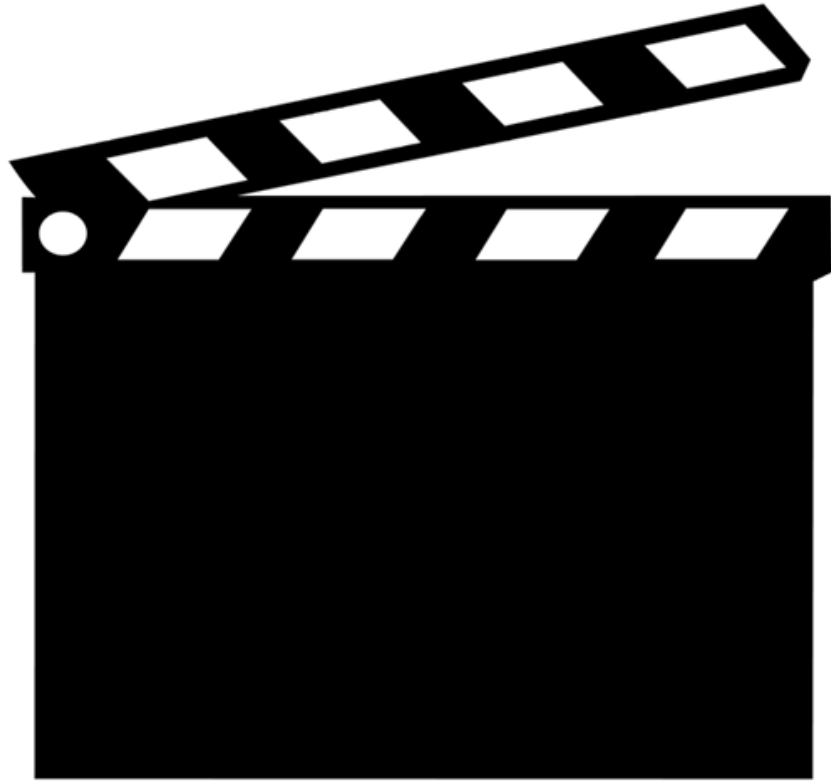


C018SA-W2-S1



SEMAINE 2 : Indexation

1. Introduction
2. Hiérarchie de mémoire
3. Fichiers indexés
4. Arbre-B
5. Hachage
6. Hachage dynamique
7. Multi-hachage

Introduction

- Dans cette semaine, nous allons considérer l'accès à de gros volumes de données
- Nous allons parler de techniques générales permettant d'accélérer cet accès
- Pour accéder à de gros volumes de données
 1. Les systèmes de fichiers
 2. Les systèmes de gestion de bases de données relationnelles (SGBD)

Systeme de fichiers

- Séquence de bits
- Interface
 - Lecture du fichier - transfert vers une application
 - Écrire du fichier - transfert depuis une application
 - Accès direct lire/écrire - à partir d'une application

Complexité de la couche d'accès aux données

- Trop bas niveau

Problème de **fiabilité**

Problème de **performance** : accès trop lent

- **Que feriez-vous ?**

Comment améliorer la lecture/écriture de fichier

1. Compression

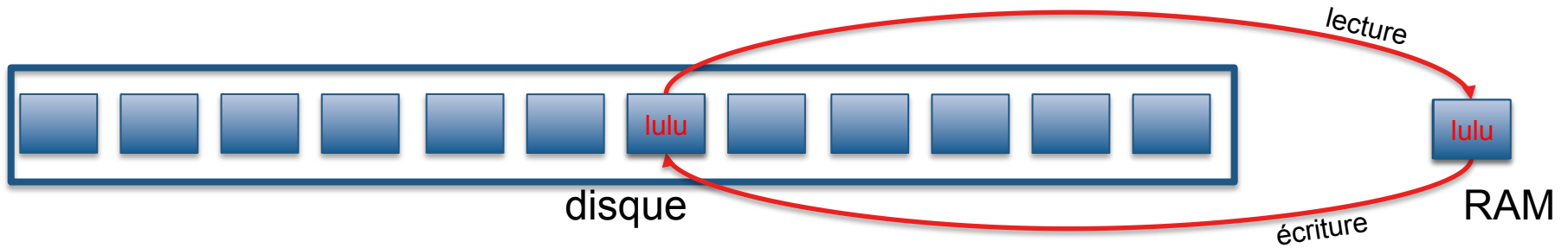
- De nombreux formats de données incluent des algorithmes de compression de données
- Zip, MP3, JPEG...

2. Partager le fichier en fragments

- Lecture/écriture en parallèle
 - ✓ 10 fragments sur 10 disques : gain de 10

3. Le partager en beaucoup de fragments : blocs

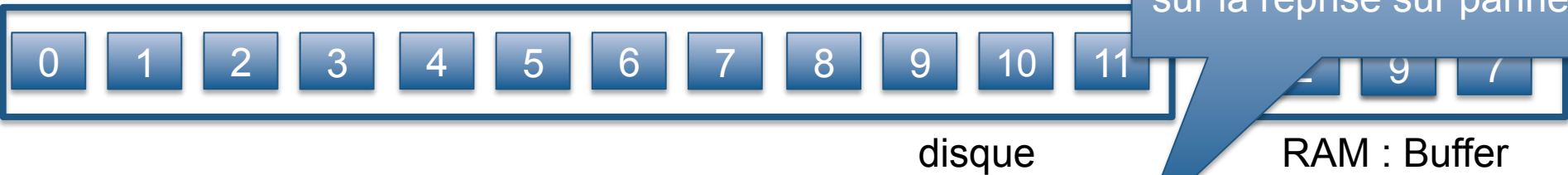
Partager le fichier en blocs



- Le bloc est l'unité de transfert avec la mémoire principale
- Ecrire/lire des blocs
 - ✓ Lire seulement ceux dont on a besoin
 - ✓ Ecrire seulement les blocs modifiés – le Δ

Buffer en mémoire principale

Nous reviendrons là-dessus en Semaine 5 sur la reprise sur panne



- On garde plusieurs blocs en mémoire : **buffer**
 - ✓ Suivant la place dont on dispose
- Prédicatif (pas simple) :
 - Prévoir quels blocs seront utilisés et les lire à l'avance
 - Quand on a besoin d'un nouveau bloc, prévoir quel bloc ne sera plus utile
- Stratégie de remplacement des blocs
 - Par ex: **LRU = Least Recently Used**

Regroupement (*clustering*)

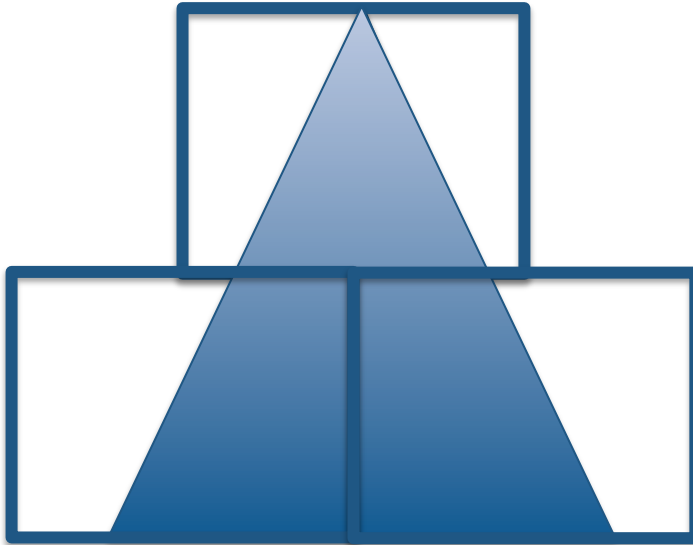
Regrouper dans un même bloc de données utilisées ensemble

Exemple : les informations disponibles sur un film

- Option A
 - ✓ Le synopsis est dans un bloc
 - ✓ Le casting dans un autre et
 - ✓ Les photos dans un troisième (sur le même disque)
 - ✓ 3 lectures de bloc par film
- Option B
 - ✓ Toutes les infos sur un film sont dans un même bloc
 - ✓ 1 lecture de bloc par film
 - ✓ Débit (nombre de requêtes par mn) multiplié par 3

Regroupement (fin)

- Et si une information ne tient pas sur un bloc
- Exemple Arbre XML...
- On la partage en **plusieurs blocs**



La suite du programme de la semaine ?

- Une technique pour améliorer le rapport performance/coût
 - ✓ Séquence 2 : La hiérarchie de mémoire
- Trois structures de données qui permettent d'améliorer les performances
 - ✓ Séquence 3 : Fichiers indexés
 - ✓ Séquence 4 : Arbre-B
 - ✓ Séquences 5-7 : Hachage