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

Structures d'index

Philippe Rigaux

le cnam

Structures d'index

Contenu de ce cours :

- Qu'est-ce qu'un index ?
- Principes généraux
- Index denses et non-denses

En l'absence d'un index, seule solution : le parcours séquentiel du fichier.

Avec un index : **accès direct à l'enregistrement**, amélioration **très importante** des temps de réponse

Ces diapositives correspondent au support en ligne disponible sur le site `http://sys.bdpedia.fr/arbreb.html#s1-indexation-de-fichiers`

Exemple 1 : une table

titre	année	...	titre	année	...
Vertigo	1958	...	Annie Hall	1977	...
Brazil	1984	...	Jurasic Park	1992	...
Twin Peaks	1990	...	Metropolis	1926	...
Underground	1995	...	Manhattan	1979	...
Easy Rider	1969	...	Reservoir Dogs	1992	...
Psychose	1960	...	Impitoyable	1992	...
Greystoke	1984	...	Casablanca	1942	...
Shining	1980	...	Smoke	1995	...

Exemple 2

La table des films, avec :

- 1 000 000 (un million) de films
- Un enregistrement = 1200 octets
- Un bloc = 4K => 3 enregistrements par bloc
- environ 300 000 blocs, 1,2 Go

Index = concept bien connu

Prenons un livre à contenu technique (pas une fiction !), par exemple un livre de recettes. Il contient (au moins) un index.

- L'index présente les termes importants, classés par ordre alphabétique
- À chaque terme sont associés les numéros de page où on trouve le terme.
- En parcourant l'index (par dichotomie !) on trouve la ou les pages qui nous intéressent.

Les index informatiques

Même principe ! C'est un fichier qui permet de trouver un enregistrement dans une table. Vocabulaire :

- **Clé d'indexation** = une **liste** d'un ou plusieurs attributs.
- Une **adresse** (déjà vu) est une adresse de bloc ou une adresse d'enregistrement.
- **Entrée d'index** : enregistrements de la forme `[valeur, addr]`, `valeur` est une valeur de clé.
- L'index est **trié** sur `valeur`

Exemples

Clés de recherche :

- Le titre du film (c'est aussi la clé primaire)
- l'année du film
- Une paire constituée du titre et de l'année

Opérations :

- Rechercher *Vertigo*
- Rechercher les films parus entre 1960 et 1975
- Rechercher les films commençant par 'V'

L'index ne sert à rien pour toute recherche ne portant pas sur la clé.

Le cas des dictionnaires

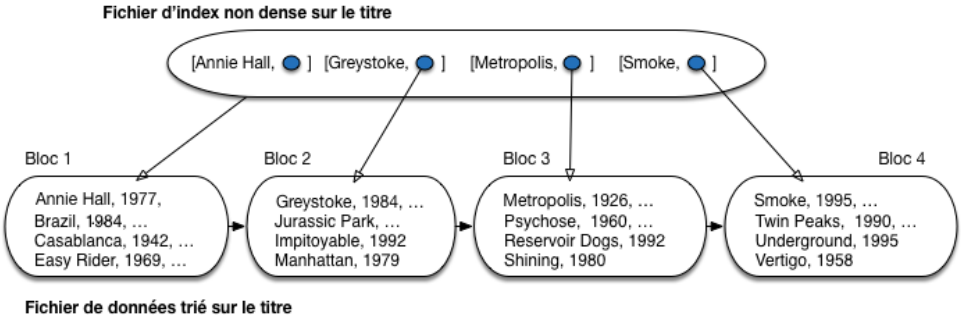
Les dictionnaires ont la particularité de **trier** les termes.

On peut alors créer un index qui ne référence que **le premier mot de chaque page**.

```
- ...
- ballon, page 56
- bille, page 57
- bulle, page 65,
- cable, page 72
- ...
```

On peut quand même utiliser cet index pour trouver n'importe quel mot. ("armée", "crabe", "botte", "belle").

Index non dense



Hypothèse : le fichier de données est **trié sur la clé**, comme un dictionnaire. **L'index ne référence que la première valeur de chaque bloc**.

Opérations

Recherches :

- **Par clé** : je recherche *Shining*
- **Par intervalle** : tous les films entre *Greystocke* et *Psychose*.
- **Par préfixe** : tous les films commençant par 'M'.

Mais pas par suffixe : tous les films terminant par 'e' ?

Exemple concret

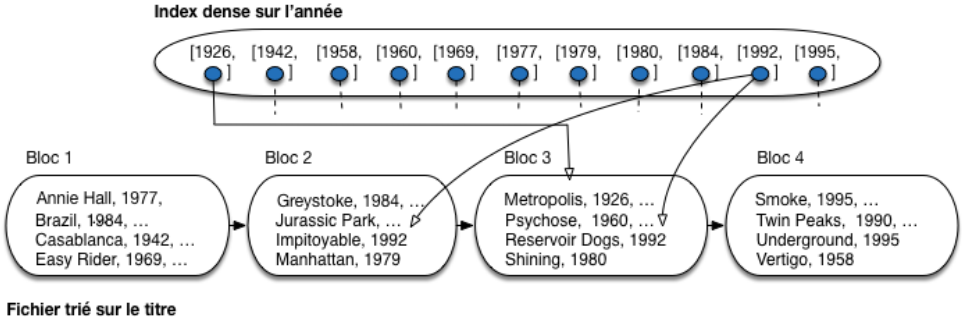
Sur notre fichier de 1,2 Go

- En supposant qu'un titre occupe 20 octets, une adresse 8 octets
- Taille de l'index : $300\,000 * (20 + 8) = 8,4 Mo$ octets

Beaucoup plus petit que le fichier !

Problème : maintenir l'ordre sur le fichier **et** sur l'index.

Index dense



Fichier de données non trié. Toutes les valeurs de clé sont représentées

Exemple concret

Sur notre fichier de 1,2 Go

- Une année = 4 octets, une adresse 8 octets
- Taille de l'index : $1\,000\,000 * (4 + 8) = 12$ Mo

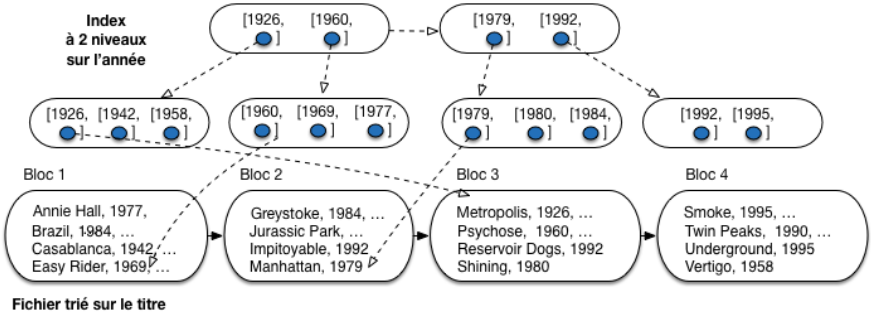
Encore 100 fois plus petit que le fichier.

Recherches :

- Par clé : comme sur un index non-dense
- Par intervalle (exemple [1950, 1979]) :
  recherche, dans l'index de la borne inférieure
  parcours séquentiel *dans l'index*
  à chaque valeur : accès au fichier de données

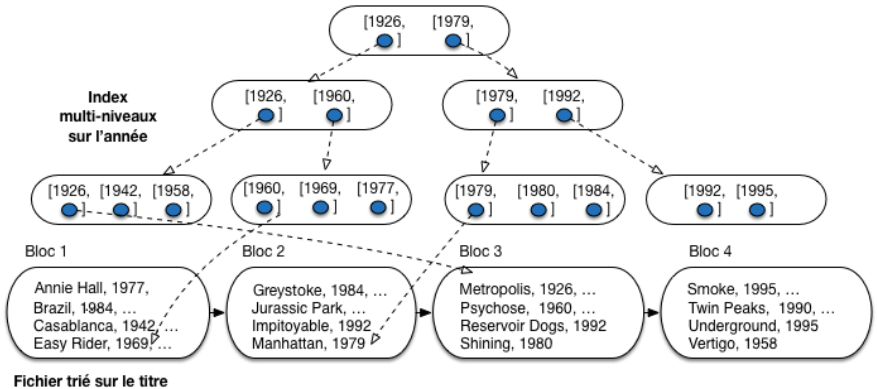
  Engendre des accès aléatoires.

Index multi-niveaux : on indexe l'index



- **Essentiel** : l'index est trié, donc on peut l'indexer par un second niveau **non-dense**
- Sinon ça ne servirait à rien (pourquoi ?)

Index multi-niveaux : jusqu'où ?



- Arrêt quand racine constitué d'un seul bloc.
- structure **hérarchique** ; recherche **de bas en haut**.

Résumé

Retenir :

- Un index accélère des opérations de recherche portant sur la **clé d'indexation** ou sur un préfixe de la clé
- La structure repose sur le tri des valeurs de clé indexées
- On peut créer un **index non dense** sur un fichier trié, **dense** sur un fichier non-trié.
- Les index multi-niveaux s'appuient sur un premier niveau dense, puis sur des niveaux supérieurs non-denses.

Les index ont un coût, il faut les maintenir au fur et à mesure des insertions dans le fichier indexé.