Périphérique de stockage

1. Le disque dur

Le principe de fonctionnement d'un disque dur est au croisement du vinyle et de la cassette audio. Il est constitué de plateaux magnétiques et de têtes de lecture.

Un air de famille entre le gramophone et le disque dur ne trouvez-vous pas ?

Source : https://pixabay.com/en/schell-corner-plate-gramophone-78rpm-1655188/

Source : https://pixabay.com/en/hard-drive-hdd-macro-disk-computer-463922/

La ressemblance atteint rapidement ses limites malgré tout.

Le disque dur est composé physiquement par le CHS ou géométrie du disque :

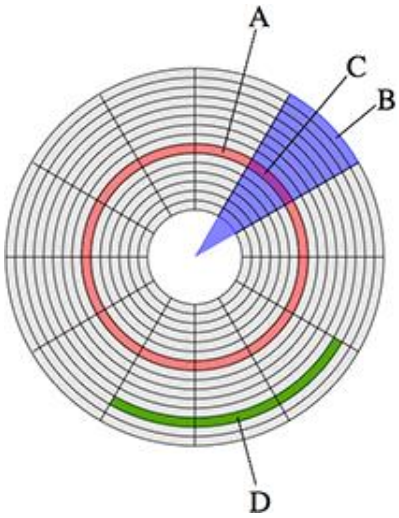
- Le cylindre (*cylinder*) correspond à l'ensemble des différentes pistes sur un même emplacement à travers l'ensemble des plateaux.
- La tête (*head*) ou tête de lecture/écriture qui lit et écrit les pistes.
- Le secteur (*sector*) ou secteur géométrique correspond à la portion du disque comprise entre deux rayons.

Il faut ajouter à cette liste :

- La piste (*track*) correspond à un cercle sur une face des plateaux magnétiques dont le centre est l'axe de rotation.
- Le secteur de piste correspond au croisement d'une piste et d'un secteur géométrique. C'est ce que l'on nomme un bloc.

- Le cluster correspond à un ensemble de secteurs de pistes contiguës. Historiquement il a une taille de 512 octets, mais avec l'augmentation de l'espace, cela crée des baisses de performance, de la lenteur pour être plus exact. Afin de pallier ce problème, les constructeurs de disques ont défini une nouvelle taille de cluster de 4 ko.

Afin de garder la rétrocompatibilité, les systèmes modernes ont une fonction d'émulation connue sous le nom de 512e pour émulation 512 octets. Windows supporte cette norme depuis Windows Vista à condition d'avoir le correctif KB 2553708. Le 512e est supporté par vSphere depuis la version 6.5 via le système de fichier VMFS 6 tandis que le 4 ko est supporté par vSphere Virtual SAN depuis la version 6.0 (https://kb.vmware.com/kb/2091600).



Cette figure provient de Wikipédia (https://en.wikipedia.org/wiki/Disk_sector). Elle permet de voir schématiquement la structure et les relations entre les différentes parties

- A : Piste (*track*)
- B : Secteur géométrique (*Geometrical sector*)
- C : Secteur de piste (*Track sector*)
- D : Cluster

Nous pouvons alors calculer la taille d'un disque dur en faisant la multiplication suivante :

Taille disque dur = Cylinders x Head x Sector par track x Taille du Secteur

Ce qui limitait la taille du disque à 8 Gigaoctets.

À titre d'exemple, pour un Seagate ST40000DM005 (http://www.seagate.com/www-content/product-content/barracuda-fam/barracuda-new/en-us/docs/100804656b.pdf), la géométrie du disque est la suivante.

Cylinders: 16,383

Read/write heads: 16

Sectors per track: 63

Taille du disque = 16383 X 16X 63 X 512

Taille du disque = 8455200768 bytes

Taille du disque = 8455200768 / 1073741824

Taille du disque = 7,84 soit environ 8 GB

Cela qui nous a forcés à évoluer du mode CHS vers le mode LBA - *Logical Bloc Addressing* ou Adressage de bloc logique issue du monde SCSI. Là où une géométrie spatiale se base sur trois axes (CHS), le mode de fonctionnement du LBA est linéaire. L'ensemble des secteurs ou blocs sont numérotés de manière consécutive de 0 (C=0, H=0, S=0) à N-1, où N est le nombre de secteurs garantis par le constructeur. Vu la taille des disques actuels, plusieurs Téraoctets, il est actuellement calculé sur 48bits. Le LBA est appelé par le BIOS via l'instruction 13h.

Il se calcule de la manière suivante :

LBA = ((cylinder * heads_per_cylinder + heads) * sectors_per_track) + sector - 1

Taille du disque= LBA* cluster size

La documentation du Seagate ST40000DM005 nous indique qu'il y a 7814037168 secteurs garantis.

Tandis que sa configuration LBA est la suivante :

4000787030016 Bytes =7814037168*512

4000787030016/1099511627776=3,63 TeraBytes

a. Master Boot Record

Historiquement, sur les périphériques magnétiques, les 512 premiers octets correspondent au Master Boot Record (MBR) pour les disques durs et secteur de boot pour les disquettes floppy. Ils se trouvent sur le premier secteur du périphérique. Le concept MBR a été introduit en 1983 par IBM avec le PC DOS 2.0. Comme nous le voyons, il n'a aucune résilience. La perte (virus, corruption, secteur défectueux) entraîne la perte des partitions.

Le MBR est constitué de :

- La table des partitions. Cette table permet d'avoir jusqu'à 4 partitions primaires. Afin d'aller au-delà de cette limitation, un type particulier de partitions a été conçu : la partition étendue, pouvant contenir des partitions logiques.

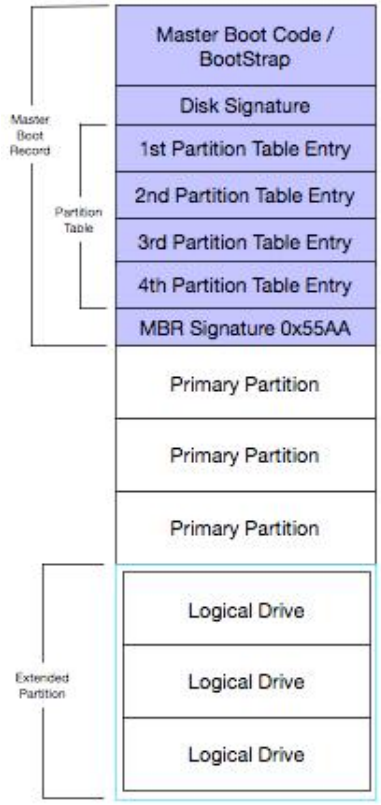


Schéma d'un MBR (https://technet.microsoft.com/en-us/library/cc976786.aspx)

- Bootstrap ou Boot Loader est un programme inclus dans le MBR. Son rôle est de déterminer la ou les partition(s) active(s) et d'en lire le premier secteur (*boot sector*). Il représente les 440 premiers octets du MBR.
- Disk signature correspond à l'identifiant du disque dur.

b. GUID Part Table

Initialement le LBA était sur 32 bits ce qui limitait la taille des disques à 2 TB. Liant ce problème aux limitations du MBR (4 partitions primaires, ou 3 partitions primaires et une partition étendue).

L'introduction du *Global Unique Identifiant Part Table* (GPT) est liée au standard *Unified Extensible Firmware Interface* (UEFI) qui est le remplaçant du BIOS.

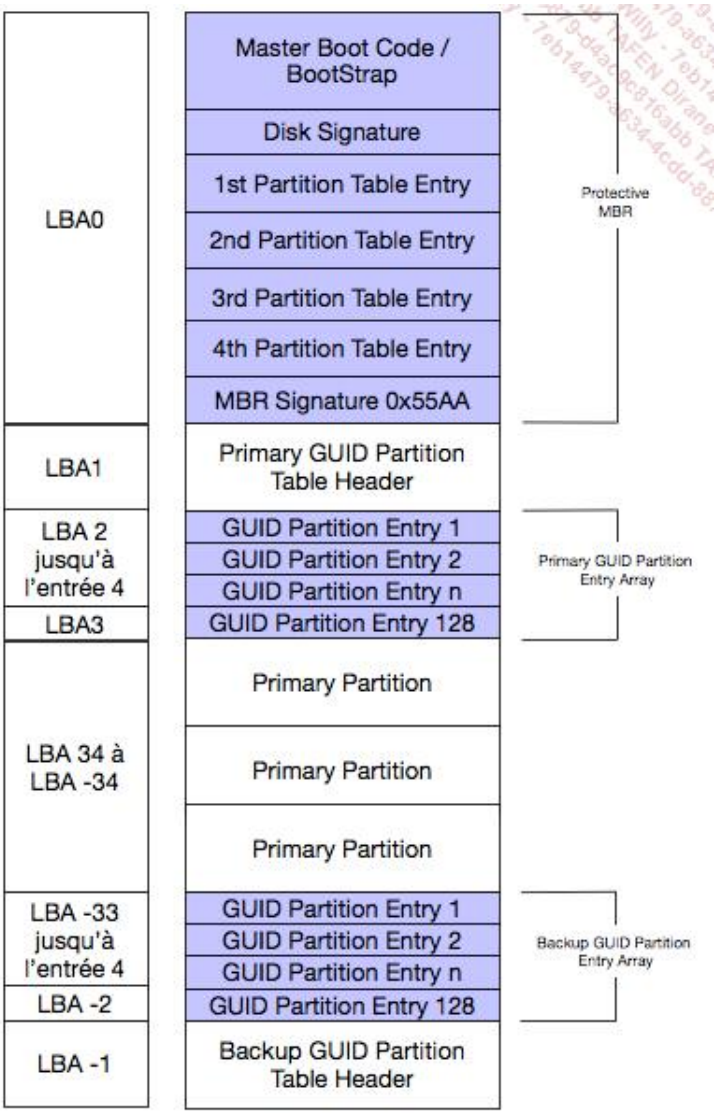
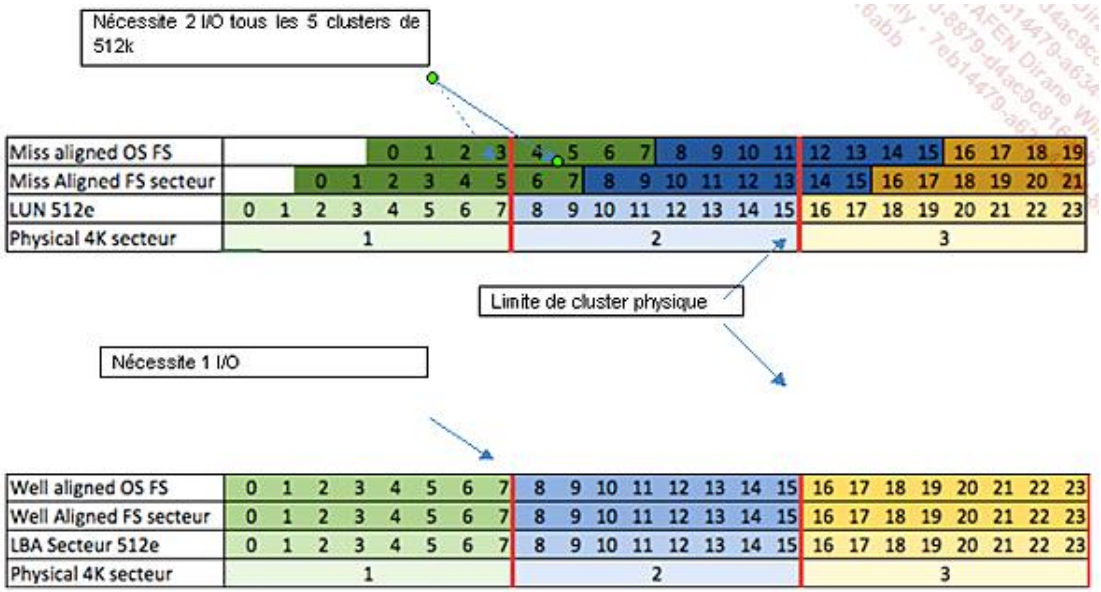


Schéma du GPT

c. De l'importance de l'alignement

Cette notion de cluster ou de bloc est importante car si pour un disque dur simple, le non-alignement de la partition système avec la structure physique (géométrie ou LBA) à un impact limité, dans une infrastructure hébergeant des machines virtuelles, il y a plusieurs niveaux entre les blocs physiques accédés par le LBA, et le bloc vu par le système d'exploitation. Ceci peut générer des lectures et écritures (I/O) inutiles.

Dans le cas d'une infrastructure VMware, il est nécessaire d'aligner la partition VMFS (https://kb.netapp.com/support/s/article/how-to-align-blocks-in-vmware-esx?language=en_US) sur le bloc physique (FS secteur sur les schémas) et ensuite d'aligner la structure du système d'exploitation sur cette même structure physique (OS FS sur les schémas).



Le fait de créer les datastores avec l'interface graphique de vSphere permet d'obtenir automatiquement une partition VMFS alignée avec le disque logique présenté par la baie de stockage.

2. Le SSD

Le SSD est un composant de plus en plus répandu dans le quotidien, aussi bien dans les baies de stockage que dans les serveurs et dans nos postes de travail.

Avant tout, que signifie SSD ?

C'est une unité de stockage utilisant des semi-conducteurs statiques, par opposition aux disques durs qui stockent les données sur un média en rotation (oui, le disque dans disque dur). On peut le qualifier de stockage électronique, et c'est principalement pour cette raison qu'on les oppose aux disques (durs) mécaniques. Une autre raison est que le mode de fonctionnement n'a pas grand-chose à voir avec les unités de stockage traditionnelles.

Comme pour la plupart des notions en informatique, nous avons affaire à un acronyme. Ainsi SSD est utilisé pour désigner un « Solid State Drive ».

Un SSD est un « drive » le D ne signifie en aucun cas « Disk ». On parle de puces électroniques, il n'y a pas de disque !

Les SSD surpassent les disques durs en latence d'accès, en IOPS et surtout en débit séquentiel mais surtout en accès aléatoire.

a. Les différents types de cellules

Les SSD sont constitués de différents types de cellules (puces électroniques utilisées pour stocker les données appelées NAND). La NAND est un type de mémoire flash apparu en 1989 créée par Toshiba.

Chaque puce contient au moins un bit, et ce bit peut avoir deux états (1 ou 0, c'est quelque chose de connu en informatique). L'état de chaque bit est modifié grâce à l'application d'une charge électrique.

Cependant les charges appliquées de manière répétitive dégradent le support, c'est pour cela qu'une cellule a un nombre limité de changements avant que l'application d'une charge électrique n'ait plus d'effet sur la cellule. Chaque cellule a donc une durée de vie limitée à un nombre de cycles d'écriture (oui car c'est le changement d'état

qui fait diminuer la vie de la cellule, la lecture n'a pas d'incidence).

De manière simple on peut les séparer en trois catégories :

- SLC ou *single level cell* ayant une endurance de plus de 100000 cycles
- MLC ou *multi level cell* ayant une endurance d'environ 10000 cycles
- TLC ou *triple level cell* ayant une endurance d'environ 3000 cycles

On trouve aussi un type de mémoire optimisé : e-MLC ou enterprise MLC ayant une endurance comprise entre 20 000 et 30 000 cycles.

Une mémoire SLC comporte un bit par cellule, une mémoire MLC ou DLC pour Dual Level Cell comporte deux bits par cellule et une mémoire TLC en comporte trois.

Les mémoires TLC sont les moins chères à produire mais du fait de leur endurance réduite, elles ne sont pas adaptées au milieu de l'entreprise (utilisation dans les serveurs ou baies de stockage, que ce soit en stockage capacitif ou en utilisation pour des caches lecture / écriture).

Un des plus gros avantages des SSD au-delà des débits sans commune mesure avec les disques durs est le temps de latence de l'ordre de 0,1ms quand un disque dur est à plus de 4ms pour les meilleurs (disques durs performants pour l'entreprise avec une vitesse de rotation de 10 000 tours par minutes et une interface de connexion en SAS/Near Line - SAS).

b. Les différents formats de connexion

Les SSD utilisaient les mêmes interfaces que les disques durs (à partir du SATA 3Gbps généralement).

Le SATA 6Gbps, qui représentait la dernière évolution de l'interface, est saturé depuis au moins deux ans par les SSD, même grand public ! Inutile de préciser que pour une configuration en RAID (0) l'interface représente un goulet d'étranglement.

L'interface mSATA n'est ni plus ni moins que du « mini » SATA adapté à la taille réduite des cartes SSD et pensé au départ pour les ordinateurs ultraportables.

L'interface m2, plus récente, présente un avantage : m2 est un connecteur uniquement et peut être associé au SATA ou au PCI-express. Ainsi pour du m2/PCI-Express, les limitations de débit du SATA sont de l'histoire ancienne.

Mais l'évolution des SSD étant rapide, on a vu arriver des SSD sur plusieurs lignes PCI-Express ce qui a permis une augmentation très importante des performances.

Très simplement, les SSD étaient considérés par les cartes mères comme des « disques durs très rapides » c'est pourquoi la plupart étaient gérés comme tels avec le protocole AHCI.

Aujourd'hui, en entreprise et pour les SSD les plus performants / récents, le remplaçant du protocole AHCI est arrivé : il s'agit de NVMe ou Non Volatile Memory Express. Ce protocole permet l'exploitation de toute la performance des SSD sur les points suivants : latences extrêmement faibles (pour certains SSD de l'ordre de quelques microsecondes) et gestion de plus d'entrées sorties par seconde - IOPS.

Attention : les SSD NVMe requièrent un pilote dont ne disposent nativement que les systèmes d'exploitation les plus modernes (fonctionnant avec des machines disposant d'EFI en lieu et place de l'antique BIOS). Intel fournit des pilotes pour les systèmes Windows 2008R2 par exemple.

c. Les optimisations

Le fait que les SSD soient basés sur des composants ayant une durée de vie limitée impose certaines optimisations afin d'augmenter la durée de vie et de maintenir les performances de l'unité de stockage.

Overprovisionning (surallocation)

Quand vous achetez un SSD avec une capacité indiquée, il n'est pas rare que le fabricant prévoie une quantité supplémentaire d'environ 6 à 10 % permettant de pallier un éventuel problème sur certaines cellules. Ainsi le fabricant prévoit la dégradation prématurée d'un nombre de cellules, ce qui permet de maintenir la capacité nominale car ces cellules sont masquées par le contrôleur et utilisables qu'en cas de défaut de certaines cellules.

Wear levelling

Le wear levelling est une fonctionnalité intégrée dans les contrôleurs de SSD. Il s'agit d'effectuer une répartition de charge en utilisant au mieux les cellules. Le but est d'éviter que certaines cellules arrivent en fin de vie alors que certaines autres sont encore fonctionnelles. Les cellules sont donc utilisées de manière uniforme jusqu'à la fin de vie du SSD (fin de vie de la plupart des cellules dans le même temps).

Un SSD en fin de vie de supporte plus l'écriture.

RAIN

Redundant Array of Independent NAND est une optimisation des SSD permettant d'écrire les données à plusieurs endroits et sur plusieurs composants à l'intérieur du SSD. Le but est d'éviter la perte de données due à une défaillance (par exemple sur un ensemble de cellules). La répartition et l'écriture simultanée sur plusieurs zones permet aussi d'augmenter les performances : http://www.crucial.com/usa/en/support-rain-technology

Garbage collector

Contrairement au disque dur, la structure physique et la structure logique du SSD ne sont pas identiques, et la structure physique n'est pas exposée au logiciel (l'OS). Une cellule de SSD ne peut pas être écrite deux fois de suite. Avant la deuxième écriture sur une même cellule, il faut l'effacer (rendre la cellule vierge, ce qui permettra une autre écriture), c'est pour cela qu'on parle de cycle pour une cellule NAND.

Quand un SSD est neuf, aucun problème : toutes les cellules seront au moins utilisées une fois. Tant que cet état n'est pas atteint, les performances du SSD sont natives (très bonnes généralement). Par contre, une fois chaque cellule utilisée une fois, on comprend aisément que les performances se dégradent très vite (une latence énorme se fait sentir car il faut « nettoyer » les cellules avant de les réutiliser). Les cellules sont divisées en blocs et les blocs en pages. On peut écrire directement une page mais l'effacement se fait par bloc.

Le garbage collector a pour tâche principale de réorganiser les pages afin de pouvoir supprimer les données qui ne sont plus valides (par blocs).

En général le wear levelling se fait pendant l'action du garbage collector. Avec les déplacements de données, le garbage collector génère des cycles d'utilisation des blocs mémoire (oui maintenir les performances du SSD le dégrade).

Trim

La commande trim est dépendante de l'OS. Elle permet à l'OS d'indiquer au SSD que telle ou telle page n'est plus valide car la donnée ne l'est plus non plus. Cela permet d'éviter des déplacements/copies de données car le

garbage collector évite de déplacer les données non valides.

Un garbage collector combiné à un OS gérant la commande trim permet de maintenir les performances en écriture d'un SSD.

Plus d'informations sur ces processus ici : http://www.thessdreview.com/daily-news/latest-buzz/garbage-collection-and-trim-in-ssds-explained-an-ssd-primer/