

Hardware

Notions théoriques liées au hardware.

- [Hardware - Le binaire](#)
- [Hardware - Le RAID](#)
- [Hardware - Un ordinateur](#)

Hardware - Le binaire



Difficulté : Débutant

Notions : Binaire, base 2

I. Introduction

“ Il y a 10 types de personnes dans le monde : celles qui savent compter en binaire et les autres.

- *Blague de geek*

Le langage binaire est le langage informatique le plus **bas niveau**.

En effet un ordinateur fonctionnant avec du courant électrique, il n'y a que deux états possible pour un bit à l'instant T.

Circuit fermé, présence de courant : **1**

Circuit ouvert, absence de courant : **0**

De par ses deux valeurs possible, le binaire est donc un système de calcul en **base 2**.

II. Les bases de calcul

Information : Quel que soit le calcul, il est à noter que la première valeur possible est toujours 0 (Zero).

2.1 Base 10

La base 10 est la base naturelle pour l'humain et celle couramment utilisée dans le monde.

De par le fait que les mains humaines disposent de 10 doigts, c'est en effet plus pratique et instinctif pour compter.

on a donc les valeur suivantes possibles :

Valeurs base 10
0 = 00
1 = 01
2 = 02
3 = 03
4 = 04
5 = 05
6 = 06
7 = 07
8 = 08

$$9 = 09$$

Une fois la valeur maximum atteinte (09) pour incrémenter au delà, on ajoute une dizaine devant l'unité et on continue ainsi à incrémenter.

10, 20, 30 etc... etc...

2.1 Base 2

En binaire le principe reste le même qu'en base 10 , on à donc 0, 1 et quand la valeur maximum est atteinte, on incrémente.

Valeurs base 2

$$0 = 0$$

$$1 = 1$$

$$2 = 10$$

$$3 = 11$$

$$4 = 100$$

$$5 = 101$$

$$6 = 110$$

$$7 = 111$$

$$8 = 1000$$

$$9 = 1001$$

etc...

Si de prime abord, cela peut ne pas sembler naturel, il s'agit juste de reproduire la même logique de comptage que la base 10, mais avec seulement 2 valeurs disponible.

III. l'Octet

3.1 Construction

Heureusement, pour rendre tout cela plus lisible et car un bit seul ne fait pas sens dès lors qu'il s'agit de compter au delà de 1, on va regrouper ces bits par paquets. Ces packets sont appelés des **registres**.

Le registre le plus connu et celui qui sera couramment utilisé est l'**octet** (un paquet de 8 bits).

Basé sur ce principe on aura donc la représentation suivante : **00000000**

Et là encore, si cela paraît compliqué, il existe un moyen simple et mnémotechnique de se représenter facilement les valeurs en binaire.

En effet, le binaire à une particularité, pour chaque bit que l'on rajoute devant, on double les possibilité, ce qui fait que chaque bit a une valeur différente. Cela s'appelle un **Poid** de bit.

Pour se donner une bonne idée, voici la valeur de chaque bit dans un octet :

128	64	32	16	8	4	2	1
0	0	0	0	0	0	0	0

3.2 Conversion base 10 / Base 2

Dès lors, pour convertir des nombres binaires en base 10 et de base 10 en binaire, cela deviens facile.

Il suffit d'additionner les valeurs de tous les bits égaux à 1. Ainsi,

01101011 par exemple se note :

128	64	32	16	8	4	2	1
0	1	1	0	1	0	1	1

soit : $64 + 32 + 8 + 2 + 1 = 107$.

De la même manière, pour convertir un nombre en base 10 en binaire, il faudra additionner en partant de la droite les valeurs pour arriver à la somme du nombre que l'on veut et passer leurs valeurs à 1.

218 par exemple se note :

Détail de l'opération

128 < 218, donc on part avec 128 = 1.

128	64	32	16	8	4	2	1
1	0	0	0	0	0	0	0

128 + 64 = 192.

192 < 218, le bit avec une valeur de 64 est donc à 1.

128	64	32	16	8	4	2	1
1	1	0	0	0	0	0	0

192+32 = 224.

224 > 218 donc le bit avec une valeur de 32 est à 0. On passe à la valeur suivante.

128	64	32	16	8	4	2	1
1	1	0	0	0	0	0	0

$192 + 16 = 208$.

$208 < 218$, donc le bit avec une valeur de 16 est à 1.

128	64	32	16	8	4	2	1
1	1	0	1	0	0	0	0

$208 + 8 = 216$.

$216 < 218$, donc le bit avec une valeur de 8 est à 1.

128	64	32	16	8	4	2	1
1	1	0	1	1	0	0	0

$216 + 4 = 220$.

$220 > 218$, donc le bit avec une valeur de 4 est à 0.

128	64	32	16	8	4	2	1
1	1	0	1	1	0	0	0

$216 + 2 = 218$.

$218 = 218$. On passe donc notre dernier bit (celui dont la valeur est 2) à 1. La valeur des bits restant à droite sera donc 0.

128	64	32	16	8	4	2	1
1	1	0	1	1	0	1	0

128	64	32	16	8	4	2	1
1	1	0	1	1	0	1	0

Note : Si l'on additionne toutes les valeurs, le résultat est de 255. Par conséquent, si l'on doit traiter des nombres supérieurs à 255, on ajoutera simplement un octet devant, et l'on poursuivra le tableau de la manière suivante.

32768	16384	8192	4096	2048	1024	512	256
-------	-------	------	------	------	------	-----	-----

octet 2

128	64	32	16	8	4	2	1
-----	----	----	----	---	---	---	---

octet 1

3.3 Les Échelles de tailles

Partant du principe qu'un octet peut stocker toutes sortes de données et que les bits sont donc l'unité de mesure la plus petite, si l'on veut stocker plus de données, on sera donc obligé d'accoler des octets, ou utiliser des registres plus grands.

Au bout d'un moment, il faudra que nos échelles de mesures continuent d'avoir un sens.

Si on empile beaucoup d'octets, on ne va pas continuer à dire, la taille de ce bloc est de 1 228 283 219 233 274 272 octets.

Pour cela, sur le même principe que les tableau de conversion, on va utiliser d'autre échelles.

A l'instar des unités de mesures classiques (gramme, kilogramme) (centimètre, mètre, kilomètre) il va falloir établir un tableau de conversion pour faire la mise à l'échelle.

Les unités sont donc les suivante :

- **bit** (unité)
- **octet** (8 bits)
- **KiloOctet** ou **Ko** (1024 octets)
- **MegaOctet** ou **Mo** (1024 Ko)
- **GigaOctet** ou **Go** (1024 Mo)
- **TéraOctet** ou **To** (1024 Go)
- **PétaOctet** ou **Po** (1024 To)

Note : A l'américaine, l'octet se dit **Byte**. Un équivalent est donc, le GigaByte ou Mega byte, etc... Notés : Kb, Mb, Gb, etc...

Trivia : au dessus du péta, il y a l'Exa, le Zeta, le Yotta, Ronna, Quetta, ...

IV. Les opérations en binaire

4.1 Addition

Cela fonctionne plus ou moins comme une addition standard, avec les retenues.

Selon les principes suivants :

$$0 + 0 = 0$$

$$(0 + 1 \text{ ou } 1+0) = 1$$

$1+1 = 10$ (équivalent de 2) Donc dans ce cas, l'on pose 0 et on retient 1.

Par exemple : **00100101 + 10110010** (soit 37 + 178)

Détail de l'opération

rete nue								
No mbr e 1	0	0	1	0	0	1	0	1
No mbr e 2	1	0	1	1	0	0	1	0
Tot al								1

On commence par la droite. 1+0 = 1, pas de retenue.

rete nue								
No mbr e 1	0	0	1	0	0	1	0	1
No mbr e 2	1	0	1	1	0	0	1	0
Tot al				1	0	1	1	1

0 + 1 = 1, pas de retenue,
1 + 0 = 1, pas de retenue,
0 + 0 = 0, pas de retenue,
0 + 1 = 1, pas de retenue.

rete nue		1						
No mbr e 1	0	0	1	0	0	1	0	1
No mbr e 2	1	0	1	1	0	0	1	0
Tot al			0	1	0	1	1	1

1 + 1 = 10. je pose 0 et je retiens 1.

rete nue		1						
No mbr e 1	0	0	1	0	0	1	0	1
No mbr e 2	1	0	1	1	0	0	1	0
Tot al		1	0	1	0	1	1	1

1 + 0 = 1 puis 1 + 0 = 1. Pas de retenue

rete nue		1						
No mbr e 1	0	0	1	0	0	1	0	1
No mbr e 2	1	0	1	1	0	0	1	0
Tot al	1	1	0	1	0	1	1	1

enfin, $0 + 1 = 1$, pas de retenue.

1

Nombre 1	0	0	1	0	0	1	0	1
Nombre 2	1	0	1	1	0	0	1	0
TOTAL	1	1	0	1	0	1	1	1

on a donc **00100101 + 10110010 = 11010111** (soit $37 + 178 = 215$)

4.2 Soustraction

La soustraction est un peu plus complexe que l'addition, en effet il va falloir jongler sur les chiffres.

Selon les principes suivants :

$$0 - 0 = 0$$

$$1 - 1 = 0$$

$$1 - 0 = 1$$

$0 - 1 =$ Ne peux pas directement être résolu, nécessite un emprunt.

Par exemple : **10110010 - 00100101** (soit 178 - 37)

Détail de l'opération

rete nue							0	10
No mbr e 1	1	0	1	1	0	0	1	0
No mbr e 2	0	0	1	0	0	1	0	1
Tot al								1

On commence par la droite. 0-1, emprunt à gauche.
On barre le 1 à gauche on le remplace par un **0**. (car 1-1=0)
On ajoute 2 soit **10₂** en remplacement du 0 ce qui donne le calcul 2-1 = 1.

rete nue							0	10
No mbr e 1	1	0	1	1	0	0	1	0
No mbr e 2	0	0	1	0	0	1	0	1
Tot al							0	1

0 - 0 = 0 pas d'emprunt.

rete nue s								
				0	10		0	10
No mbr e 1	1	0	1	1	0	0	1	0
No mbr e 2	0	0	1	0	0	1	0	1
Tot al							0	1

0 - 1 non solvable sans emprunts.
Pas de 1 adjacent à gauche, on va donc devoir aller chercher le premier disponible pour l'emprunter.
on barre le 1 que l'on remplace par **0** (1-1=0). Puis on barre le 0 à sa droite pour remplacer par 2 soit **10₂**.

rete nue s					1	10		
				0	10		0	10
No mbr e 1	1	0	1	1	0	0	1	0
No mbr e 2	0	0	1	0	0	1	0	1
Tot al						1	0	1

On continue à décaler. On soustrait 1 de 2 (**10**₂), il reste **1**.

Enfin, on reporte le 1 à droite et on exécute le calcul.
2 - 1 = 1.

rete nue s					1	10		
				0	10		0	10
No mbr e 1	1	0	1	1	0	0	1	0
No mbr e 2	0	0	1	0	0	1	0	1
Tot al	1	0	0	0	1	1	0	1

On continue ainsi :

$$1 - 0 = 1$$

$$0 - 0 = 0$$

$$1 - 1 = 0$$

$$1 - 0 = 1$$

Nombre 1	1	0	1	1	0	0	1	0
Nombre 2	0	0	1	0	0	1	0	1
TOTAL	1	0	0	0	1	1	0	1

on a donc **10110010 - 00100101 = 10001101** (soit 178 - 37 = 141)

V. Les fonctions binaires

Les fonctions en binaire ne sont pas des opérations à proprement parler mais le résultat d'un test livré par ce que l'on appelle en informatique et en électronique une porte logique.

5.1 AND (et)

0	0	0
0	1	0
1	0	0
1	1	1

Le résultat de la condition est vrai uniquement si les deux termes sont vrais.

5.2 OR (ou)

0	0	0
0	1	1
1	0	1
1	1	1

Le résultat de la condition est vrai si au moins l'un des deux termes est vrai.

5.3 XOR (ou exclusif)

0	0	0
0	1	1
1	0	1
1	1	0

Le résultat de la condition est vrai seulement si l'un des deux terme est vrai.

Trivia : c'est ce système là qui va être utilisé pour calculé des bits de parité. Dans les matrices RAID par exemple.

Pour aller plus loin : [Wikipedia | Tables de vérité](#)

VI. Conclusion

Au sein de l'ordinateur, tout est binaire. De la donnée écrite sur les supports de stockage, jusqu'à l'information transitant par le réseau, en passant par les opérations processeurs et le contenu de la mémoire.

Nous manipulons juste de moins en moins ce type de contenu au fil de l'évolution des langages informatiques et de programmation.

Mais il est cependant important de comprendre (au moins en théorie) comment fonctionne l'informatique sur sa couche la plus basique.

Hardware - Le RAID



Difficulté : Intermédiaire

Notions : RAID, Stockage, volumes, sécurité matérielle.

I. Notions Importantes

Le RAID ou '**Redondant Array of Independent Disk**' consiste à prendre plusieurs supports de stockage indépendant (HDD, SSD, NVME) et les assembler en un volume virtuel dans le but d'améliorer la performance, la capacité et/ou la tolérance à la panne.

Le RAID ne remplace pas la sauvegarde !!!

- Le but du RAID est avant tout de fournir une extension de capacité de volume et/ou une redondance de disque. Ce qui rendra la machine tolérante à la panne d'un ou plusieurs disques. Mais il ne s'agit en aucun cas d'une sauvegarde et une donnée perdue au sein du volume l'est pour de bon.
- Sur les RAID apportant une tolérance à la panne, si un disque tombe en panne, celui-ci peut être remplacé. Le volume commencera alors automatiquement une reconstruction de volume.
- Durant une reconstruction, le système **n'est plus tolérant à la panne** jusqu'à la fin de la reconstruction. Toute nouvelle défaillance durant cette période entraînera la perte des données.

Il est fortement recommandé de prendre des disques de même performance et capacité.

- En effet, lorsque les disques seront agrégés ensemble dans un volume, l'ensemble des disques ajusteront leurs performances sur celui qui a la plus basse disponible. Cela s'explique par la nécessité d'utiliser simultanément plusieurs disques, ce qui ne peut se faire que si tous les disques ont la même vitesse de lecture/écriture. Sinon il y aurait un risque de désynchronisation.
- Concernant la capacité, il s'agit à peu près de la même raison. Si un disque est plein avant les autres, les données ne pourront théoriquement plus être écrites que sur les disques où il reste de l'espace libre. Or le but recherché ici est d'avoir une redondance des données ou d'agréger ensemble des volumes.

Pour ajouter un niveau de sécurité en plus, il est conseillé de prendre des disques provenant de lots différents.

- En effet, des disques provenant tous du même lot et du même constructeur auront statistiquement la même chance de tomber en panne. Or la plupart des types de RAID classiques ne tolèrent qu'une seule panne. Ainsi, choisir des disques provenant de lots différents et donc n'ayant pas été fabriqués au même moment réduira le risque d'avoir une panne simultanée de plusieurs disques, ce qui entraînera la perte des données du volume.

II. Raid matériel vs raid logiciel

2.1 Raid matériel

Ce type de RAID fait appel à une carte contrôleur matérielle dédiée. La carte est ajoutée au serveur et les disques sont branchés dessus.

Le principal avantage de cette méthode est que la carte dédiée dispose de son propre processeur, mémoire et stockage pour faire la mise en cache des données et les calculs de lecture ou écriture. Le volume RAID est directement présenté au système.

Celui-ci n'a donc pas à gérer les calculs liés à la répartition des blocs et toute la puissance de calcul de la machine peut être exploitée par le système.

Convient à des environnements de production.

Pours :

- Plus performant que le raid logiciel.
- Ne sollicite pas la puissance de calcul.
- Indépendant du système d'exploitation.

Contres :

- Plus coûteux (achat d'une carte dédiée).

2.2 Raid logiciel

Ce type de RAID est géré par le système. Cela le rend indépendant du Hardware mais l'absence de matériel dédié implique que les calculs doivent être gérés par le système. Utilisant donc le processeur, la mémoire et le cache des disques pour réaliser les calculs et impactant donc les performances systèmes.

Il faut donc prévoir plus de puissance que nécessaire pour compenser la baisse de performance.

Cependant, en cas de panne totale ou changement du matériel, l'ensemble du matériel peut être remplacé sans problème. Ce qui n'est pas forcément toujours le cas sur du RAID matériel (si la carte RAID lâche par exemple).

Plus adapté à un usage sur les postes de travail ou un usage domestique.

Pours :

- Moins coûteux.
- indépendant du matériel.

Contres :

- impacte les performances.

2.3 Raid Pseudo-matériel

C'est un hybride des deux modes ci-dessus. Un contrôleur RAID est intégré à la carte mère et le système pourra s'appuyer dessus pour contrôler le RAID. Cependant, celui-ci reste géré logiciellement .

L'avantage de ce mode est surtout pour les PC de bureau. Beaucoup de cartes mères haut de gamme disposent de cette option et cela permet de mettre en place un RAID a moindre coût.

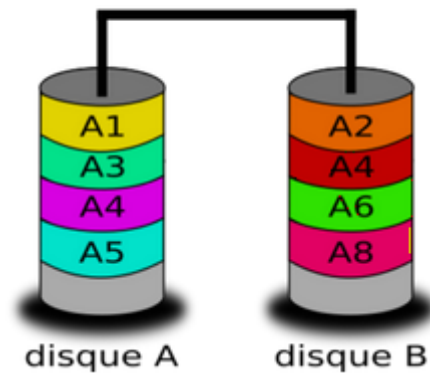
III. Principaux types de raid

3.1 Raid 0 - Striping

Prérequis : 2 disques ou plus.

Le RAID 0 ou '**striping**' est le niveau le plus basique de RAID.

Il s'agit tout simplement de prendre tous des disques participant et de les mettre ensemble pour présenter un seul gros volume au système. Permettant ainsi de cumuler les capacité des disques et d'exploiter en parallèle leurs bandes passantes. Améliorant très significativement les performances lors de la la lecture ou écriture des données.



Source : amenschool

espace exploitable : Total des disques.

Dans ce cas, deux disques de 500Go par exemples seront vu comme un volume logique de 1To (2*500Go)

Pours :

- Très performant.
- Permet de disposer de gros volumes.
- Capacité maximale.

Contres :

- N'offre aucune sécurité.

3.2 Raid 1 - Mirroring

Prérequis : 2 disques

Maximum : 32 *voir RAID Combiné 0+1

Le Raid 1 ou '**mirroring**' consiste à prendre un disque et en faire un 'miroir' sur un second disque.

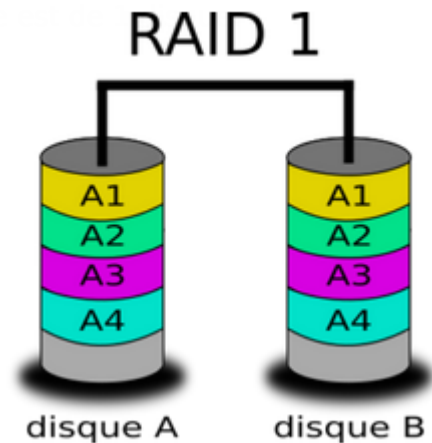
Chaque donnée est ainsi écrite en double, une fois sur le disque 1 et une fois sur le disque 2.

Bien que cela soit une perte nette de capacité, le système devient ainsi redondant et donc tolérant a la panne d'un des deux disques.

En situation normale, les données seront lues à partir des deux disques, augmentant ainsi les performances en lecture.

L'écriture en double de toute données quand à elle, diminue légèrement les performances en écriture.

La tolérance a la panne



Source : amenschool

espace exploitable : Total des disques / 2

Dans ce cas, deux disques de 500Go par exemples seront vu comme un volume logique de 500Go

Pours :

- Sécurisé (perte 1 disque).
- Idéal pour un disque système.

Contres :

- Seule la moitié de la capacité est utilisable.

3.3 Raid 5 - Striping with parity Across drives

Prérequis : 3 disques minimum

Maximum : 32 (dépendant du contrôleur utilisé)

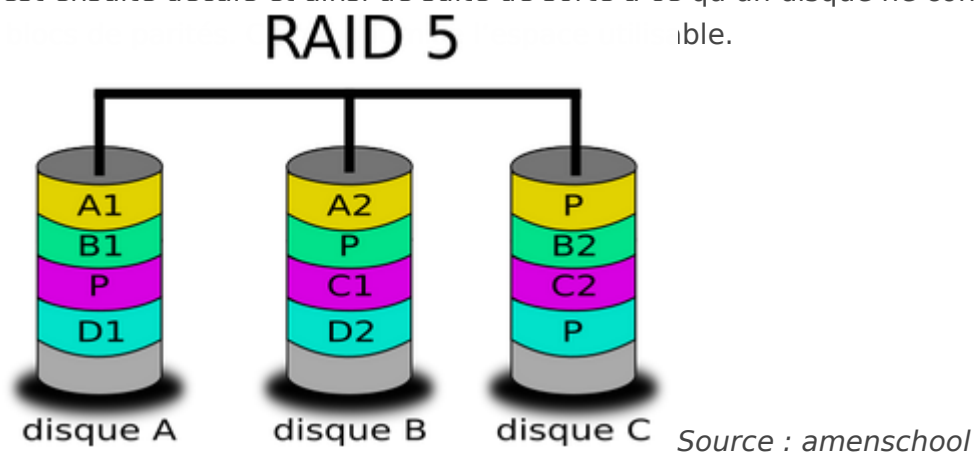
Le RAID 5 utilise un algorithme combinant les avantages des deux niveaux de RAID précédents, bien que fonctionnant de manière totalement différente.

En effet, dans ce mode de fonctionnement, les données seront constituées en blocs.

Chaque bloc sera coupé en deux et un bloc de parité sera créé et placé sur un autre disque que ceux contenant les demi blocs.

Sur le schéma suivant, le bloc A est coupé en deux blocs A1 et A2 placés chacun sur un disque puis un bloc de parité P est créé et placé sur le disque 3.

Le bit de parité est ensuite décalé et ainsi de suite de sorte à ce qu'un disque ne contienne pas uniquement les



Pour aller plus loin

Pour aller plus loin, l'opération faite sur les blocs au niveau binaire est un XOR.

Table XOR :

- $0 \text{ xor } 0 = 0$
- $0 \text{ xor } 1 = 1$
- $1 \text{ xor } 0 = 1$
- $1 \text{ xor } 1 = 0$

Ce qui donne par exemple : bloc initial 0110110111010100

Bloc1 (disque 1) = 01101101

Bloc 2 (disque 2) = 11010100

Bloc de parité (disque 3) = 10111001

espace exploitable : Capacité Totale des disques - Capacité de 1 disque

Dans ce cas, trois disques de 500Go. la capacité utilisable est donc $(500 \times 3) - 500 = 1\text{To}$

Pours :

- Sécurisé (perte 1 disque).
- Idéal pour un disque de données.
- Bon compromis entre capacité et fiabilité.

Contres :

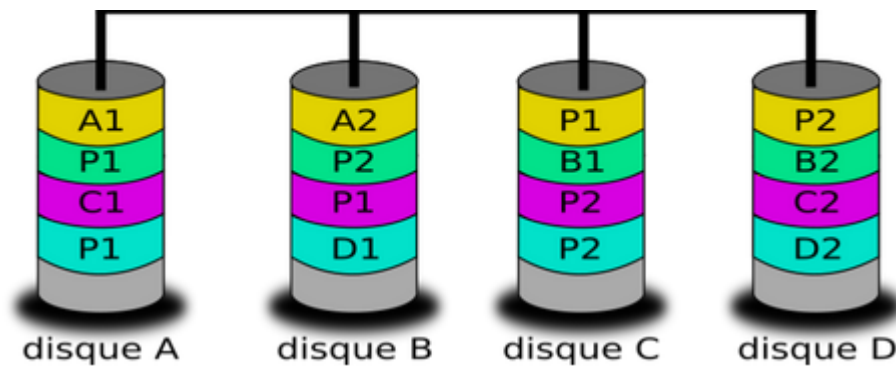
- perte de capacité équivalente à 1 disque
- impact sur les performances

3.4 Raid 6 - Striping with dual parity Across drives

Prérequis : 4 disques minimum

Maximum : 32 (dépendant du contrôleur utilisé)

Le RAID 6 fonctionne sur le même principe que le RAID 5 mais améliore néanmoins celui-ci en redondant le bloc de parité, ce qui augmente la tolérance à la panne.



Source : amenschool

espace exploitable : Capacité Total des disques - Capacité de 2 disque

Dans ce cas, quatre disques de 500Go. La capacité utilisable est $(500 \times 4) - (2 \times 500) = 1\text{To}$

Pours :

- Sécurité exceptionnelle (2 disques).
- Idéal pour volume de stockage infra.

Contres :

- Seule la moitié de la capacité est utilisable.
- Coûteux

3.5 Autres types de RAID

Il existe d'autres types de RAID assez exotiques, comme le RAID 3, 4, RAIDn ou RAIDdp

Pour des renseignements exhaustifs sur le RAID : [https://fr.wikipedia.org/wiki/RAID_\(informatique\)](https://fr.wikipedia.org/wiki/RAID_(informatique))

IV. Combiner les niveaux de RAID

Il est possible de combiner plusieurs Type de RAID entre eux pour tirer le maximum de performance ou de sécurité de ceux-ci. Cependant, ces solutions sont assez peu répandues à moins d'avoir un besoin spécifique car elles deviennent rapidement très coûteuses.

La notation se fera alors de la façon suivante :

01 ou 0+1 = un RAID 1 de volumes en RAID 0

10 ou 1+0 = un RAID 0 de volumes en RAID 1

4.1 Le RAID 0+1

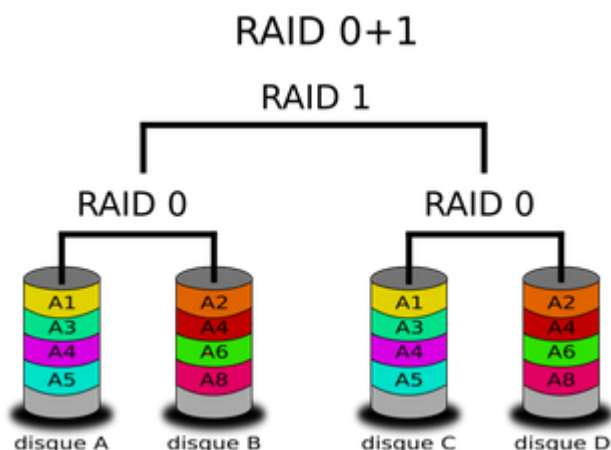
Prérequis : 4 disques minimum

Maximum : 32 (dépendant du contrôleur utilisé)

Il s'agit de prendre plusieurs volumes en '*striping*' et de les '*mirorer*' entre eux.

Ce qui permet d'agréger ensemble plusieurs volumes pour présenter un gros volume, qui sera ensuite lui-même mis en miroir sur un autre agrégat équivalent.

Dans ce niveau, même si une grande RAID 0 peut être perdue, on vise avant tout les volumes de grosses capacité



Source : amenschool

espace exploitable : Capacité totale des disques / 2

Dans ce cas, quatre disques de 500Go. La capacité utilisable est $(500*4)/2=1\text{To}$

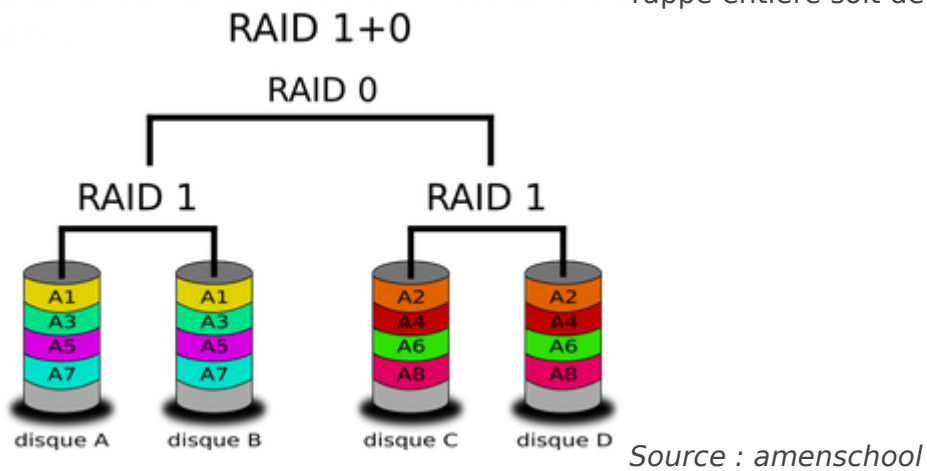
4.2 Le RAID 1+0

Prérequis : 4 disques minimum

Maximum : 32 (dépendant du contrôleur utilisé)

Il s'agit de l'inverse du RAID 01. Au lieu d'agréger des disques et de les '*miroriser*' ensuite. On agrégera une série de disques en miroirs. Cela permet de répartir différemment les pertes de disques au sein des grappes.

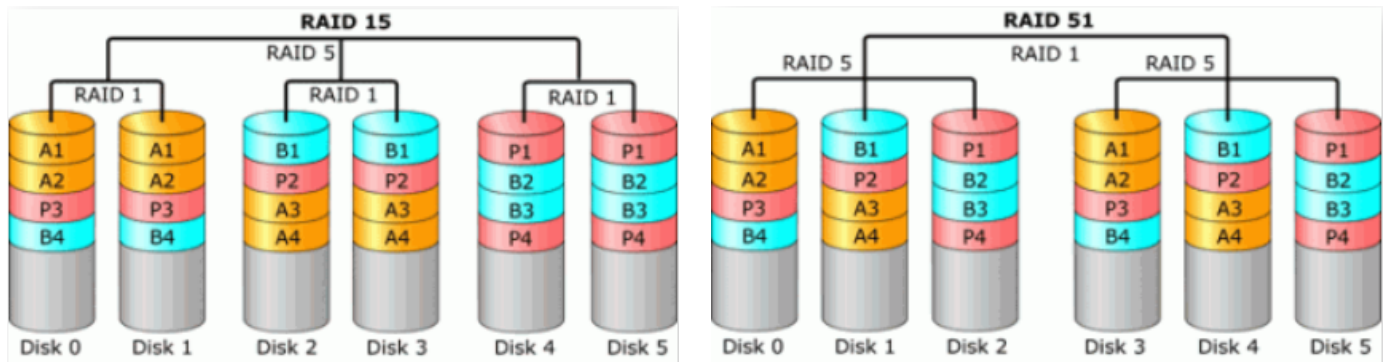
Ce niveau de RAID est extrêmement fiable car il faut qu'une grappe entière soit défectueuse avant de perdre les données.



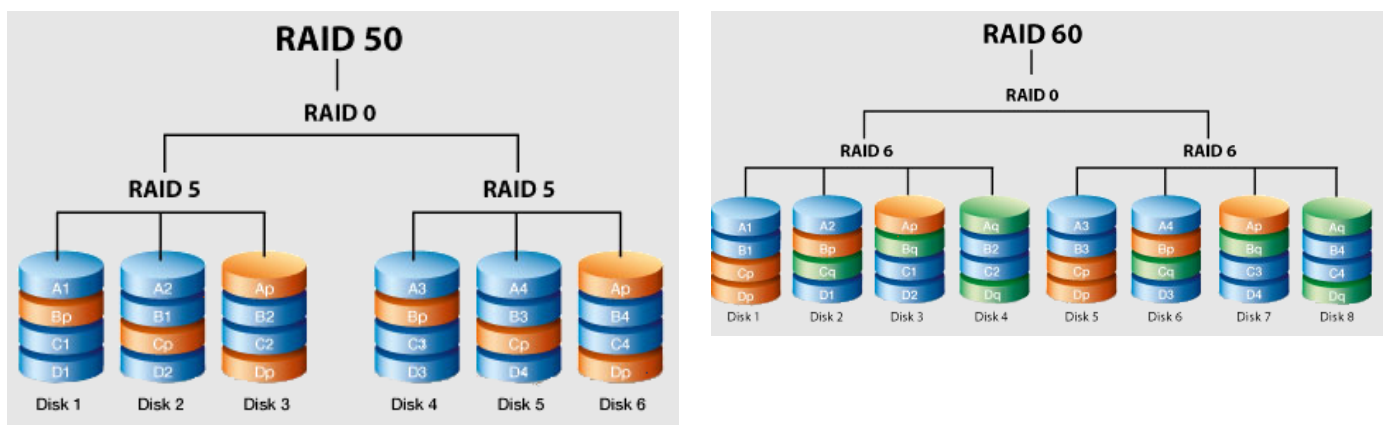
espace exploitable : Capacité totale des disques / 2

Dans ce cas, quatre disques de 500Go. La capacité utilisable est $(500 \times 4) / 2 = 1\text{To}$

4.3 Autres exemples



source : Storage-insider



Source : recoverhdd.com

Hardware - Un ordinateur



Difficulté : Débutant

Notions : ordinateur, composants.

I. Introduction

Un ordinateur est défini comme étant une machine automatique de traitement de l'information, obéissant à des programmes formés par des suites d'opérations arithmétiques et logiques.

Est considéré comme ordinateur tout système possédant les caractéristiques suivantes :

- Effectuer des calculs et opérations préprogrammées.
- Stocker des informations.
- Avoir une capacité de mise en réseau.

lien utile : <https://fr.wikipedia.org/wiki/Ordinateur>

II. Différents Types d'ordinateurs



Fixe (desktop)



Portable (laptop)



Tablette



Serveur tour



Serveur rackable



Micro ordinateur

Il existe également des “Barebones” qui sont des bases dites “nues” et sans OS pour créer un système personnalisé. Le coût de ceux-ci sont relativement bas, mais il faut prévoir d’ajouter un OS voire certaines pièces.



III. Les composants

