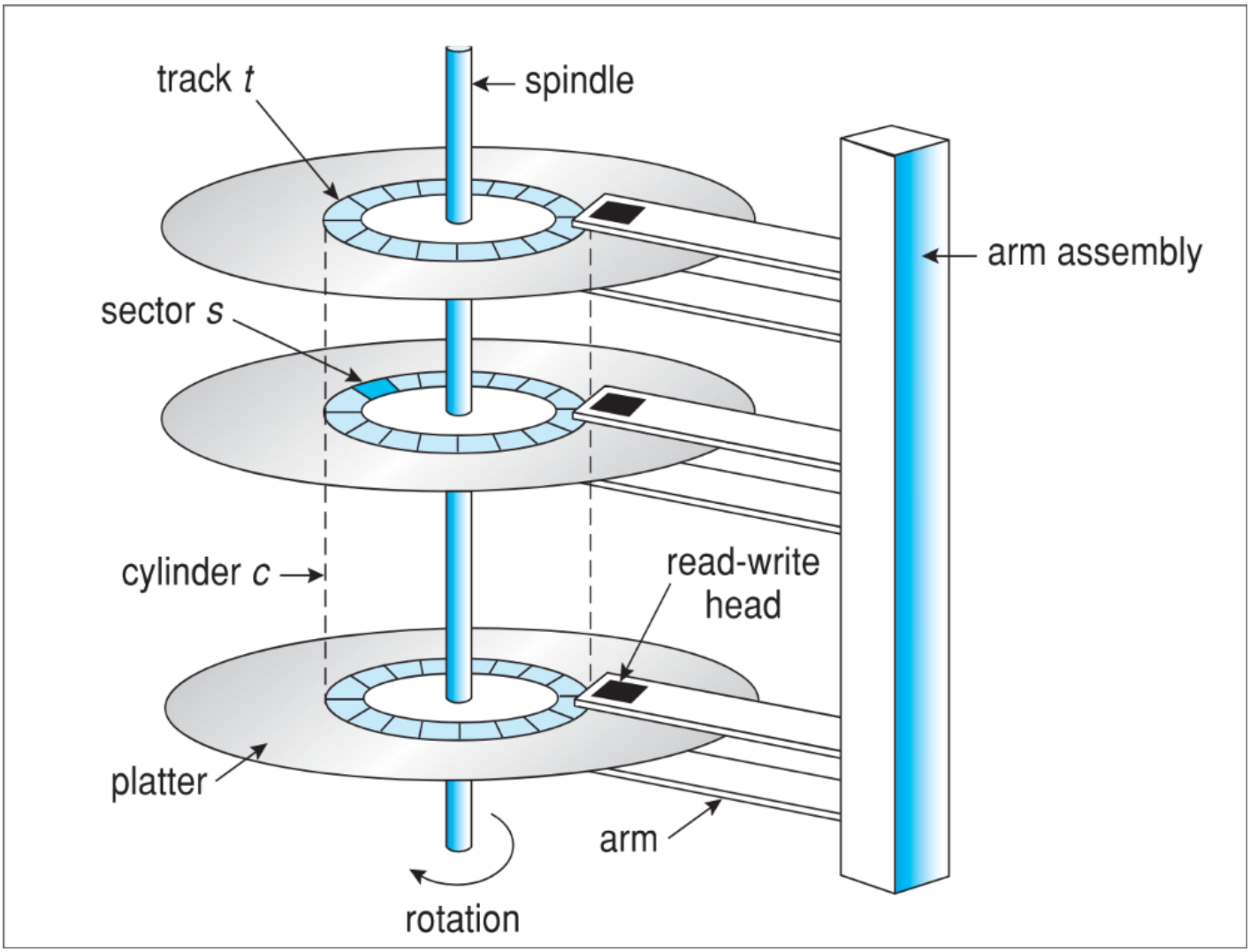
Disque

Matériel

Un disque magnétique est composé de plusieurs disques physiques appelés les **plateaux**.

Le disque est découpé en cylindres, pistes et secteurs. Il y a autant de pistes que de positions différentes de la tête de lecture.



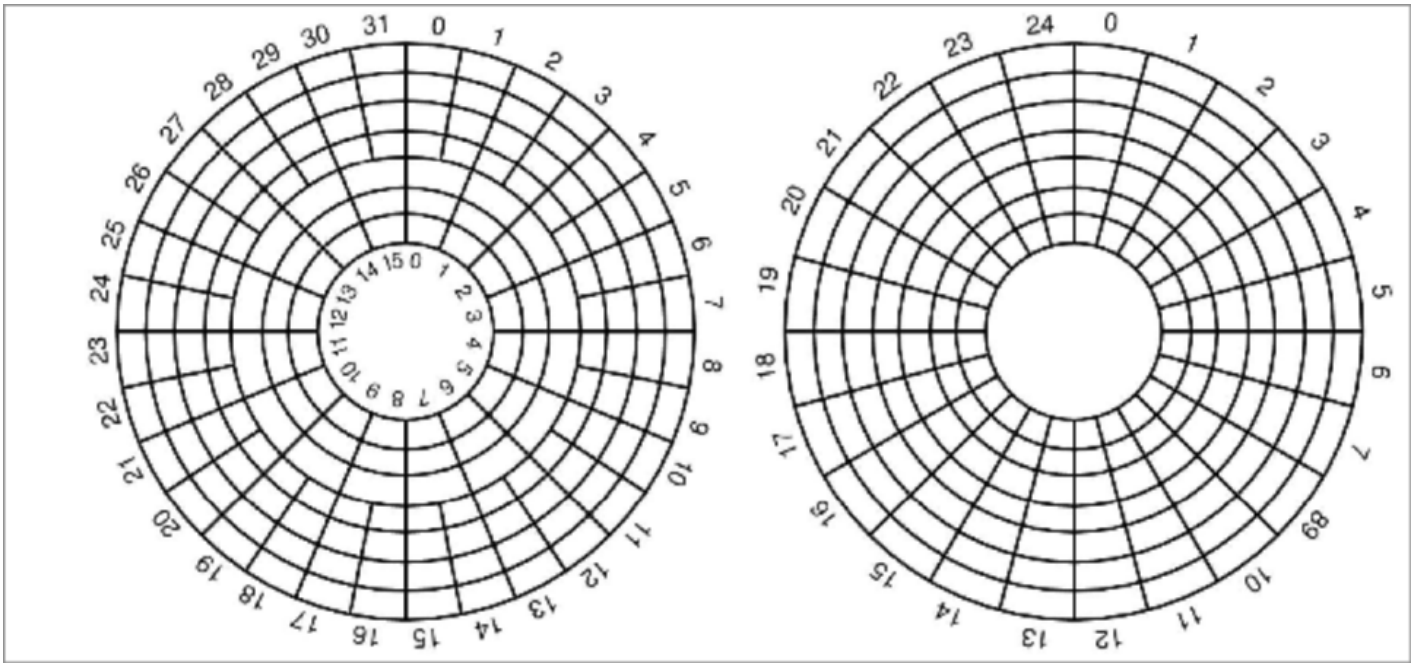
Un **cylindre**, c'est l'ensemble de pistes à une position donnée de la tête de lecture.

Une **piste** correspond à un cercle sur un plateau à une certaine position de la tête de lecture. Chaque piste est elle-même découpée en secteur.

Les **secteurs** sont des blocs de tailles fixes qui compose chaque piste.

> Pour en savoir plus sur le fonctionnement des disques dur magnétiques, vous pouvez consulter cette page Wikibooks.

Les disques dur ont une interface qui permet au contrôleur d'utiliser des commandes de haut niveau. La géométrie du disque dur n'est pas nécessairement celle qui est annoncée, car le nombre de secteurs par piste n'est pas constant.



A gauche, véritable géométrie du disque. À droite, géométrie du disque simplifie pour rendre les accès plus simple

RAID

RAID est une technologie qui consiste à combiner plusieurs disques afin d'améliorer les performances (si on a plusieurs disques, on peut par exemple écrire sur plusieurs disques en même temps) et/ou la fiabilité (si on a plusieurs disques, on peut par exemple dupliquer les informations sur tous les disques, comme ça si un disque tombe en panne, on peut restaurer les données).

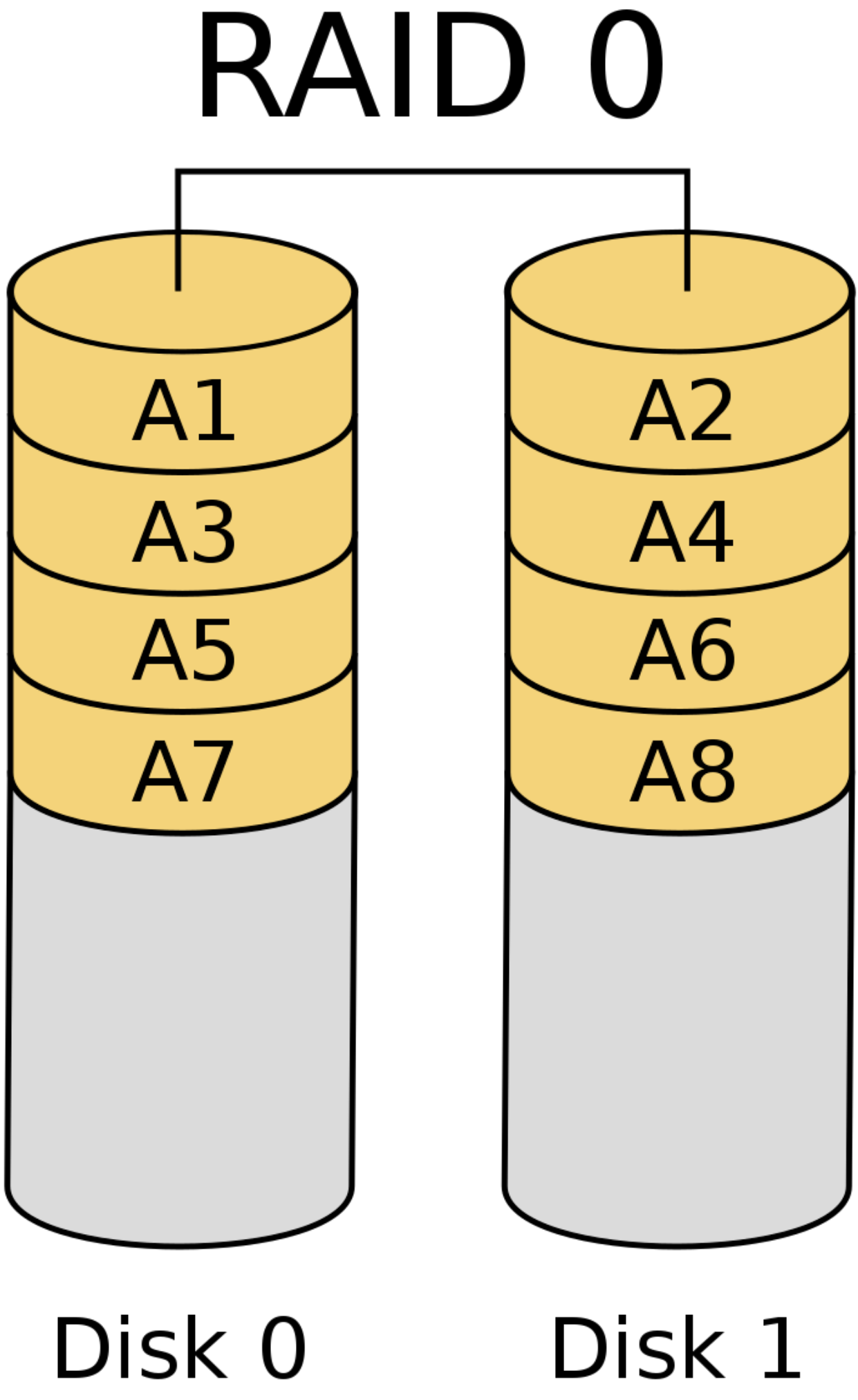
Le **mirroring** consiste à dupliquer toutes les informations sur un second. Ainsi, lors d'un accès en écriture, le contrôleur effectue la modification sur les deux disques simultanément et indique qu'il a terminé lorsque l'opération est achevée sur les deux disques.

Le **stripping** consiste à répartir l'information sur plusieurs disques. De cette manière, si on a huit disques, on peut distribuer les informations sur chaque disque de façon à écrire huit fois plus rapidement qu'avec un seul.

Il existe plusieurs niveaux de RAID, certains utilisent du mirroring, d'autre du stripping et d'autres les deux de manière à favoriser les performances et la fiabilité.

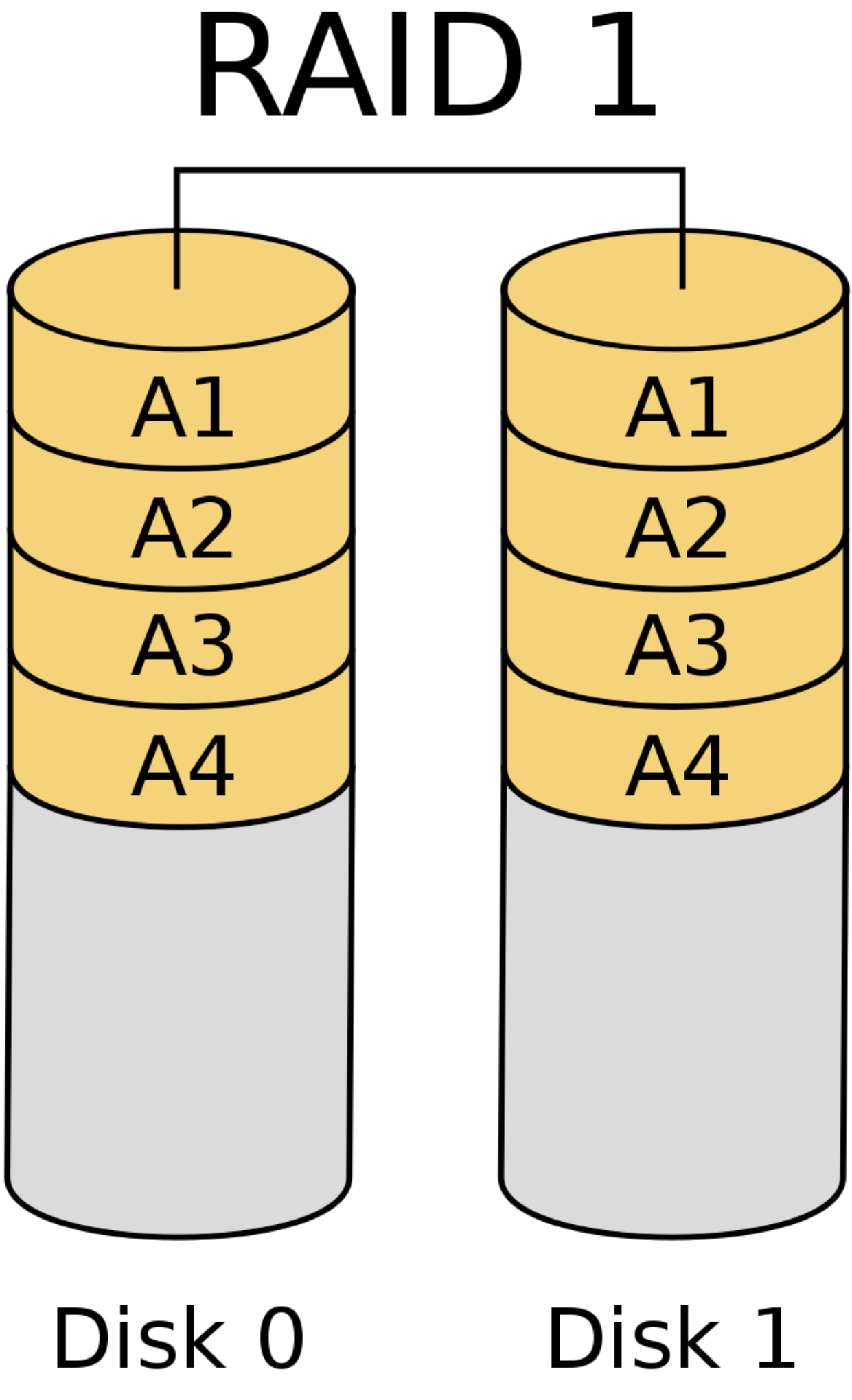
> Si vous souhaitez en savoir plus sur les niveaux de RAID, vous pouvez consulter cet article Wikipédia et celui-ci.

RAID 0


RAID 0
A1
A3
A5
A7
A2
A4
A6
A8
Disk 0
Disk 1

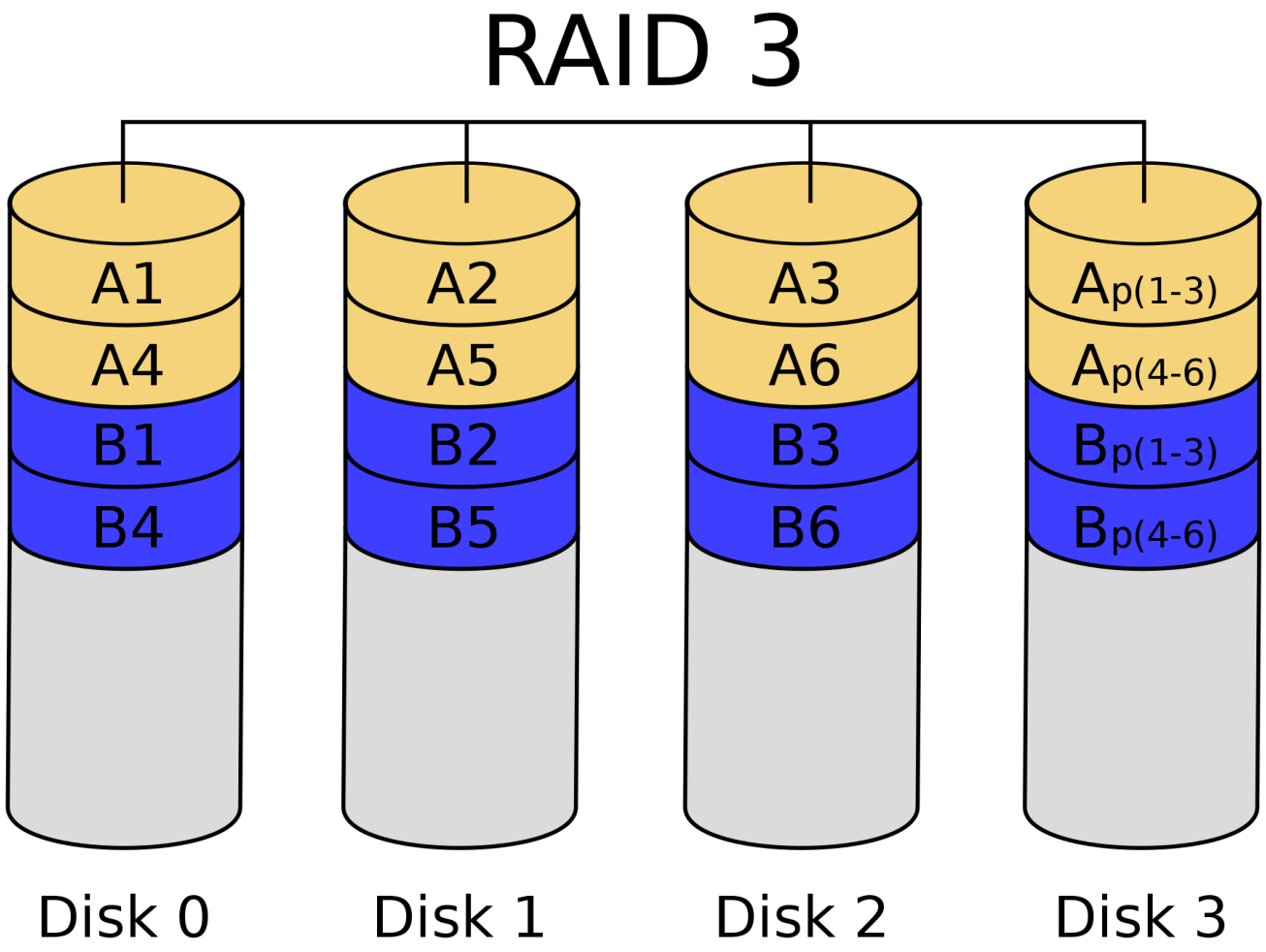
Le RAID 0 fait uniquement du stripping. Il va donc construire un grand espace disque en combinant les disques. Ainsi, les données sont réparties entre des différents disques de manière à améliorer les performances.

RAID 1


RAID 1
A1
A2
A3
A4
A1
A2
A3
A4
Disk 0
Disk 1

Le RAID 1 fait exclusivement du mirroring, il faut donc doubler le nombre de disques. Cela améliore beaucoup la fiabilité de l'information, mais il est assez couteux.

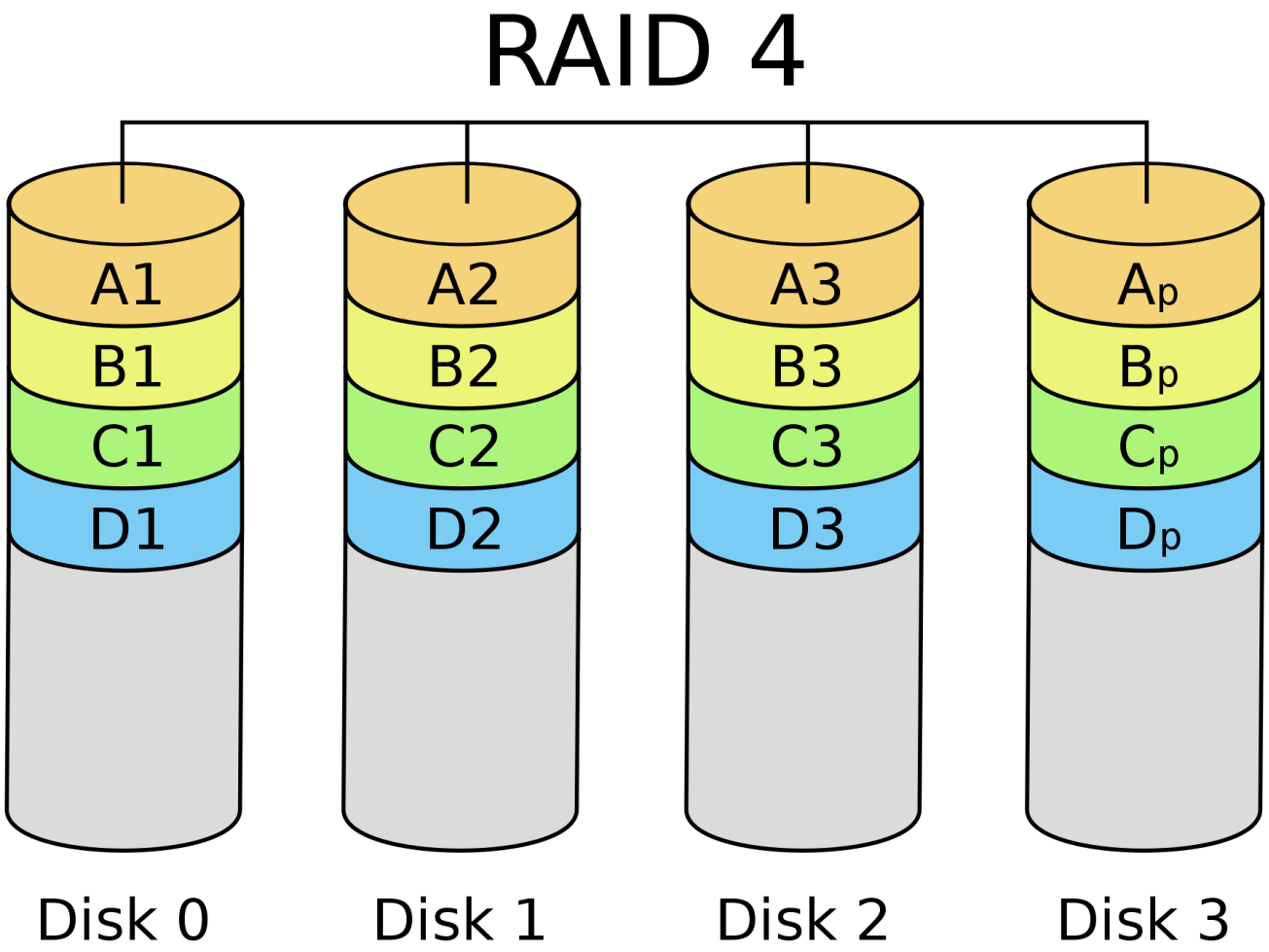
RAID 3



Le RAID 3 fait du stripping au niveau des octets. Un bit de parité est gardé sur un disque séparé et les octets vont être réparti sur les différents disques.

L'avantage du RAID 3 est que si on perd un disque, on peut reconstruire l'information à la volée en utilisant le disque de parité.

RAID 4

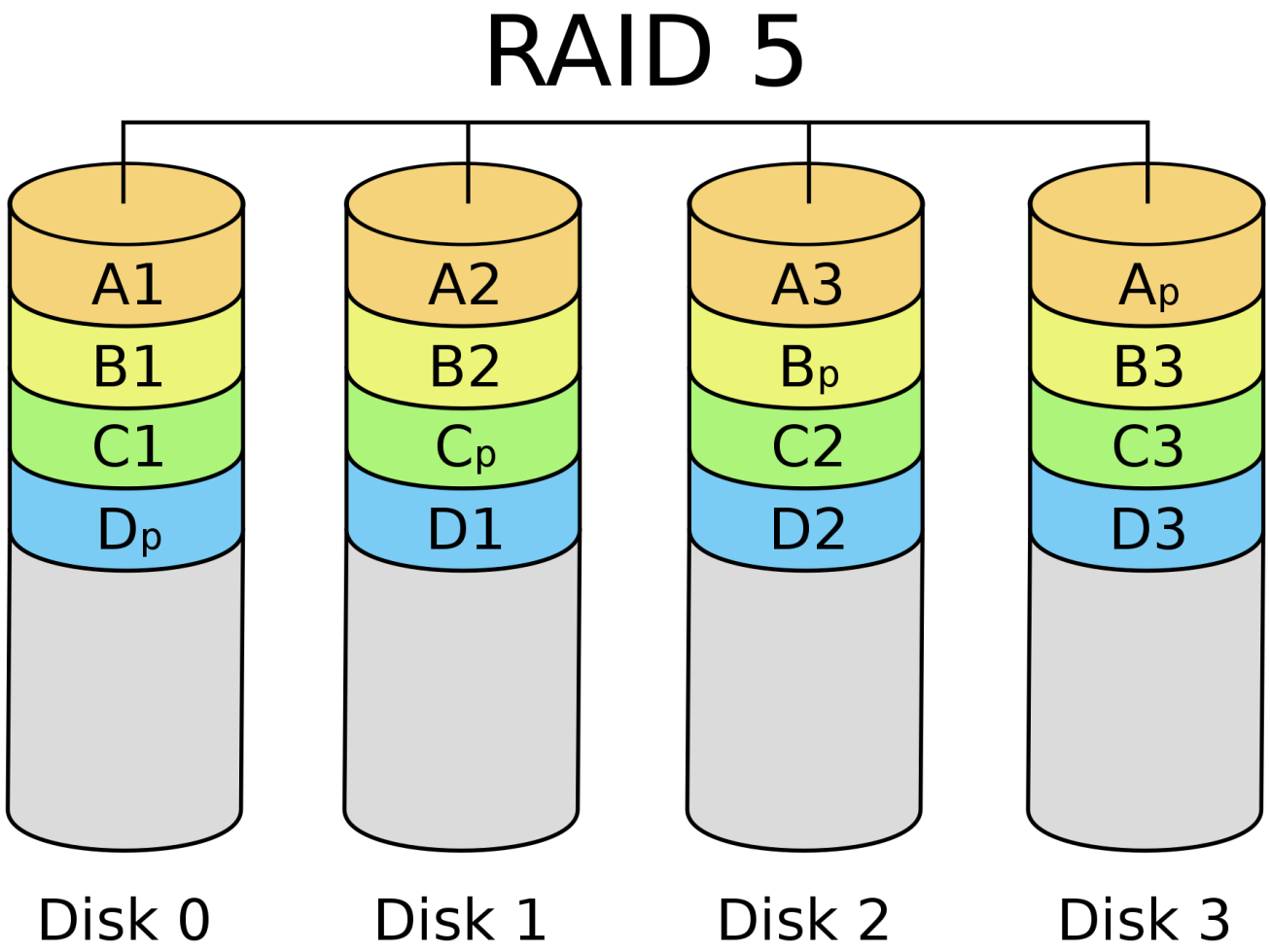


Le RAID 4 fait du stripping au niveau des blocs. Les blocs de parités sont gardés sur un disque séparé.

Un avantage du RAID 4 est que la lecture d'un bloc ne nécessite l'accès qu'à un seul disque (contrairement au RAID 3), on peut donc également satisfaire la lecture de plusieurs blocs simultanément si ceux-ci ne sont pas localisés sur le même disque.

Il y a cependant un problème à l'écriture, étant donné que tous les bits de parité sont sur le disque de parité, on ne peut pas écrire des blocs en parallèle, car on ne peut avoir qu'un seul accès au disque de parité à la fois.

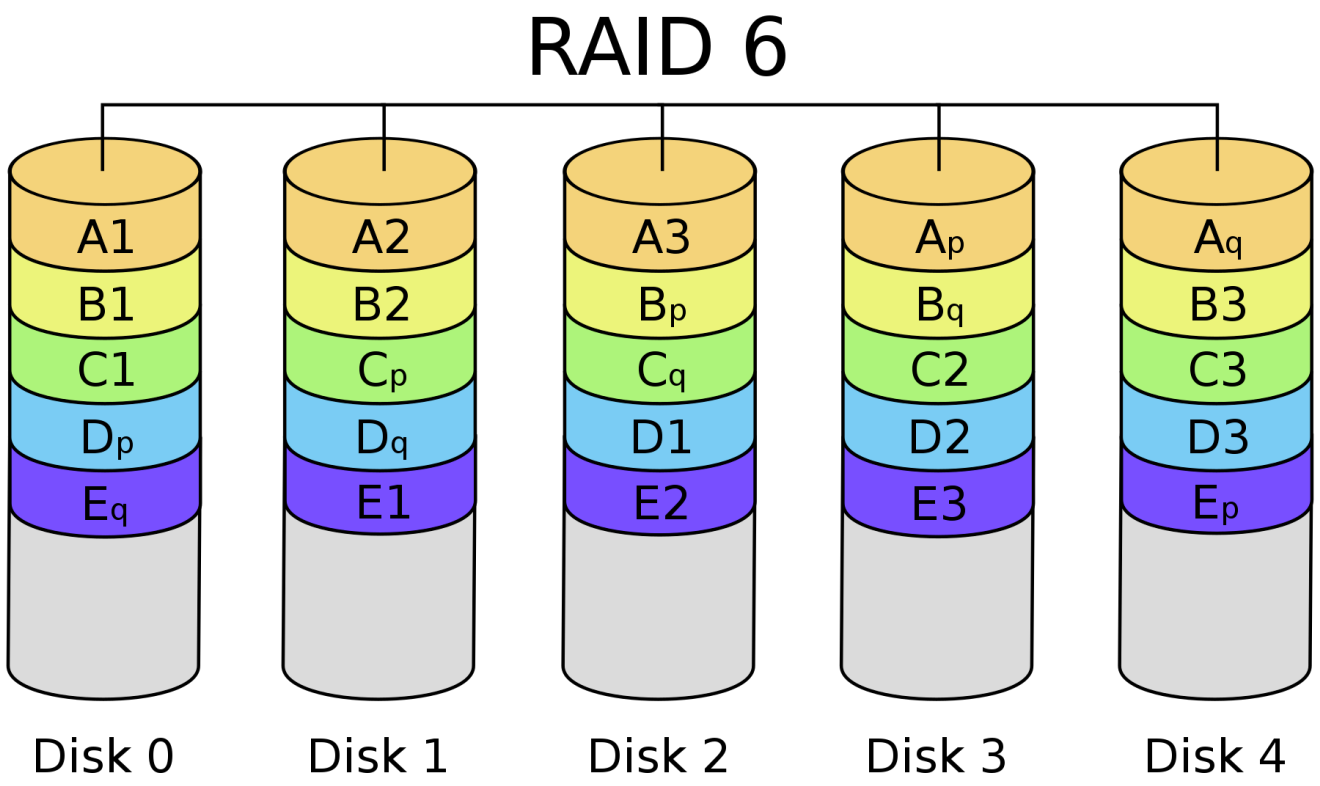
RAID 5



Le RAID 5 est une amélioration du RAID 4 qui va distribuer les informations sur la parité, de cette manière chaque disque contient des blocs de données et de parité. À chaque bloc de donnée correspond un bloc de parité stocké sur un disque.

Lors d'une écriture, il y a alors moins de problème, car plusieurs requêtes d'écriture peuvent être faites en même temps parce que tout dépend de la position de la donnée à écrire et de la localisation de l'information de parité.

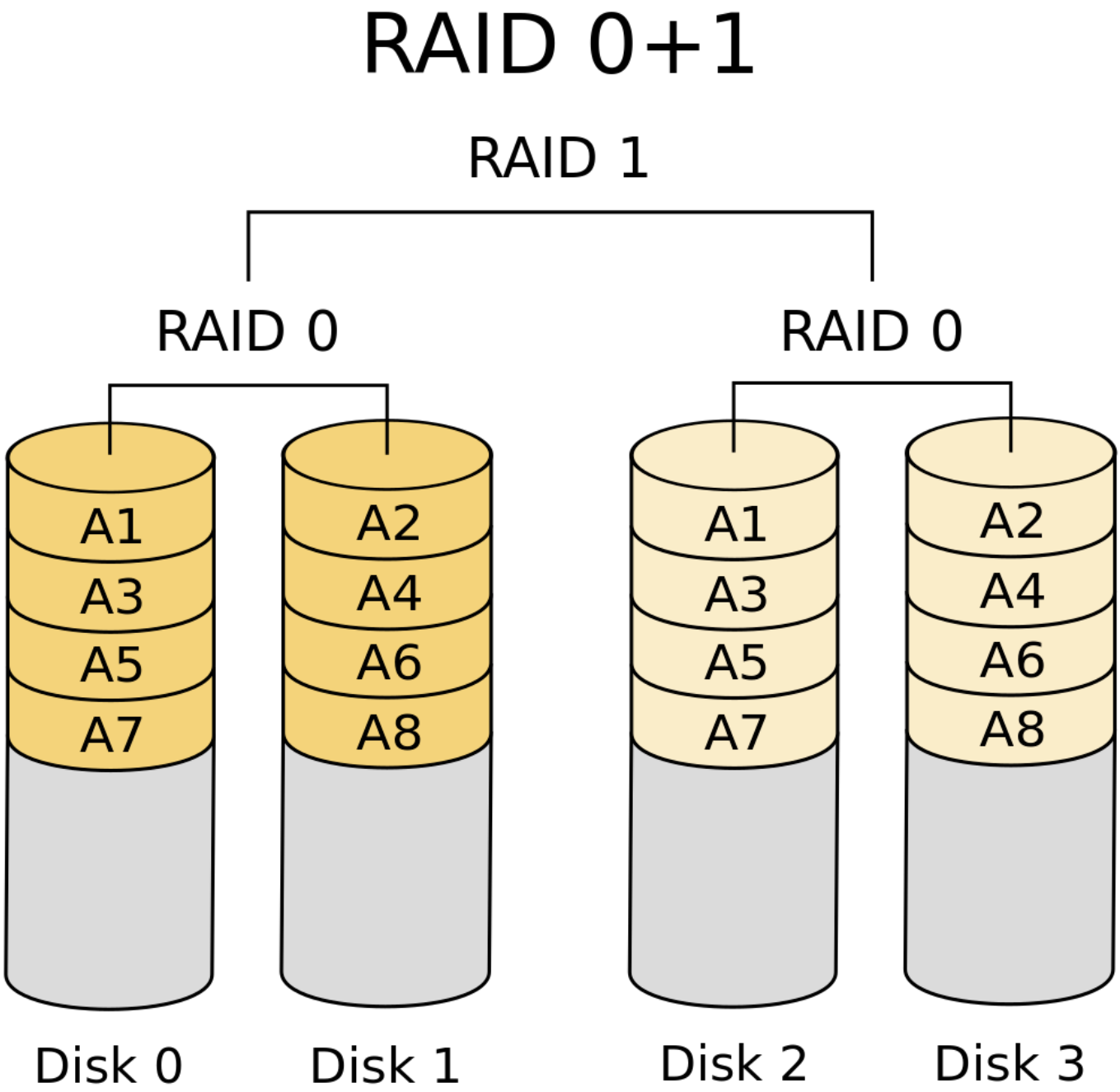
RAID 6



Le RAID 6 permet de protéger de la perte de deux disques dur, elle utilise un algorithme de parité plus complexe et nécessite un CPU plus important pour le contrôleur.

Plus il y a de données, plus le RAID 6 est préférable au RAID 5.

RAID 0+1



Ce niveau utilise RAID 0 pour le stripping et RAID 1 pour le mirroring de tous les disques en stripping. Il est très utile si l'efficacité et la protection sont importantes, cependant il est assez cher.

Lequel choisir ?

Généralement, c'est le RAID 5 qui est choisi s'il y a plusieurs disques ou alors le RAID 1 s'il n'y a que deux disques. Sauf dans des cas où il y a des demandes plus particulières au niveau de la performance et/ou de la fiabilité du système.

Options particulières

Il y a des options particulières sur les différents systèmes, notamment la possibilité de faire du **hot-swap** ou d'avoir des disques **spare**.

Le hot-swap consiste à pouvoir retirer des disques pendant que ceux-ci sont connectés (du moment qu'il n'est pas actuellement utilisé).

Le spare consiste à avoir un disque présent inutilisé, mais prêt à l'emploi. Ainsi, si un disque dur tombe en panne, le disque spare peut alors être utilisé.

Autres considérations

Il est important de bien considérer que le système ait un débit important pour avoir les meilleures performances.

Mais surtout, il est important d'avoir une certaine diversité entre les disques dur. Il ne faut donc pas prendre tous les disques dur de la même marque au même moment, de la même génération. Car alors, il y a de grandes chances que les disques se fatiguent de la même manièrent et tombe en panne plus ou moins en même temps.

Formatage

Avant qu'un disque ne puisse être utilisé, il doit être formaté (**formatage de bas niveau**), cette opération consiste à écrire la géométrie du disque (secteur, cylindre, piste, etc).

Cette opération doit être réalisée par le fabricant. De cette manière, chaque secteur contient un préambule (informations décrivant le secteur tel que le numéro ou le cylindre), les données et un code ECC pour la gestion des erreurs.

Une fois l'espace créé, il est possible de mettre en place des partitions, c'est la première structure logique du disque.

Le disque est séparé en plusieurs partitions, et chaque partition est vue par le système comme un espace séparé. Par exemple, un système Linux va voir les partitions `/dev/sda1` et `/dev/sda2` comme deux espaces complètements différents.

Le secteur zéro du disque contient la partition control block et le boot control block qui mentionnent comment le disque et découpé et comment le système peut démarrer.

Le **formatage de bas-niveau** est une opération de formatage réalisée par le système d'exploitation, elle consiste à donner une structure à la partition (un format de système de fichier).

Scheduling

Le disk arm scheduling est une opération faite par le système d'exploitation pour planifier les requêtes d'entrées-sorties.

En effet, plusieurs requêtes d'E/S peuvent arriver depuis plusieurs processus pendant que le disque est toujours en train de traiter une requête, donc les autres requêtes doivent attendre.

L'importance du scheduling (avec les HDD) est de minimiser le **seek-time**, c'est-à-dire minimiser les mouvements de la tête de lecture afin de pouvoir lire les informations plus vite.

Nous allons donc, voire plusieurs algorithmes de scheduling différents,

- Le **FCFS** (First Come First Served), soit les requêtes sont servies dans l'ordre dans lesquelles elles arrivent. Cet algorithme a l'avantage d'être équitable, mais a de mauvaises performances, car il ne fait rien pour améliorer le temps de réponse du disque.
- Le **SSTF** (Shortest Seek-Time First), autrement dit de servir toujours la requête la plus proche de la position courante. Cet algorithme a l'avantage de faire diminuer le temps de réponse du disque. Mais il a le désavantage de devoir calculer les positions et risque également de causer de la famine parmi les requêtes les plus éloignées.
- Le **SCAN** (ou algorithme de l'ascenseur), où le bras de lecture va toujours dans la même direction jusqu'à arriver à la fin du disque avant de retourner dans le sens inverse, aussi, il n'y a plus de famine, car quoi qu'il arrive, on avance.
- Le **C-SCAN** (SCAN circulaire), fonctionne comme le SCAN sauf qu'au lieu de partir dans l'autre sens lorsque la fin du disque est atteinte, elle retourne à l'autre extrémité du disque.
- Le **LOOK** est une variante du SCAN qui a la place de continuer jusqu'aux extrémités du disque, va s'arrêter lorsqu'il n'y a plus de requête dans la direction.
- Le **C-LOOK** (LOOK circuliare), comme le LOOK, excepté qu'à la place d'aller dans la direction inverse, il va directement à la première requête.

> Vous pouvez découvrir d'autres algorithmes ou avoir plus d'explication et d'illustrations sur les algorithmes décrit ici en allant lire cette page.

Choix de l'algorithme

Le SSTF est courant et fournit un bon remplacement à FCFS.

Si le disque est très chargé, C-SCAN et C-LOOK sont intéressants.

Le plus souvent on va trouver du SSTF ou une variante de LOOK.

Aujourd'hui, le scheduling est réalisé à plusieurs niveaux, par exemple au niveau du système d'exploitation, mais également au niveau des contrôleurs.

Gestion des erreurs

Il peut y avoir beaucoup d'erreurs différentes sur un disque.

Par exemple, des secteurs défectueux peuvent survenir lors de la construction du disque dur, notamment lors du formatage de bas niveau, ou même survenir durant l'utilisation du disque.

C'est pourquoi le système d'exploitation doit pouvoir se souvenir des blocs défectueux afin de ne plus les utiliser dans le futur.

Revision #2
Created 5 January 2024 13:48:53 by SnowCode
Updated 6 January 2024 18:13:15 by SnowCode