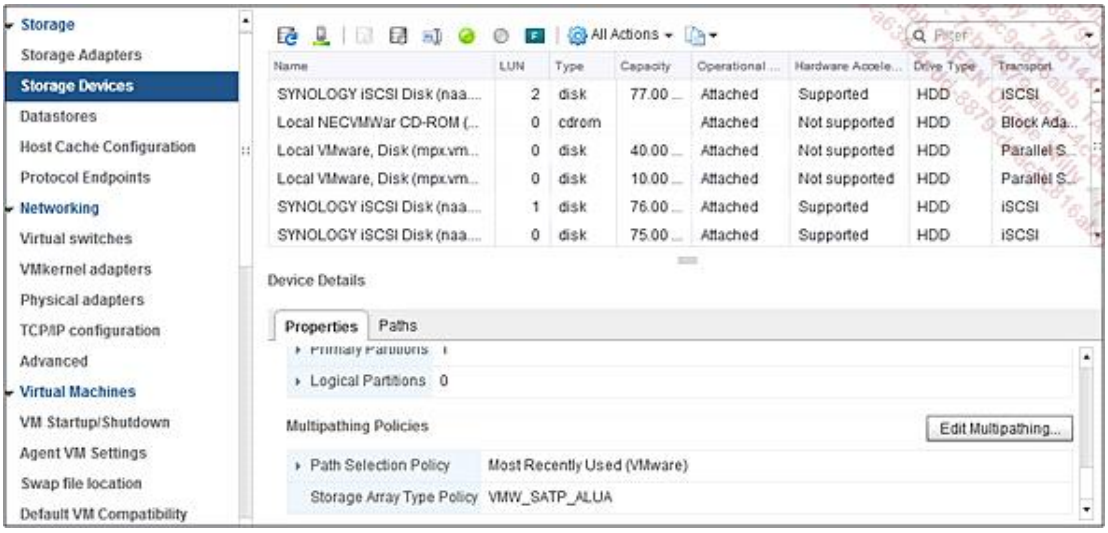
https://kb.vmware.com/KB/1011340

Il est important dans le cadre de la mise en place de résilience sur les chemins d'accès aux baies de stockage de prendre en compte le MPIO (*MultiPathing Input/Output*) ou en français la gestion d'accès via chemins multiples pour les entrées/sorties. Un chemin MPIO peut avoir quatre états différents :

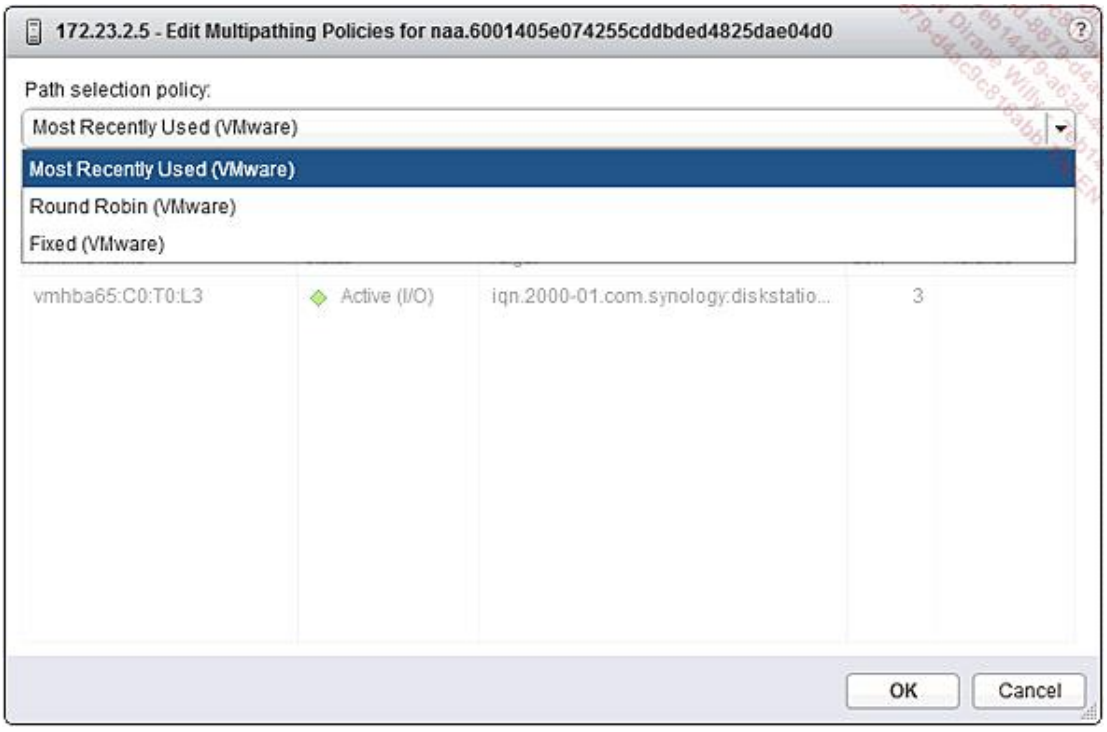
- Active (Actif) : ce chemin est disponible pour accéder aux LUNs, et transporte des I/O.
- Standby (en attente) : ce chemin est en attente d'utilisation en cas de perte du chemin actif.
- Disable (Désactivé) : le chemin est désactivé et aucune donnée ne peut l'utiliser.
- Dead (Mort) : accès aux LUNs impossible via ce chemin.

Ces implémentations de MPIO sont disponibles via le PSA (*Pluggable Storage Architecture*).

Au niveau d'un hyperviseur via le client web vSphere, il est facile de modifier la politique de gestion des chemins d'accès via l'interface de gestion des périphériques de stockage ou/et des Protocol Endpoints (proxy d'entrées sorties pour vVOL). Nous sélectionnons le périphérique de stockage et dans ses propriétés, nous descendons jusqu'à avoir accès à la politique de gestion du multipathing. Et nous cliquons sur éditer.



Nous sélectionnons la politique que nous voulons utiliser.

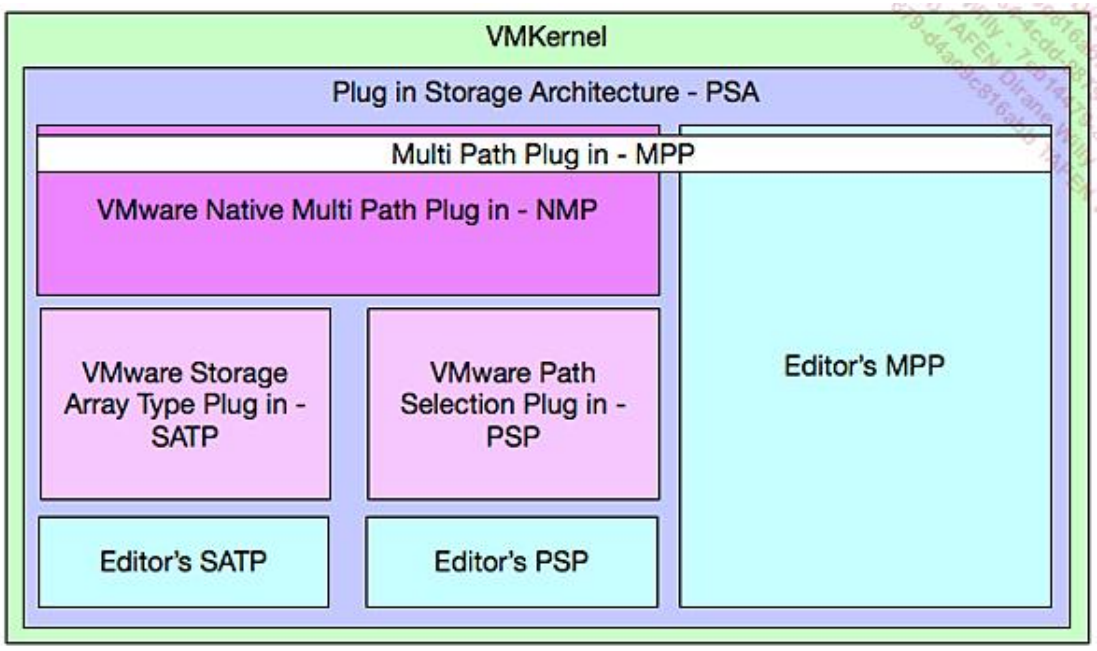


Si vous voulez modifier ce paramètre en ligne de commande, nous vous recommandons ce site : http://buildvirtual.net/how-to-change-the-default-psp-for-an-satp-on-esxi/

Pluggable Storage Architecture - PSA

Le PSA (https://pubs.vmware.com/vsphere-65/index.jsp#com.vmware.vsphere.storage.doc/GUID-C1C4A725-8BE4-4875-919E-693812961366.html) est une boîte à outils (framework). Il est composé soit d'une solution éditeur de MPP (*Multi Path Plug-ins*), soit d'une la solution VMware MPP qui est le VMware NMP (*Native Multi Path Plug-ins*). Le NMP est composé par deux éléments : le VMware SATP (*Storage Array Type Plug-ins*) et le VMware PSP (*Path Selection Plug-in*). Les éditeurs peuvent aussi fournir leurs SATP et PSP.

Le PSA s'intègre au niveau du VMKernel.



Le rôle du MPP et du NMP est identique, la différence entre ces deux éléments tient au fait que l'un (le NMP) est de VMware tandis que l'autre le MPP est fourni par des éditeurs tiers (constructeur de baie de stockage).

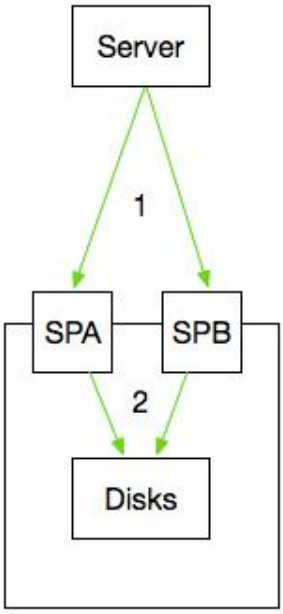
Le rôle de ces composants est de fournir et de garantir aux ESXi les chemins d'accès aux LUN tout en gérant l'équilibrage de charge (*Load Balancing*) et de gérer la défaillance d'un des accès (*fail-over*).

PowerPath/VE d'EMC est un exemple de MPP. Le MPP et le NMP fonctionnent en parallèle au sein du PSA, chacun gérant ses propres accès aux stockages.

Quelques détails sur les fonctions des plug-ins :

Le SATP gère l'accès aux stockages, la qualité des liens d'accès aux stockages (optimisé/non-optimisé). En fonction de la baie, le SATP choisi sera différent, que ce soit pour l'accès ou la capacité des SP via les Claims Rules. Il existe trois types de stockage à ce niveau.

- Un SAN est dit en fonctionnement actif/actif lorsque les deux SP ont un accès simultané à une LUN, sans que cela n'entraîne une dégradation de la performance. En cas de perte d'un chemin d'accès aux données, le second chemin reste disponible. Dans les stockages dits Actif/Actif, nous distinguons deux sous-types :
  - Le stockage Actif/Actif Symétrique

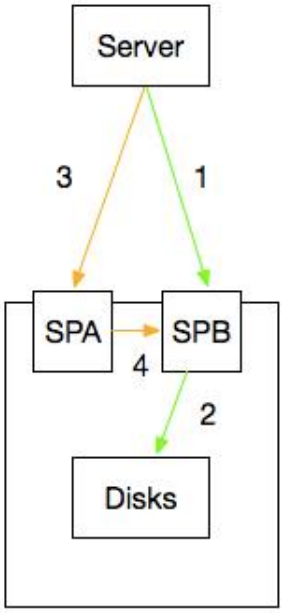


1 : accès simultané aux SPs (les deux SPs sont propriétaires de la LUN)

2 : écriture sur le stockage

- Le stockage actif/actif asymétrique (ALUA - *Asymetric Logical Unit Access*)

- Un SAN dit ALUA fournit deux niveaux d'accès aux LUNs à travers ses SP. ALUA permet aux serveurs de déterminer l'état des ports et de prioriser les chemins d'accès. Se basant sur cette détermination, le chemin primaire est le plus rapide, le chemin secondaire est plus lent.



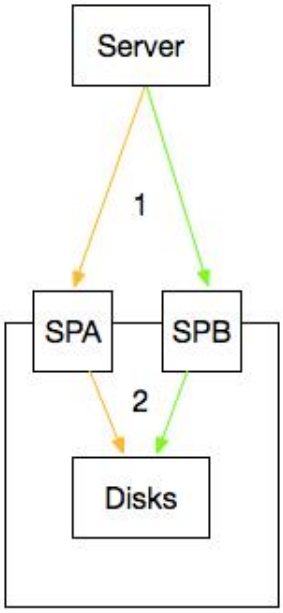
1 : chemin d'accès au SP propriétaire de la LUN

2 : écriture sur le disque

3 : chemin d'accès au SP non propriétaire de la LUN

4 : transfert l'I/O vers le SP propriétaire de la LUN

- Un SAN est dit actif/passif lorsque seul un SP fournit l'accès à une LUN. Le second SP agit en tant que SP de secours pour cette LUN. Il devient actif après la perte du premier SP fournissant l'accès à cette LUN. Dans le même temps, le second SP, peut être le SP primaire pour d'autres LUNs.



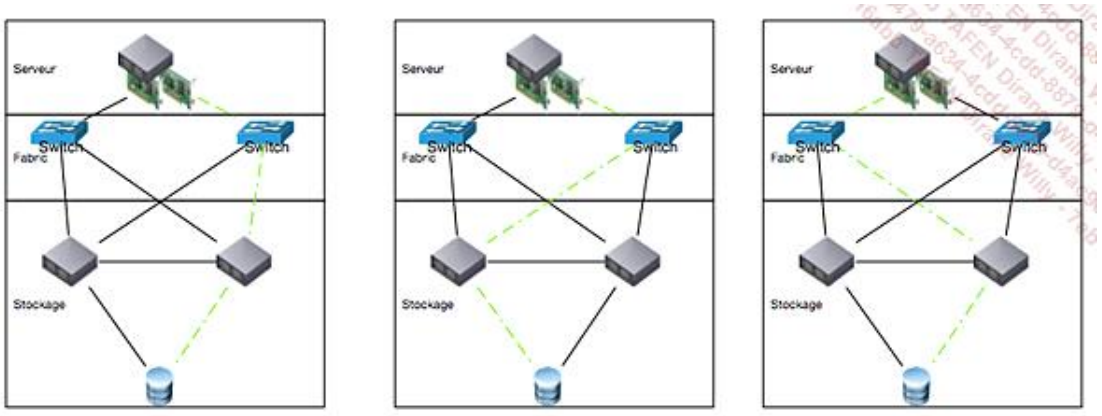
1 : droite, chemin d'accès au SP actif (Propriétaire de la LUN)

1 : gauche, chemin d'accès au SP Passif en cas de perte du SP Actif

2 : droite et gauche, écriture de l'I/O sur le disque en fonction de SP

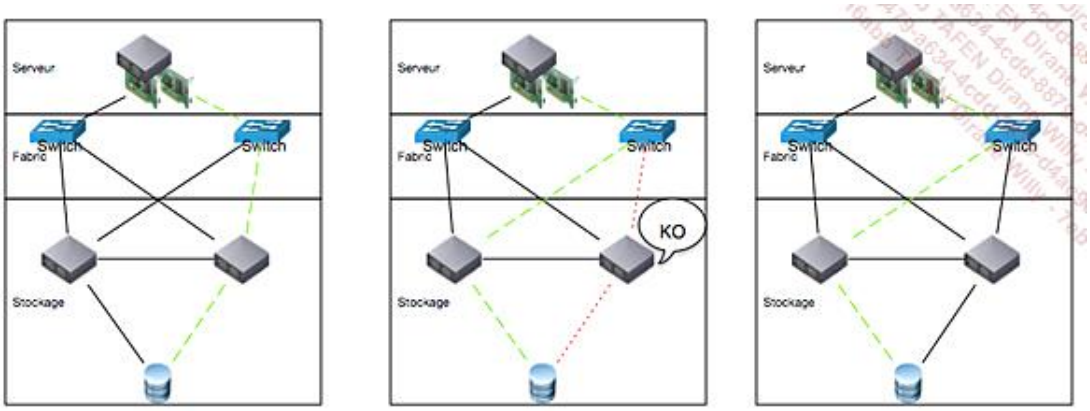
Le PSP est garant de la sélection du chemin d'accès au stockage. Le PSP fournit trois types de gestion d'accès aux stockages.

- Round robin ; comme son nom l'indique, le RR (*Round Robin*) est un tourniquet permettant la distribution des I/O entre les différents chemins disponibles ce qui active en même temps une répartition de la charge. C'est le cas lors de la connexion à des SAN actif/actif, dans le cas d'un SAN actif/passif, la répartition de charge se fera entre les chemins actifs.



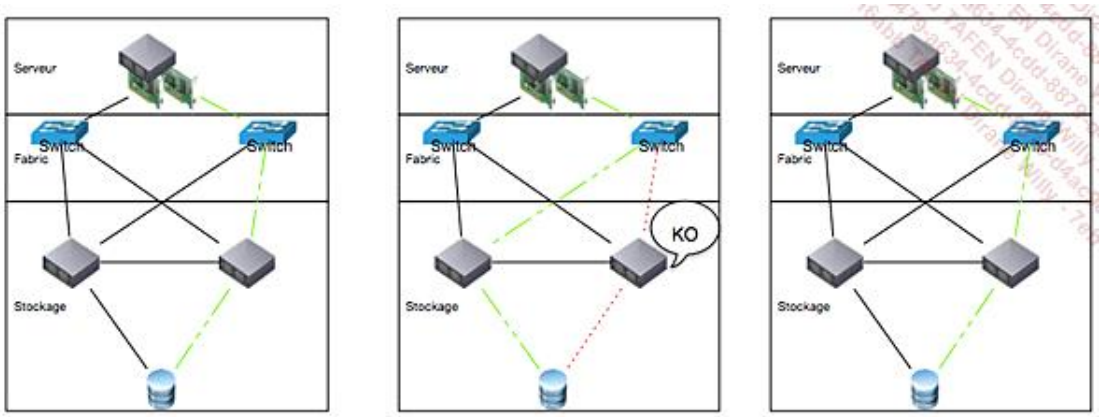
- MRU (*Most Recently Used*) ; le MRU, ou chemin utilisé le plus récemment, est lié à la sélection du premier chemin découvert lors du boot de l'ESXi. Si ce chemin devient non disponible, l'ESXi va basculer sur un chemin alternatif. Lorsque le premier chemin est de nouveau disponible, l'ESXi ne rebascule pas dessus.

C'est la configuration par défaut pour les baies de stockage actives/passives.

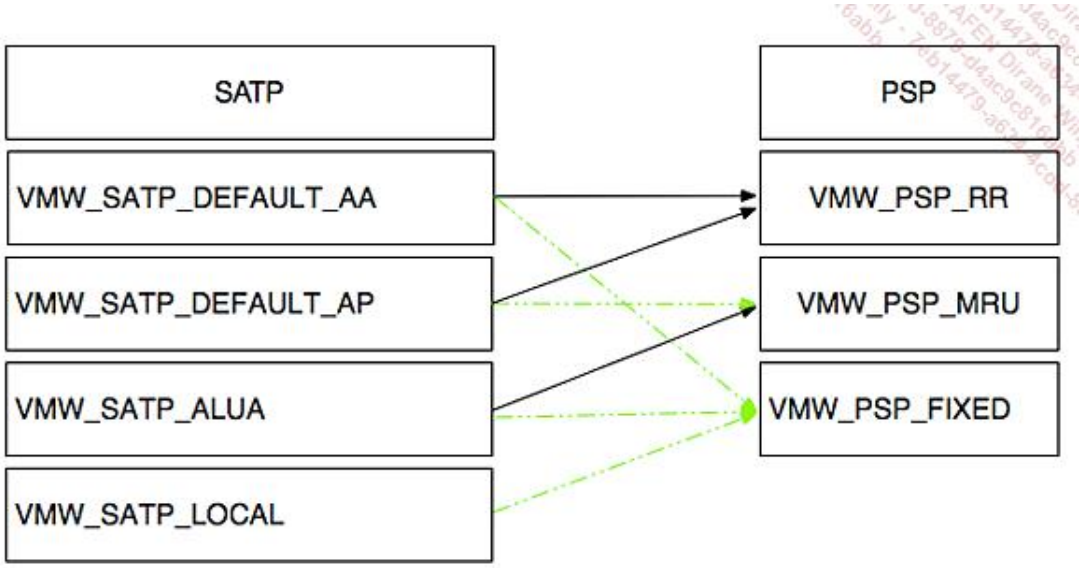


- Fixed ; comme pour le MRU, le premier chemin découvert lors du boot est le chemin actif, qui sera utilisé. Il restera le chemin préféré tant qu'il restera actif, en cas de perte d'accès, le chemin qui était alternatif devient le chemin préféré.

Il y a un bémol toutefois, si l'on déclare spécifiquement un chemin préféré (*Prefered Path*), ce chemin restera même en cas de perte d'accès à la LUN. Cette configuration MPIO est la configuration par défaut pour les baies de stockage fonctionnant en actif/actif.

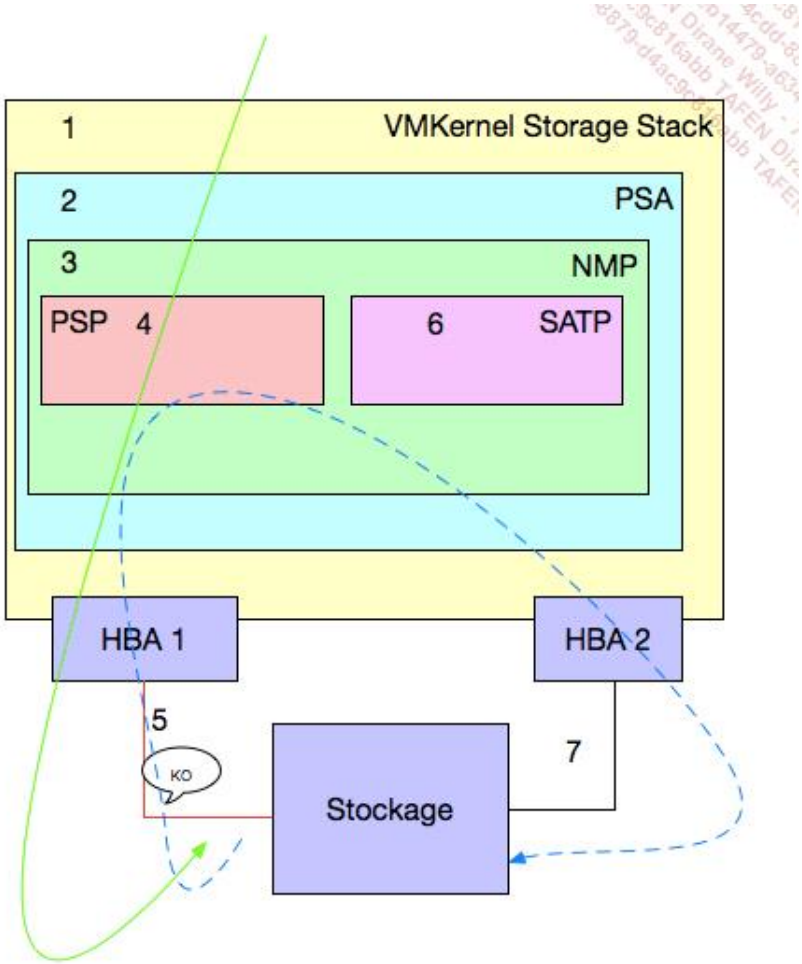


Voici l'association entre les différents SATP et PSP de VMware. En pointillé la configuration par défaut, en noir les configurations possibles.



Source : https://kb.vmware.com/kb/1011340

Il nous reste à voir comment est pris en charge un I/O à travers le PSA en pointillé. Lorsque l'I/O arrive dans le VMKernel sur la partie stockage, il passe de VMKernel au NMP via le PSA (1-2-3), le NMP appelle le PSP (4) afin de déterminer quel chemin l'I/O va prendre (5) en fonction du PSP appliqué (RR, ALUA, Fixe). En cas d'un retour en erreur en pointillé, de la part du stockage, le NMP (3) demande au SATP (6) d'inactiver le chemin en erreur, et de passer le chemin alternatif en actif, l'I/O est renvoyé à ce nouveau chemin actif (7).



Claim Rules

En anglais « claim » est une revendication, un droit d'accès ou à avoir quelque chose. Dans le contexte d'un serveur ESXi, une Claim Rule est utilisée par le PSA afin de déterminer quel MPP/NMP doit revendiquer pour gérer l'accès à une LUN en particulier.

Pour avoir une vue en profondeur des claims rules, nous vous recommandons le blog de Guido HAGEMANN : http://virtualguido.blogspot.fr/2016/09/vmware-esxi-claim-rules-unleashed.html, ainsi que celui de Sébastien BARYLO : http://vmwaremine.com/2014/07/07/manage-psa-claimrules-satp-rules-esxcli/#sthash.x3MrN3I7.dpbs

e. vSphere APIs for I/O Filtering (VAIO)

Les filtres I/O fournissent des services aux données (data service). Ces filtres sont développés par VMware, les partenaires (SANDisk...) et quelques sociétés non partenaire (PernixDATA - racheté par Nutanix, Zerto). Ces filtres se trouvent sous forme VIB à installer sur les ESX et s'intègrent dans le vCenter.

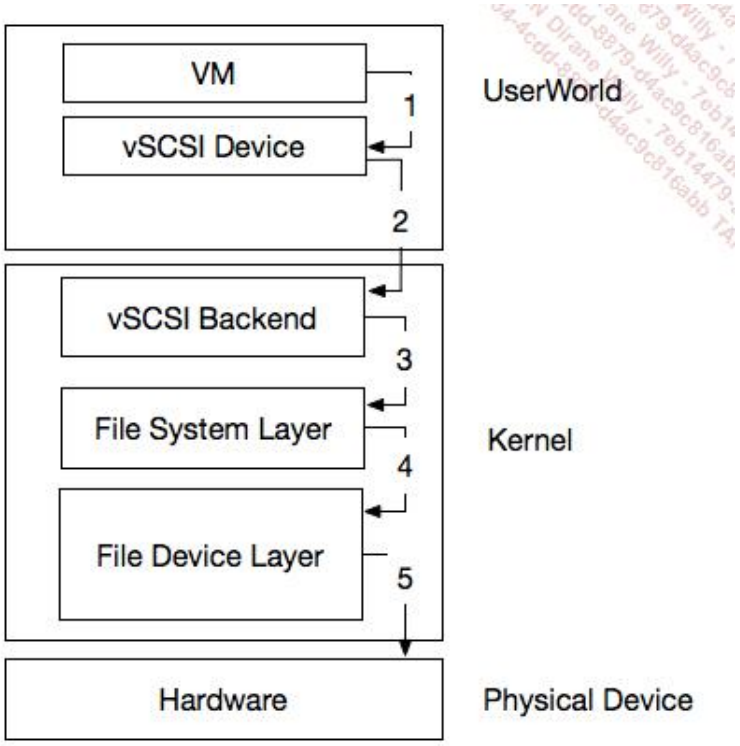
L'architecture de VAIO est composée de deux éléments :

- L'I/O Filter se situant au niveau de la VM.
- Le framework VAIO se situant au niveau du VMkernel.

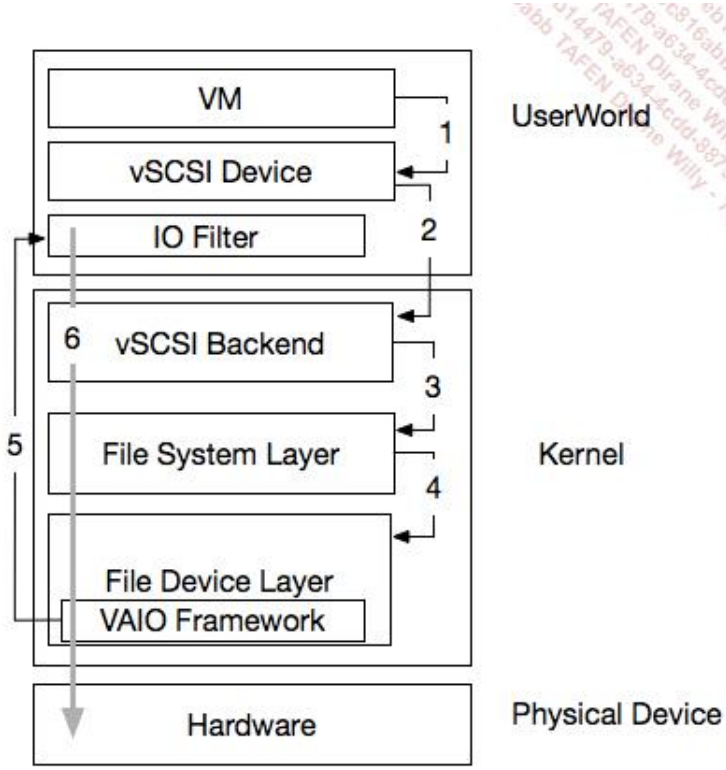
Actuellement, VAIO est utilisé dans les cadres suivants :

- Réplication de l'ensemble des I/O en écriture vers une cible extérieure (serveur ESXi ou Cluster).
- Chiffrement (VM encryption)
- Tampon (Cache) : utilisation d'un périphérique de stockage de type flash local à l'hôte ESXi.
- Storage I/O Control : gestion des I/O au niveau de la baie de stockage.

Dans la version 6.0 de vSphere, seul le tampon et la réplication étaient disponibles.



Ce schéma, reprend le chemin de l'IO path sans VAIO. L'I/O est généré par la VM et est inscrit sur le périphérique de stockage.



Ce schéma reprend le chemin de l'I/O path avec le VAIO. L'I/O est généré par la VM, jusqu'au Framework VAIO se trouvant au dernier niveau avant l'accès au périphérique de stockage (file device). Au niveau du framework VAIO (5), une vérification des filtres pour la VM est faite, dans le cas où il n'y a pas de filtre, l'I/O est écrit, sinon il est renvoyé vers l'I/O filter pour exécuter le data service associé avant d'être directement envoyé sur le stockage (6).