

Structures de données avancées :

MTH (Multidimensional trie hashing)

Pr ZEGOUR DJAMEL EDDINE
Ecole Supérieure d'Informatique (ESI)
www.zegour.univ.dz
email: d_zegour@esi.dz

Hachage digital multidimensionnel

Concepts

- Utiliser d arbres digitaux en mémoire, un arbre par attribut
- Les nœuds feuilles désignent des indexes au lieu des adresses de cases du fichier
- Afin de localiser un article de clé (k_1, k_2, \dots, k_n) , chaque k_i est recherché dans l'arbre digital correspondant à l'attribut i .
- Appliquer une fonction de "mapping" afin de transformer le d-uplet formé en une adresse linéaire.

Hachage digital multidimensionnel

Concepts

- Les cases sont représentées dans un espace d-dimensionnel où les d axes sont les d attributs.
- Un point avec les coordonnées (i_1, i_2, \dots, i_d) dans l'espace représente la case du fichier d'adresse $F(\langle i_1, i_2, \dots, i_d \rangle)$.
- Le fichier est un tableau à d dimensions rangé linéairement sur le disque.

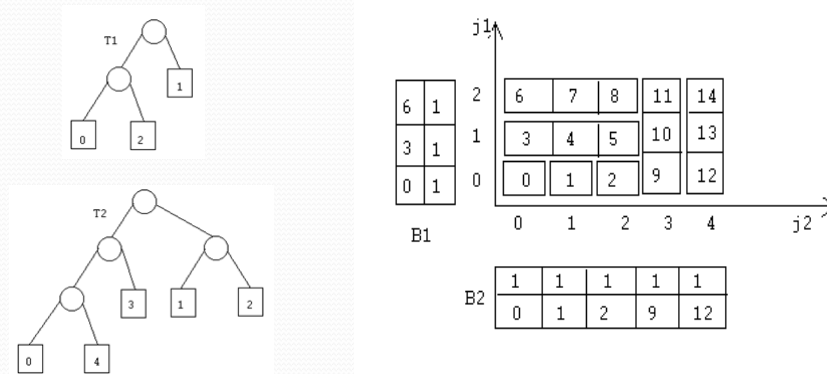
Hachage digital multidimensionnel

Concepts

- La fonction de mapping utilise la technique des tableaux extensibles dans n'importe quelle direction.
- A chaque extension un bloc(d-1)-dimensionnel (Segment) est rajouté.
- La fonction de mapping utilise d tableaux bi-dimensionnel contenant les bases et le facteurs multiplicatifs pour chaque dimension.

Hachage digital multidimensionnel

Exemple



Hachage digital multidimensionnel

Fonction de Mapping (Rappel)

Calcul de $F(j_1, j_2, \dots, j_d)$

1. Choisir $t = m$ tel que $B_m[j_m, m] = \text{Max} \{ B_r[j_p, r] \}, r = 1, 2, \dots, d$
2. $\text{Adr} = B_t[j_p, t] + \sum_{r=1, d} B_t[j_p, r] * j_r$ et $r \neq t$
3. Retourner Adr

Hachage digital multidimensionnel

Principe de construction

- Une insertion peut causer une collision.
- Le fichier est étendu par un bloc (d-1) dimensionnel de cases rajoutées à la fin du fichier
- Taille du bloc = $\prod_{j \neq t} (U_j + 1)$

t étant l'axe sur lequel se fait l'extension.

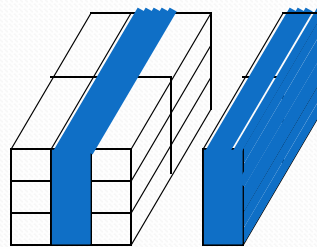
U_j index maximal dans l'arbre T_j

Hachage digital multidimensionnel

Insertion

En cas de collision :

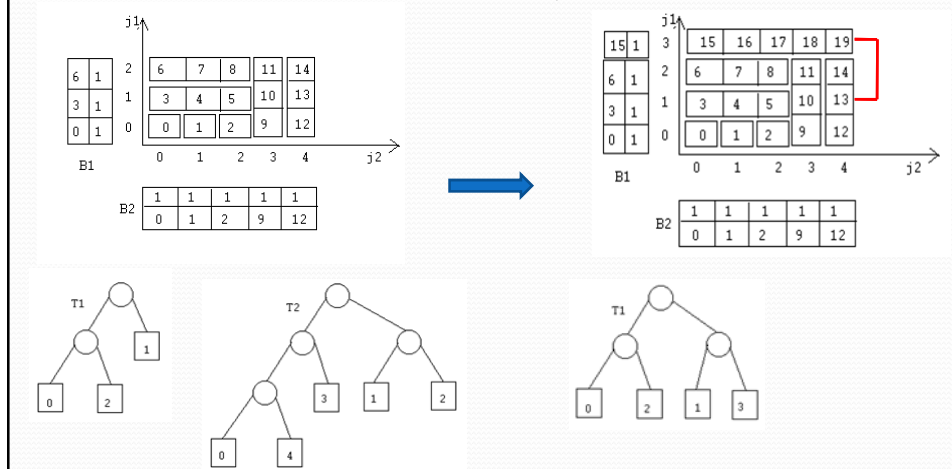
- Choisir un axe d'extension (de manière cyclique)
- Ajouter un nouveau segment
- Redistribuer les clés entre l'ancien et le nouveau segment



Extension par bloc de plusieurs cases

Hachage digital multidimensionnel

Exemple d'éclatement selon l'axe j_1



Hachage digital multidimensionnel:

Algorithme d'insertion

Insérer $K = (k_1, k_2, \dots, k_d)$

1. Appliquer T_1, T_2, \dots, T_d aux valeurs $k_1, k_2, \dots, k_d \rightarrow (i_1, i_2, \dots, i_d)$
2. Pour $j := 1, d$
 Si $i_j = \text{Nil}$
 - étendre le tableau selon l'axe j
 - mettre à jour l'arbre T_j
3. Calculer $F(i_1, i_2, \dots, i_d) \rightarrow$ adresse de case
4. Si Collision alors choisir un axe a cycliquement
 Étendre selon l'axe a
 Éclater l'arbre T_a
 Pour toutes les cases de l'ancien segment faire
 - re hacher les clés
 - si Nil ajouter segment

Hachage digital multidimensionnel:

Suppression

- Opération inverse de l'insertion
- Possibilité de contraction du fichier par fusion de blocs de cases

Hachage digital multidimensionnel

Types de requêtes

- Requête exacte

C'est l'algorithme de recherche, puisque tous les attributs sont spécifiés.

- Requête partielle

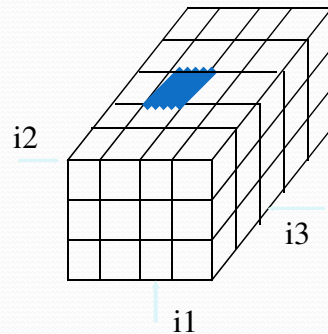
Considérer tous les indices pour les attributs non spécifiés

- Requête à intervalle

Déterminer un ensemble d'indices dans chaque dimension

Hachage digital multidimensionnel

Requête exacte



$\langle k_1, k_2, \dots, k_d \rangle$



Arbres digitaux

$\langle i_1, i_2, \dots, i_d \rangle$

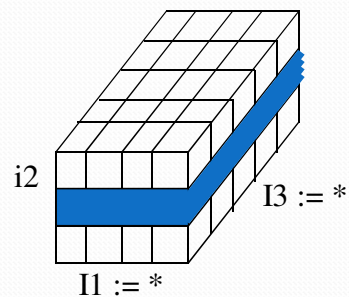


Fonction de mapping

$\langle \text{Adresse de case} \rangle$

Hachage digital multidimensionnel

Requête partielle



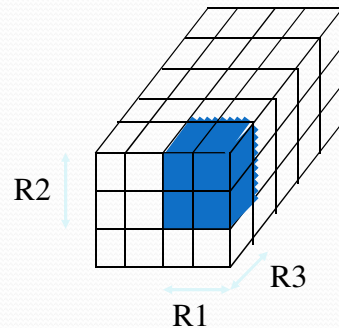
Spécification de q valeurs



Parcours d'un tableau à $(d-q)$ dimension

Hachage digital multidimensionnel

Requête par intervalle



Intervalle pour
chaque dimension



Sous tableau d-dimensionnel

Hachage digital multidimensionnel

Conclusion

- ✓ Fonction de mapping utilise un index : d tableaux contenant les bases et facteurs multiplicatifs
- ✓ Très bonnes performances d'accès.
- ✓ Facteurs de chargement de l'ordre de 40%. (extension du fichier par des segments de plus plus grands)
- ✓ Inconvénients : d-arbres digitaux en mémoire (sensibilité de la méthode)