

Bases de Données Avancées : Optimisation (Introduction)

Thomas Gerald

February 2, 2026

Laboratoire Interdisciplinaire des Sciences du Numérique – LISN, CNRS

`thomas.gerald@lisn.upsaclay.fr`

Comment minimiser le temps de réponse pour une requête dans un SGBD ?

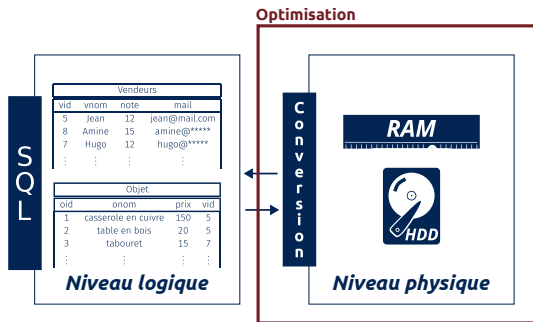
- Au niveau logique ?
- Au niveau physique ?

Comment minimiser le temps de réponse pour une requête dans un SGBD ?

- Au niveau logique ?
- Au niveau physique ? → **Oui**

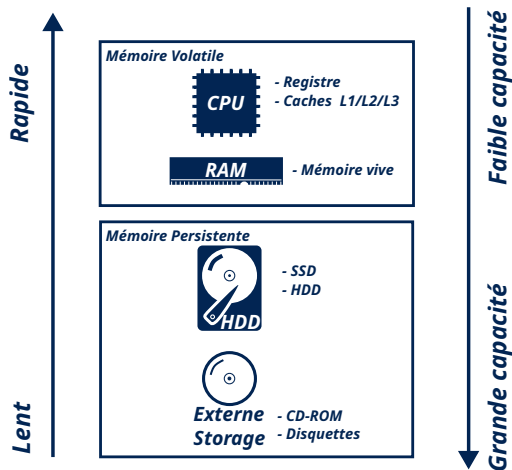
Comment minimiser le temps de réponse pour une requête dans un SGBD ?

- Au niveau logique ?
- Au niveau physique ? → Oui



Les capacités de la mémoire

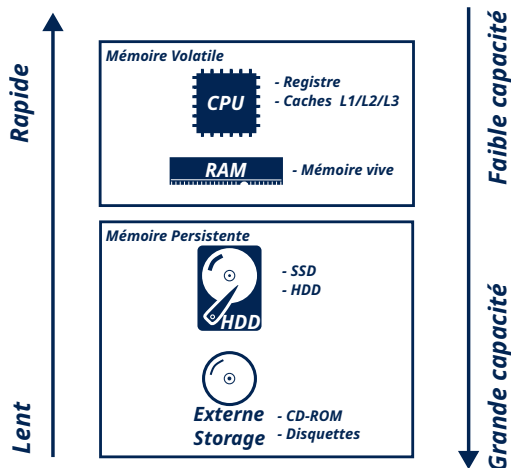
1. Mémoire du processeur (cache), quelques Ko à plusieurs dizaine de Mo
2. Mémoire vive, 4Go à 512Go
3. HDD/SSD, quelques 500Go à plusieurs To
4. CD-ROM/DVD/Blu-ray, 700Mo à 100Go



Optimisation : Les capacités mémoire

Les capacités de la mémoire

1. Mémoire du processeur (cache), quelques Ko à plusieurs dizaine de Mo
2. Mémoire vive, 4Go à 512Go
3. HDD/SSD, quelques 500Go à plusieurs To
4. CD-ROM/DVD/Blu-ray, 700Mo à 100Go

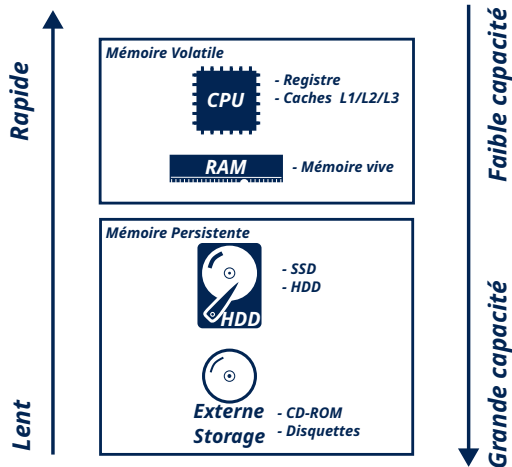


Besoins de grande capacité de stockage persistant → HDD/SSD

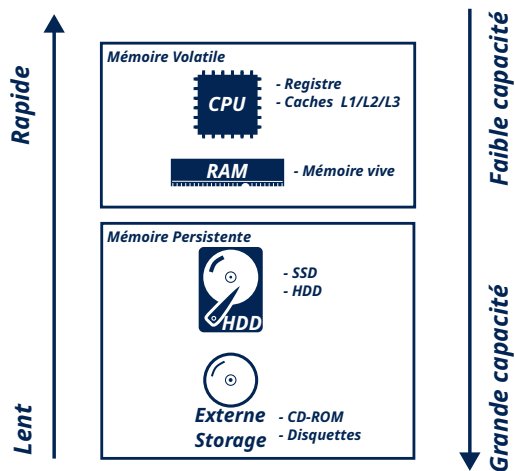
Optimisation : Les vitesses mémoire

Hiérarchie des temps d'accès (approximatif)

1. Mémoire du processeur (cache),
 $\approx 370\text{Go/s}$ à 2300Go/s
2. Mémoire vive, 20Go/s à 60Go/s (débit théorique $\text{largeur_bus} \times \text{frequence}$)
3. HDD/SSD, 100Mo/s à 8Go/s (Différence importante si données séquentielles ou non)
4. CD-ROM, quelques Mo/s

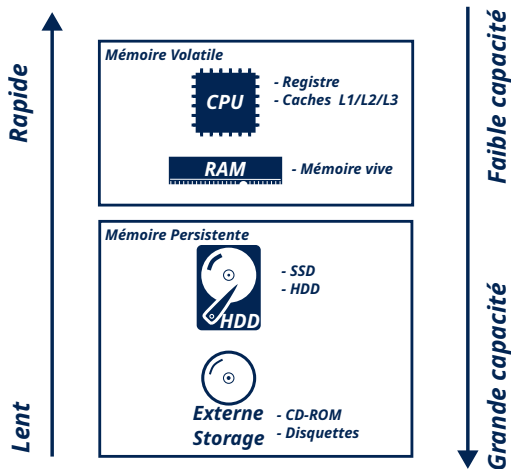


Optimisation: objectifs



Sur les supports persistants ?

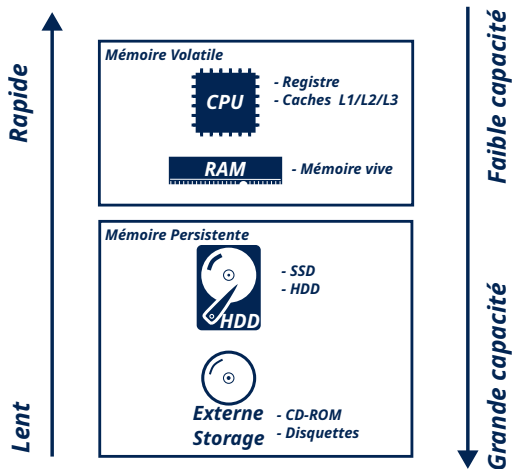
Optimisation: objectifs



Sur les supports persistants ?

- Minimiser le temps/nombre d'écritures
- Minimiser le temps/nombre de lectures

Optimisation: objectifs

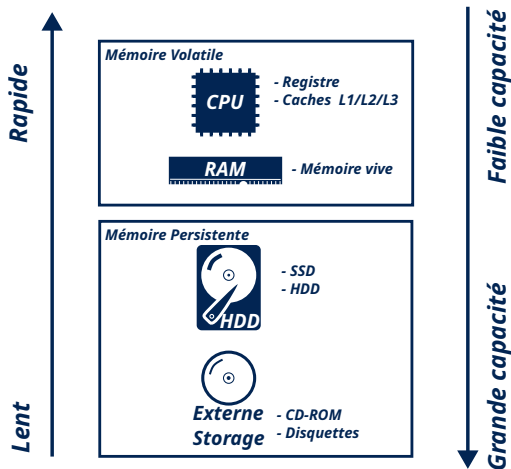


Sur les supports persistants ?

- Minimiser le temps/nombre d'écritures
- Minimiser le temps/nombre de lectures

Solutions ?

Optimisation: objectifs



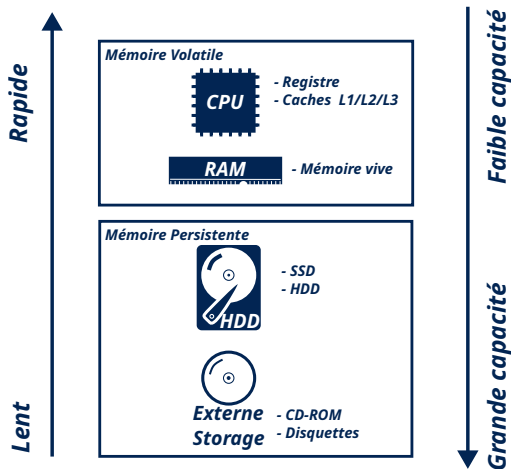
Sur les supports persistants ?

- Minimiser le temps/nombre d'écritures
- Minimiser le temps/nombre de lectures

Solutions ?

- Organisation de l'espace mémoire "fichiers" sur le disque

Optimisation: objectifs



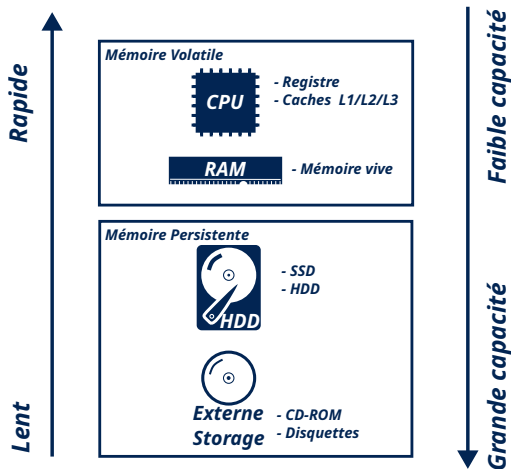
Sur les supports persistants ?

- Minimiser le temps/nombre d'écritures
- Minimiser le temps/nombre de lectures

Solutions ?

- Organisation de l'espace mémoire "fichiers" sur le disque
- Gestion de la mémoire vive (mettre en cache)

Optimisation: objectifs



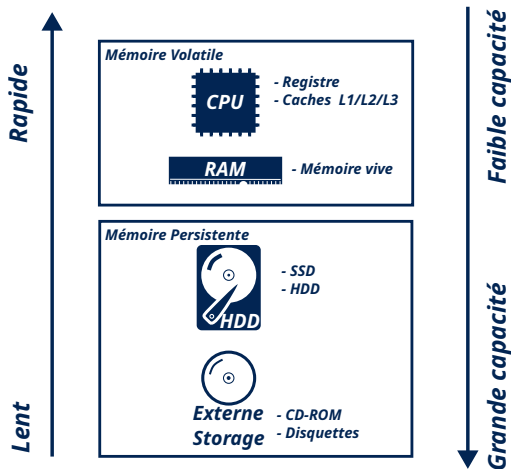
Sur les supports persistants ?

- Minimiser le temps/nombre d'écritures
- Minimiser le temps/nombre de lectures

Solutions ?

- Organisation de l'espace mémoire "fichiers" sur le disque
- Gestion de la mémoire vive (mettre en cache)
- Indexation des informations (retrouver rapidement des enregistrements)

Optimisation: objectifs



Sur les supports persistants ?

- Minimiser le temps/nombre d'écritures
- Minimiser le temps/nombre de lectures

Solutions ?

- Organisation de l'espace mémoire "fichiers" sur le disque
- Gestion de la mémoire vive (mettre en cache)
- Indexation des informations (retrouver rapidement des enregistrements)
- Optimisation des algorithmes

Gestionnaire de mémoire

Gestion de la mémoire

- Les relations (et d'autres informations) sont stockées sur un support persistant
- Pour exécuter une requête on travail sur les données en RAM

Gestion de la mémoire

- Les relations (et d'autres informations) sont stockées sur un support persistant
- Pour exécuter une requête on travail sur les données en RAM

→ Il faut faire des transferts entre le HDD/SSD et la RAM

Gestion de la mémoire

- Les relations (et d'autres informations) sont stockées sur un support persistant
- Pour exécuter une requête on travail sur les données en RAM

→ Il faut faire des transferts entre le HDD/SSD et la RAM

Optimisation :

Gestion de la mémoire

- Les relations (et d'autres informations) sont stockées sur un support persistant
- Pour exécuter une requête on travail sur les données en RAM

→ Il faut faire des transferts entre le HDD/SSD et la RAM

Optimisation :

- Optimiser les lecture/écriture sur le disque (organisation sur le disque)
→ Gestionnaire de disque/disc manager

Gestion de la mémoire

- Les relations (et d'autres informations) sont stockées sur un support persistant
- Pour exécuter une requête on travail sur les données en RAM

→ Il faut faire des transferts entre le HDD/SSD et la RAM

Optimisation :

- Optimiser les lecture/écriture sur le disque (organisation sur le disque)
→ Gestionnaire de disque/disc manager
- Optimiser ce que l'on garde dans le cache
→ Gestionnaire de cache/buffer manager

- Structure du stockage et les pages
- Le gestionnaire de disque
- Le gestionnaire de cache

Stockage

- Des relations (les tables)
- Des index (dans la suite)
- Des méta informations (statistiques)

Stockage

- Des relations (les tables)
- Des index (dans la suite)
- Des méta informations (statistiques)

→ On stocke les relations dans des fichiers !

Stockage

- Des relations (les tables)
- Des index (dans la suite)
- Des méta informations (statistiques)

→ **On stocke les relations dans des fichiers !**

- ☐ Les fichiers contiennent des pages

Stockage

- Des relations (les tables)
- Des index (dans la suite)
- Des méta informations (statistiques)

→ **On stocke les relations dans des fichiers !**

- ☐ Les fichiers contiennent des pages
- ☐ Une page contient les enregistrements (et d'autres informations)

Stockage

- Des relations (les tables)
- Des index (dans la suite)
- Des méta informations (statistiques)

→ **On stocke les relations dans des fichiers !**

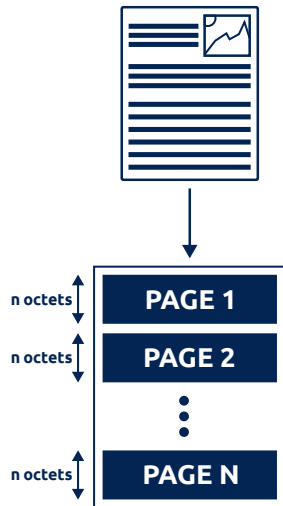
- ☐ Les fichiers contiennent des pages
- ☐ Une page contient les enregistrements (et d'autres informations)
- ☐ **Une page est de taille fixe**

Stockage

- Des relations (les tables)
- Des index (dans la suite)
- Des méta informations (statistiques)

→ On stocke les relations dans des fichiers !

- ☐ Les fichiers contiennent des pages
- ☐ Une page contient les enregistrements (et d'autres informations)
- ☐ Une page est de taille fixe



Une page :

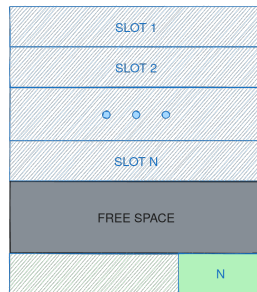
- Structure de données de taille fixe
- Contient les enregistrements
- Contient les informations d'accès aux enregistrements

Si la taille des enregistrements est fixe:

On peut accéder au n^{eme} enregistrement de la page :

$$@enregistrement \leftarrow @page + (n - 1) \times t$$

où @enregistrement l'adresse de l'enregistrement, @page l'adresse de la page et t la taille d'un enregistrement



Une page : Enregistrements de tailles variable (ex chaînes de caractères)

- Réserver un espace libre au “centre” de la page
- Stocker les offset des débuts de chaque enregistrements et la taille
- Stocker les enregistrements depuis la “fin” de la page
- Maintenir à jour un pointeur sur l’adresse (offset) de l’espace libre (en général dans l’en-tête)

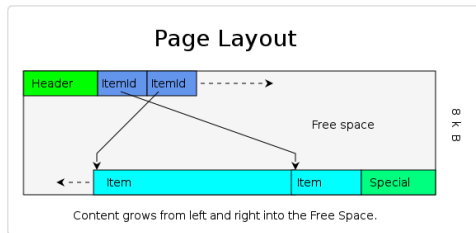


Figure 1: Une page avec enregistrements de tailles variables

En pratique avec Postgres

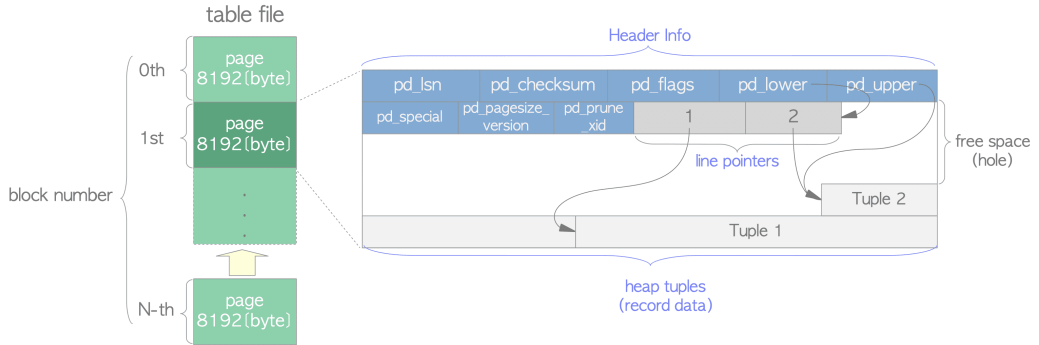


Figure 2: Les pages dans PostgreSQL

Enregistrement

- Stockage des relations, il s'agit d'une entrée de type n-uplets, d'index ou du journal
- Contient une entête avec plusieurs informations pour le contrôle de la concurrence, les valeurs nulles
- Un identifiant par enregistrement appelé **rid** (record identifier), composé d'un identifiant de la page et de l'enregistrement

Enregistrements

Les enregistrements contiennent les valeurs des attributs d'une instance de relation

Enregistrements

Les enregistrements contiennent les valeurs des attributs d'une instance de relation

- Suite de champs (valeur des attributs) stockés dans un espace mémoire (contigu) dans une page

Enregistrements

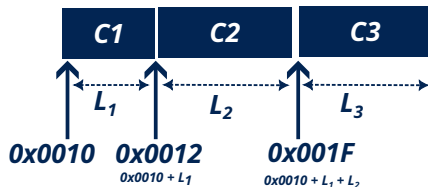
Les enregistrements contiennent les valeurs des attributs d'une instance de relation

- Suite de champs (valeur des attributs) stockés dans un espace mémoire (contigu) dans une page
- On retrouve les valeurs des champs en appliquant un offset depuis le début de l'enregistrement
⇒ Connaître la taille mémoire des champs (et l'ordre)

Enregistrements

Les enregistrements contiennent les valeurs des attributs d'une instance de relation

- Suite de champs (valeur des attributs) stockés dans un espace mémoire (contigu) dans une page
- On retrouve les valeurs des champs en appliquant un offset depuis le début de l'enregistrement
⇒ Connaître la taille mémoire des champs (et l'ordre)



⚠ Des champs avec taille dynamique (par exemple les chaînes de caractères) ?

Enregistrements avec des tailles variables

Les enregistrements contiennent des tuples d'une instance de relation

- Suite de champs stockés dans un espace mémoire (contigus)

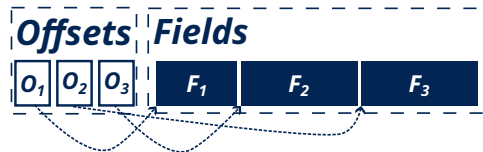


Figure 3:

Fichier

- Insertion d'une page
- Suppression d'une page
- Modification d'une page
- Rechercher une page

Fichier, pages et enregistrements : Opérations du gestionnaire de disque

Fichier

- Insertion d'une page
- Suppression d'une page
- Modification d'une page
- Rechercher une page

Page

- Rechercher un enregistrement
- Insertion et suppression d'un enregistrements
- Maintenir les méta-données de la page

Fichier, pages et enregistrements : Opérations du gestionnaire de disque

Fichier

- Insertion d'une page
- Suppression d'une page
- Modification d'une page
- Rechercher une page

Page

- Rechercher un enregistrement
- Insertion et suppression d'un enregistrements
- Maintenir les méta-données de la page

Enregistrement

- Recherche sur les attributs
- Modification d'attributs (et non suppression)

Conclusion :

- Organisation “hiérarchique” d’une table (des données) : Fichiers, Pages, Enregistrements
- Un fichier contient des pages de taille fixe
- Une page contient des enregistrements et les informations pour y accéder

Conclusion :

- Organisation “hiérarchique” d’une table (des données) : Fichiers, Pages, Enregistrements
- Un fichier contient des pages de taille fixe
- Une page contient des enregistrements et les informations pour y accéder

Comment optimiser sur le disque ?

Le gestionnaire de disque

- Organisation de l'espace mémoire “fichiers” sur le disque ←
- Gestion de la mémoire vive (éviter de lire sur le disque)
- Indexation des informations (retrouver rapidement des enregistrements)
- Optimisation des plans d'executions

Gestionnaire de disque

- Organiser les données en mémoire persistantes
- Lire les données sur le disque
- Écrire les données sur le disque

Comprendre les accès aux disques : Hard Disc Drive (HDD)

Les disques à plateaux

- Plusieurs plateaux ou disques (double face) contenant les données
- Une tête de lecture par plateau (dupliquées sur les deux faces)
- Plusieurs pistes pour chaque disque

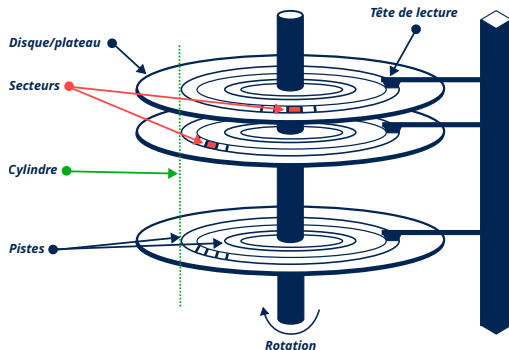


Figure 4: Schéma d'un disque à plateau

Comprendre les accès aux disques : Hard Disc Drive (HDD)

Les disques à plateaux : lire un secteur

1. Positionner la tête de lecture sur la piste ou le cylindre (**opération coûteuse**)
2. Effectuer une rotation pour accéder au début du secteur
3. Effectuer une rotation de lecture jusqu'à la fin du secteur

Comment stocker les pages pour optimiser l'accès ?

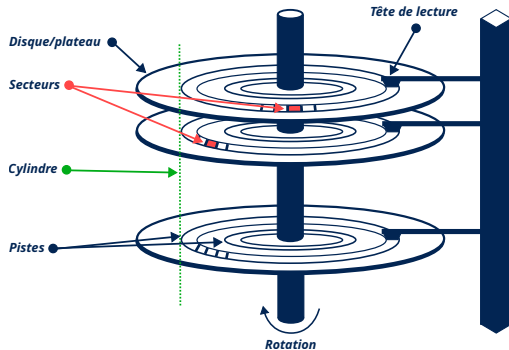


Figure 5: Schéma d'un disque à plateau

Comprendre les accès aux disques : Hard Disc Drive (HDD)

Les disques à plateaux : Lire une page

Considérons 5 pages combien d'opérations pour les lire ?

On considère :

- T le temps de déplacement d'une tête
- R le temps pour une rotation
- L le temps d'une lecture (rotation aussi)

1. Les pages sont écrites sur 5 blocs différents
2. Les pages sont écrites sur une même piste
3. Les pages sont écrites sur des blocs contigus
4. Les pages sont stockées sur un même cylindre sur des secteurs parallèles

Comprendre les accès aux disques : Hard Disc Drive (HDD)

Les disques à plateaux : Lire une page

Considérons 5 pages combien d'opérations pour les lire ?

1. Les pages sont écrites sur 5 blocs différents
2. Les pages sont écrites sur une même piste
3. Les pages sont écrites sur des blocs contigus
4. Les pages sont stockées sur un même cylindre sur des secteurs parallèles

On considère :

- T le temps de déplacement d'une tête
- R le temps pour une rotation
- L le temps d'une lecture (rotation aussi)

Réponse :

1. Déplacement de la tête sur la piste de P_1
2. Rotation du disque jusqu'au bloc de P_1
3. Lire P_1
4. Déplacement de la tête sur la piste de P_2
5. Rotation du disque jusqu'au bloc de P_2
6. ...

$$\rightarrow T \times 5 + R \times 5 + L \times 5$$

Comprendre les accès aux disques : Hard Disc Drive (HDD)

Les disques à plateaux : Lire une page

Considérons 5 pages combien d'opérations pour les lire ?

On considère :

- T le déplacement d'une tête
- R le temps pour une rotation
- L le temps d'une lecture (rotation aussi)

1. Les pages sont écrites sur 5 blocs différents
2. Les pages sont écrites sur une même piste
3. Les pages sont écrites sur des blocs contigus
4. Les pages sont stockées sur un même cylindre sur des secteurs parallèles

Comprendre les accès aux disques : Hard Disc Drive (HDD)

Les disques à plateaux : Lire une page

Considérons 5 pages combien d'opérations pour les lire ?

1. Les pages sont écrites sur 5 blocs différents
2. Les pages sont écrites sur une même piste
3. Les pages sont écrites sur des blocs contigus
4. Les pages sont stockées sur un même cylindre sur des secteurs parallèles

On considère :

- T le déplacement d'une tête
- R le temps pour une rotation
- L le temps d'une lecture (rotation aussi)

Réponse :

1. Déplacement de la tête sur la piste de P_1
2. Rotation du disque jusqu'au bloc de P_1
3. Lire P_1
4. Rotation du disque jusqu'au bloc de P_2
5. Lire P_2
6. ...

$$\rightarrow T + R \times 5 + L \times 5$$

Comprendre les accès aux disques : Hard Disc Drive (HDD)

Les disques à plateaux : Lire une page

Considérons 5 pages combien d'opérations pour les lire ?

On considère :

- T le déplacement d'une tête
- R le temps pour une rotation
- L le temps d'une lecture (rotation aussi)

1. Les pages sont écrites sur 5 blocs différents
2. Les pages sont écrites sur une même piste
3. Les pages sont écrites sur des blocs contigus
4. Les pages sont stockées sur un même cylindre sur des secteurs parallèles

Comprendre les accès aux disques : Hard Disc Drive (HDD)

Les disques à plateaux : Lire une page

Considérons 5 pages combien d'opérations pour les lire ?

1. Les pages sont écrites sur 5 blocs différents
2. Les pages sont écrites sur une même piste
3. Les pages sont écrites sur des blocs contigus
4. Les pages sont stockées sur un même cylindre sur des secteurs parallèles

On considère :

- T le déplacement d'une tête
- R le temps pour une rotation
- L le temps d'une lecture (rotation aussi)

Réponse :

1. Déplacement de la tête sur la piste de P_1
2. Rotation du disque jusqu'au bloc de P_1
3. Lire P_1
4. Lire P_2
5. Lire P_3
6. ...

$$\rightarrow T + R + 5 \times L$$

Comprendre les accès aux disques : Hard Disc Drive (HDD)

Les disques à plateaux : Lire une page

Considérons 5 pages combien d'opérations pour les lire ?

1. Les pages sont écrites sur 5 blocs différents
2. Les pages sont écrites sur une même piste
3. Les pages sont écrites sur des blocs contigus
4. Les pages sont stockées sur un même cylindre sur des secteurs parallèles

- T le déplacement d'une tête
- R le temps pour une rotation
- L le temps d'une lecture (rotation aussi)

1. Déplacement de la tête sur la piste de P_1
2. Rotation du disque jusqu'au bloc de P_1
3. Lire P_1, P_2, P_3, P_4, P_5

$$\rightarrow T + R + L$$

Gestionnaire de cache

Accès disque

- Trop coûteux !!!
- Les mêmes pages/enregistrements sont demandés plusieurs fois !

- Organisation de l'espace mémoire "fichiers" sur le disque
- Gestion de la mémoire vive (éviter de lire sur le disque) ←
- Indexation des informations (retrouver rapidement des enregistrements)
- Optimisation des plans d'executions

Le cache (ou buffer)

Le cache, objectifs :

Limiter les accès aux disques, charger en
RAM les données

Le cache (ou buffer)

Le cache, objectifs :

Limiter les accès aux disques, charger en RAM les données

- Transférer des pages en mémoire vive

Le cache (ou buffer)

Le cache, objectifs :

Limitier les accès aux disques, charger en RAM les données

- Transférer des pages en mémoire vive
- Garder les pages dont les accès seront ou sont requis par les processus
- Supprimer/remplacer les pages inutilisées

→ Gestionnaire de cache

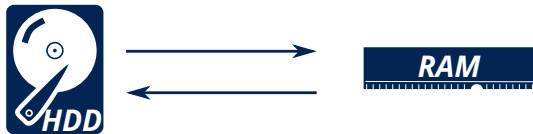


Figure 6: Stocker en mémoire vive les données d'accès fréquent

Le cache (ou buffer)

Le cache, objectifs :

Limiter les accès aux disques, charger en
RAM les données

Le cache (ou buffer)

Le cache, objectifs :

Limiter les accès aux disques, charger en RAM les données

- Transférer des pages en mémoire vive

Le cache (ou buffer)

Le cache, objectifs :

Limiter les accès aux disques, charger en RAM les données

- Transférer des pages en mémoire vive
- Garder les pages dont les accès seront ou sont requis par les processus
- Supprimer/remplacer les pages inutilisées

→ Gestionnaire de cache

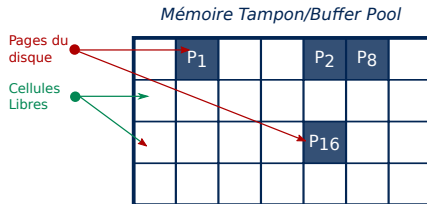


Figure 7: Le cache peut-être représenté comme un tableau de cellules, chaque cellule contient une page (ou vide)

Le cache (ou buffer)

Le cache, informations supplémentaires

- Stockage du nombre d'utilisateurs/processus pour une page (`pin_count`)
- Stockage d'un bit (`dirty bit`) informant de la modification de la page (l'écriture avant d'être fait sur le stockage dur est effectué dans le cache)

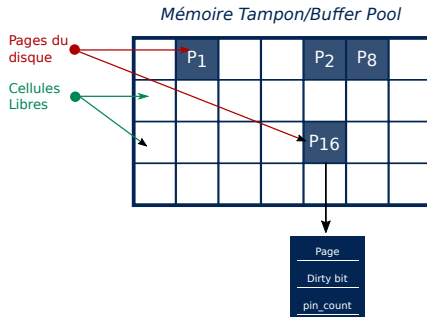
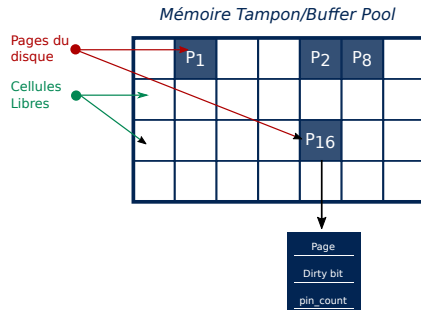


Figure 8: Le cache peut-être représenté comme un tableau de cellules, chaque cellule contient une page (ou vide)

Le cache (ou buffer)

Demande d'une page P au gestionnaire de mémoire

1. Vérification de la présence de la page dans le buffer
2. Si la page demandée n'est pas présente dans le cache
 - 2.1 Sélection d'une cellule C_i (remplacement ou vide)
 - 2.2 Si C_i n'est pas vide et **dirty_bit** vaut 1 → écriture de la page sur le disque
 - 2.3 Transfert de la page demandé depuis le disque vers le cache (remplacement)
 - 2.4 Initialiser **dirty_bit** et **pin_count** à 0 pour la page P
3. Incrémenter le compteur (**pin_count**)



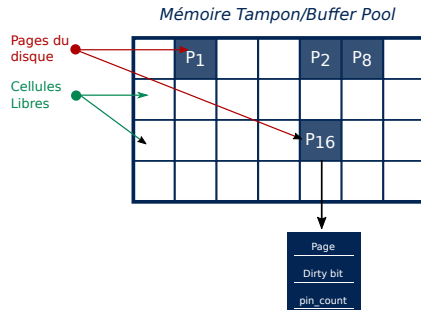
Le cache (ou buffer)

Libérer une Page P

1. Si P à été modifié par le processus appelant la libération

1.1 $\text{dirty_bit} \leftarrow 1$

2. $\text{pin_count} \leftarrow \text{pin_count} - 1$



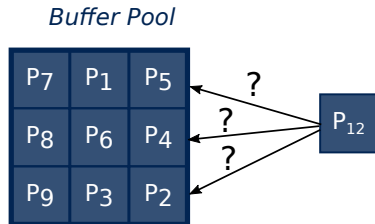
Le cache (ou buffer) : Stratégie de remplacement

Remplacer des pages du cache

Comment choisir la page qui sera remplacée ?

- **FIFO** First in First out
- **LRU** (least recently used) Les pages les moins récemment utilisées
- **MRU** (most recently used) Les pages utilisées récemment

Maintient d'une liste des cellules libres ou de cellules contenant des pages dont le `pin_count=0`

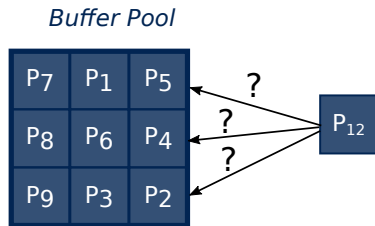


Le cache (ou buffer) : Stratégie de remplacement

Remplacer des pages du cache

Comment choisir la page qui sera remplacée ?

- **FIFO** First in First out
- **LRU** (least recently used) Les cellules les moins récemment utilisées
- **MRU** (most recently used) Les cellules utilisées récemment
- **Seconde chance**

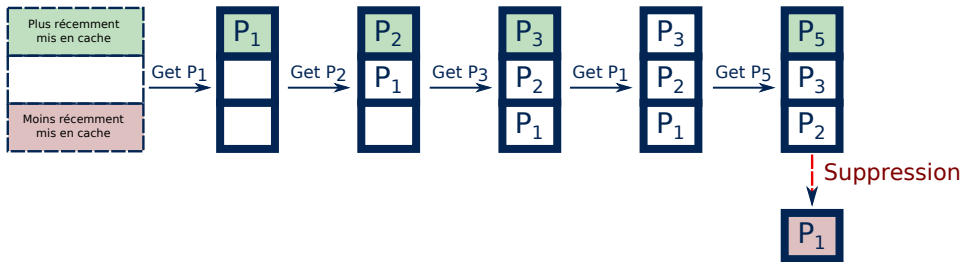


Stratégie de remplacement : FIFO

Premier arrivé, premier sorti :

Principe : Lorsque le cache est plein, on supprime l'élément le plus anciennement chargé.

- On maintiens une liste :
 - Les éléments de tête sont les plus anciens
 - Les éléments de queue sont les plus récents

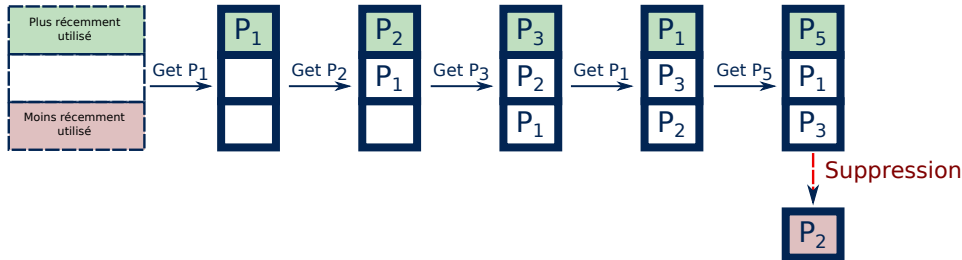


Stratégie de remplacement : LRU

Least recently used : stockage

Principe : Supprimer les pages non utilisées récemment

Mettre à jour la liste + HashMap (implémentation différentes possible) ! Faire remonter en tête de liste à chaque fois qu'un élément est demandé



⚠ En pratique on va considérer la page ne peut être enlevée du cache si le `pin_count` ≥ 1 .

Exercice

On dispose d'un cache d'une capacité de 4 pages vides (⚠ exemple jouet)

- Mesurer le nombre de transfert entre le disque et la RAM pour les opérations suivantes successives en utilisant la méthode LRU :
 1. Lecture de P_1, P_2, P_3 et P_4 (on libère dans le même ordre)
 2. Écriture sur P_1
 3. Lecture de P_5
 4. Lecture de P_2
 5. Lecture de P_1

Pour ces opérations écrire la liste correspondante

Stratégie de remplacement : Le problème du Sequential flooding

Accès

P_1
 P_2
 P_3
 P_1
 P_2
 P_3



Cellules

Sequential Flooding

Répéter une demande de suite de pages

Pour i dans A

Pour j dans B

lire la page j

- Quelles sont les conséquences pour les précédents algorithmes ?
- Quelle est le minimum d'accès possible ?

LRU

Accès

P₁
P₂
P₃
P₁
P₂
P₃
P₁



Cellules

P ₁		r = 1
P ₁	P ₂	r = 2
P ₃	P ₂	r = 3
P ₃	P ₁	r = 4
P ₂	P ₁	r = 5
P ₂	P ₃	r = 6
P ₁	P ₃	r = 7

Sequential Flooding

Répéter une demande de suite de pages

Pour i dans A

Pour j dans B

lire la page j

- Quelles sont les conséquences pour les précédents algorithmes ?
- Quelle est le minimum d'accès possible ?

Stockage : Utilisation du Système d'exploitation ?

Le système d'exploitation

- Système de fichier, gestionnaire disque
- Gestionnaire de cache

Pourquoi réimplémenter dans le SGBD le gestionnaire de mémoire ?

- Portabilité : fonctionnement sur différentes plateformes
- Très grands objets (par exemple sur plusieurs disques)
- Structurer les pages par rapport aux besoins de la base de données
- Gestion d'un buffer adapté aux données
- Gestion de la concurrence
- Gestion de la journalisation
- ...

Conclusion

- Organisation en page des données
- Les gestionnaires de mémoires

Dans la suite du cours L'unité de temps pour les calculs du coût sera un transfert disque/mémoire