

Cours de bases de données,  
aspects systèmes,  
<http://sys.bdpedia.fr>

Supports de stockage

# Techniques de stockage

Contenu de ce cours :

- Où sont les données ? Mémoires RAM, mémoire persistante.
- Les performances
- Disques, SSD

Vocabulaire

Une "donnée" désigne une unité physique d'information : pour nous, une ligne d'une table, codée sous forme d'un enregistrement (*record*).

**Ces diapositives correspondent au support en ligne disponible sur le site <http://sys.bdpedia.fr/stock.html#s1-supports-de-stockage>**

# Problématique du stockage

Un SGBD gère deux types de mémoire.

- La mémoire centrale, **volative**, rapide mais risque de perte.
- La mémoire secondaire, **persistante**, stable mais lente.

Les données sont **toujours** en mémoire secondaire dans des fichiers, et **partiellement** copiées en mémoire centrale.

Une donnée, pour être traitée, doit impérativement être en mémoire centrale.

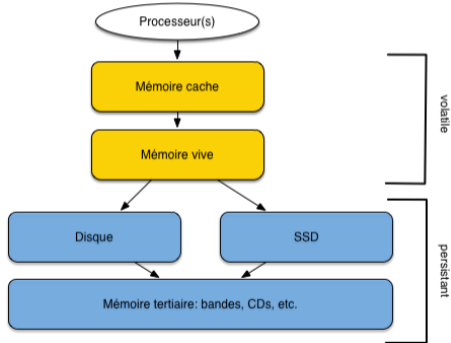
Problématique

Comment gérer au mieux ces deux ressources ?

# Les mémoires d'un ordinateur

Les mémoires dans un ordinateur forment une hiérarchie.

**Constat** : plus une mémoire est rapide, plus elle est chère , et moins elle est volumineuse.



# Performances des mémoires

Critères essentiels :

- **Temps d'accès** : connaissant l'adresse d'une donnée, quel est le temps nécessaire pour aller la chercher à l'emplacement mémoire indiqué par cette adresse ?  
Critère essentiel pour les opérations dites **d'accès direct**.
- **Débit** : volume de données lues par unité de temps dans le meilleur des cas.  
Critère essentiel pour les opérations dites de **lecture séquentielle**.

**Constat** : tout accès aux données s'effectue avec l'une de ces deux opérations.

# Quelques ordres de grandeur

Rappel : 1 Go = 1 000 Mo et 1 To = 1 000 Go.

Mémoire	Taille	Temps d'accès	Débit
cache	Qq Mo	$\approx 10^{-8}$ (10 nanosec.)	10+ Go/s
Principale (RAM)	Qq Go	$\approx 10^{-8} - 10^{-7}$ (10-100 nanosec.)	Qq Go/s
Disque	Qq To	$\approx 10^{-2}$ (10 millisc.)	100 Mo/s
SSD	Qq To	$\approx 10^{-4}$ (0,1 millisc.)	1+ Go/s

Un accès **direct** disque est (environ) un million de fois plus coûteux qu'en mémoire principale !

Le débit d'un disque est 20, 30, 40 fois plus lent que le débit RAM.

Un SSD est 10 à 100 fois plus performant qu'un disque magnétique.

# Importance pour les bases de données

Un SGBD doit ranger sur disque les données ;

- parce qu'elles sont trop volumineuses (de moins en moins vrai) ;
- parce qu'elles sont *persistantes* (et doivent survivre à un arrêt du système).

Le SGBD doit **toujours** amener les données en mémoire pour les traiter.

Si possible, les **données utiles** devraient résider le plus possible en mémoire.

**Principe général** : la capacité à placer les données **utiles** au plus près du processeur (donc en RAM si possible) est un facteur essentiel de performance.

# Organisation d'un disque

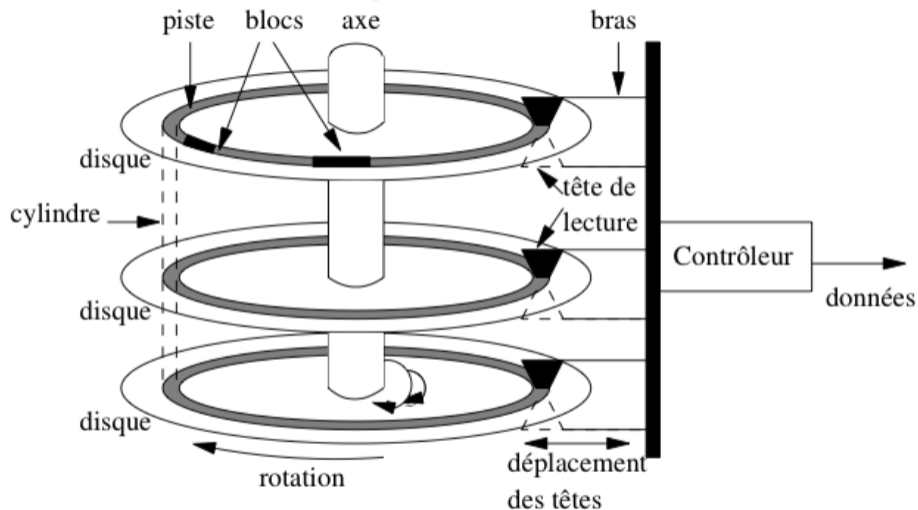
**Disque** : surface magnétique, stockant des 0 ou des 1, divisé en secteurs.

**Dispositif** : les surfaces sont entraînées dans un mouvement de rotation ; les têtes de lecture se déplacent dans un plan fixe.

- Le **bloc** est un ensemble de secteurs, sa taille est en général un multiple de 512 ;
- La **piste** est l'ensemble des blocs d'une surface lus au cours d'une rotation ;
- le **cylindre** est un ensemble de pistes situées sous les têtes de lecture.



# Structure d'un disque



# Disque = accès semi-direct = latence

Adresse = numéro du disque ; de la piste où se trouve le bloc ; du numéro du bloc sur la piste.

- **Délai de positionnement** pour placer la tête sur la bonne piste :
- **Délai de latence** pour attendre que le bloc passe sous la tête de lecture ;
- **Temps de transfert** pour attendre que le (ou les) bloc(s) soient lus et transférés.

**La latence d'un disque** : on ne peut pas lire une donnée avant qu'elle passe sous une tête de lecture : la **latence** (qq ms) est un facteur pénalisant.

# Granularité d'entrée/sortie : le bloc

**Granularité** : on lit toujours au moins un bloc, même si on ne veut qu'un octet ; le **bloc** est **l'unité d'entrée/sortie** !

⇒ le remplissage des blocs, si possible avec des données “proches” de la données requise initialement, est essentiel.

**Localité** : la latence en pratique dépend de la proximité des adresses successives demandées par l'application.

**Principe** : si deux données sont **proches** du point de vue applicatif, alors elles doivent être proches sur le disque (idéalement, dans le même bloc).

# Résumé

La gestion des disques n'est pas directement sous le contrôle de l'administrateur.

## Retenir :

- Accès (aléatoire) au disque : très lent par rapport à un accès en RAM (qq. ms).
- Débit du disque : faible (100 MO/s au mieux)
- SSD : 100 fois plus rapide qu'un disque (latence), 10 fois plus en débit

**Règle générale** : définir des données compactes (types) et stockées contiguement.

Organisation des données, les algorithmes d'accès, la concurrence : tout est conçu pour minimiser les accès aux disques, et surtout les accès **aléatoires**.