Structures de données avancées :

MLH (Multidimensional linear hashing)

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

Hachage linéaire multidimensionnel

Considérations

- **Soit K = D_0 X D_1 X.....X D_{d-1} l'espace des clés**
  D_i est le domaine correspondant à l'attribut A_i.
- **Une clé k = (k_0, k_1, …, k_{d-1}) peut être vue comme un vecteur dans un espace D-dimensionnel, où chaque axe correspond à un attribut différent.**

Hachage linéaire multidimensionnel

Principe de la méthode

- **Appliquer LH consécutivement sur chaque axe**
- **Chaque division dépend seulement de l'axe courant.**
- **Les N premières collisions sont résolues selon uniquement l'axe 0 (comme si les clés ne sont composées que d'une seule composante k_0)**
- **En général, N=1 au départ**
- **En fin de cette séquence de division, le nombre de cases double.**

Hachage linéaire multidimensionnel

Principe de la méthode

- **Par la suite, si des collisions apparaissent, elles sont résolues le long de l'axe 1, jusqu'à ce que le fichier double de nouveau.**
- **Etc.**
- **Pour chaque séquence, on éclate à partir de la case 0.**
  Pour chaque séquence, le niveau est incrémenté
- **La case rajoutée est donnée par $P + 2^l$ (Si N=1, Sinon $P + 2^l.N$)**

  P pointeur de la case à éclater
  l : niveau de la fonction

Hachage linéaire multidimensionnel

Remarques importantes

- **En général, le prochain axe sur lequel les éclatements sont réalisés est déterminé de manière cyclique : (i+1) Mod d**
  (i est l'axe courant, d : nombre de dimensions)
- **Dans la formule $P + 2^l$, 2^l représente la taille du fichier avant les éclatements le long d'un axe donné.**
- **A la fin d'un cycle (d séquences d'éclatements), l'espace des adresses est incrémenté par un facteur de 2^d .**
  (au départ c'est 1.N, après axe 1 : 2^1N, après axe 2 : 2^2N, ...) .

Hachage linéaire multidimensionnel

Propriété

- **Si l désigne le nombre total de séquences de divisions réalisées sur tous les axes, alors**

  $$l = Ld + r$$

  L : nombre de cycles
- **L 'ensemble des cases a été éclaté (L+1) fois le long des axes {0, 1, ...r-1} et L fois le long des axes {r, ...d-1}**

Hachage linéaire multidimensionnel

Fonction de hachage

- Supposons $N = 1$ (au départ) et $I_i = \{0, 1, \ldots 2^i - 1\}$
- H_l, pour le niveau l (avec $l = Ld+r$) , sera définie comme une fonction de $D_0 \times D_1 \times \ldots D_{d-1} \rightarrow R_0 \times R_1 \times \ldots R_{d-1}$
  avec

  $R_i = I_{L+1}$ pour $i < r$ et

  $R_i = I_L$ pour $i \geq r$

Hachage linéaire multidimensionnel

Fonction de hachage

- Pour une clé k,

$$H_l(k) = (h_{L+1}(k_0), h_{L+1}(k_1), \ldots h_{L+1}(k_{r-1}), h_L(k_r), \ldots h_L(k_{d-1}))$$

- La prochaine séquence de division s'effectuera le long de l'axe r, on utilisera alors :

$$H_{l+1}(k) = (h_{L+1}(k_0), h_{L+1}(k_1), \ldots, h_{L+1}(k_{r-1}), h_{L+1}(k_r), h_L(k_{r+1}), \ldots, h_L(k_{d-1}))$$

Hachage linéaire multidimensionnel

Fonctions de division

H_1, H_2, …sont des fonctions de division pour MLH si elles satisfont la condition de rang suivante

$H_l : K \rightarrow \{0, 1, \ldots 2^l - 1\}$ (par projection)

Hachage linéaire multidimensionnel

Fonctions de division

H_1, H_2, …sont des fonctions de division pour MLH si elles satisfont la condition de division suivante:

Pour chaque niveau l du fichier avec l = L d + r et pour chaque clé k de K :

$H_{l+1}(k) = H_l(k)$ **ou bien**

$H_{l+1}(k) = (h_{L+1}(k_0), \ldots, h_{L+1}(k_{r-1}), h_L(k_r) + 2^L, h_L(k_{r+1}) \ldots h_L(k_{d-1}))$

$= H_l(k) + (0, 0, \ldots, 2^L, 0, 0, \ldots)$

<u>**Noter que :**</u> $h_{L+1}(k_r) = h_L(k_r) + 2^L$

Hachage linéaire multidimensionnel

Exemple (Dimension d = 2)

- **Les divisions sont contrôlées par le facteur de chargement a**

  a = n / (s.b + s'.b')
  n : nombre d'articles insérés
  s : nombre de cases primaires
  s' : nombre de cases en débordement
  b : capacité case primaire
  b' : capacité case de débordement

- **On prendre b = 3 et b' = 2. seuil s= 0.6**
- **On prendra le modulo pour les fonctions de division**

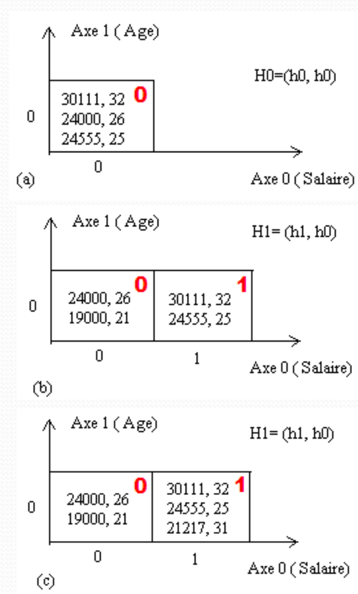
Hachage linéaire multidimensionnel

Première séquence de division

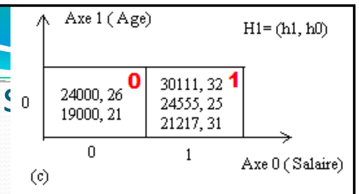
(a) Au départ, une seule case est allouée pour le fichier. Les 3 premiers articles sont insérés avec la fonction $H_0=(h_0, h_0)$ où $h_0(k)=k \text{ Mod } 1$

(b) insertion de l'article (19000, 21): Collision sur la case 0. Résolue par l'éclatement de la case 0 le long de l'axe 0, utilisant la fonction $H_1= (h_1\ h_0)$ avec $h_1(k) = k \text{ Mod } 2$

(c) La clé (21217, 31)est insérée dans la case 1.



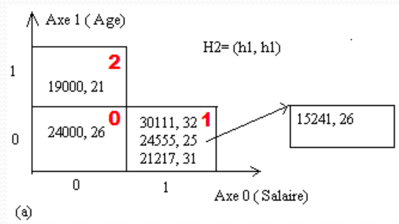
Hachage linéaire multidimens

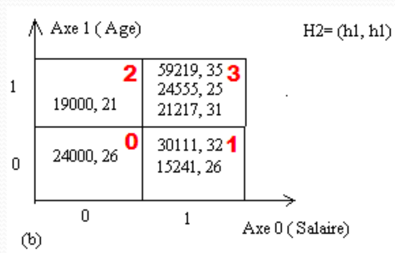


Deuxième séquence de division

(a) L'insertion de (15241, 26) cause une collision sur la case 1 le long de l'axe 1, puisque a=6/6 = 1.
L'article est mis en débordement et c'est la case 0 qui s'éclate.
L'éclatement est accompli par $H_2=(h_1, h_1)$.
Noter que
$H_2(k) = H_1(k)$ ou
$H_2(k) = H_1(K) + (0, 1)$

(b) Insertion de(59219, 35) : H2 donne (1, 1) : en dehors.
Donc H1 qui donne(1, 0) : case 1.
→ Éclatement.





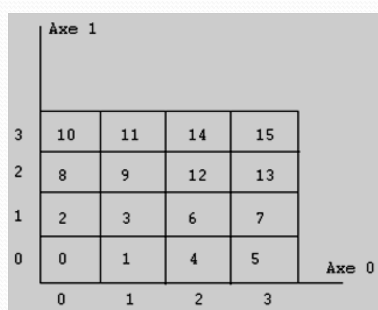
Hachage linéaire multidimensionnel

Adresses logiques des cases

durant les 4 premières séquences de division.

Problème :

Association entre les coordonnées et les adresses de cases ?



Hachage linéaire multidimensionnel

Tableau linéaire dynamique

- **Un tableau d-dimensionnel est dit *tableau linéaire dynamique* s'il peut être généré récursivement comme suit:**

  $A_0 = \{ A(0, 0, \ldots, 0) \}$
  $A_{l+1} = A_l \cup A_{l'}$

- **A_0= tableau d-dim contenant le seul élément $A(0, 0, \ldots, 0)$**

- **A_l et $A_{l'}$ sont des ensembles ordonnés tel que**
  (1) $A_l \cap A_{l'} = \varnothing$

$l = L.d + r$

Hachage linéaire multidimensionnel

Tableau linéaire dynamique

(2) Pour chaque élément $A(i_0, i_1 \ldots, i_{d-1})$ de A_l dans un ordre linéaire ajouter $A(i'_0, i'_1, \ldots, i'_{d-1})$ où

$i'_j = i_j$ **pour tout j, j#r**
$i'_r = i_r + 2^L$

avec $l = L.d + r$ (d : dimension ; L : nombre de cycles; l : nombre de séquences de divisions)

l = L.d + r

Hachage linéaire multidimensionnel

Exemple pour d = 2

1. A_0= {A(0, 0)}

2. A_1= A_0U A'_0
l=0 → L = 0 et r=0 donc 2^L = 1
A_1= { A(0, 0), A(1, 0) }
(Ajouter 2^L à la composante r=0)

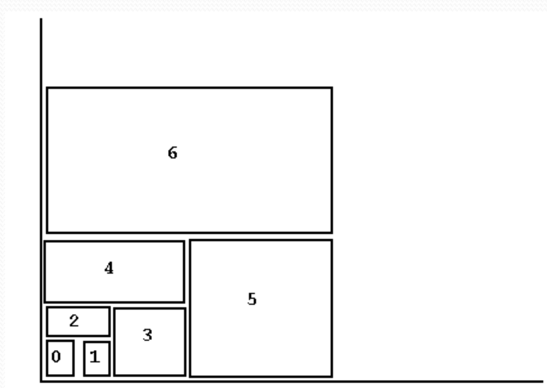
3. A_2= A_1U A'_1
l=1 ==> L=0 et r=1 donc 2^L = 1
A_2= { A(0, 0), A(1, 0) , A(0, 1), A(1, 1)}
(Ajouter 2^L = 1 à la composante r=1 des éléments de A_1)

4. A_3= A_2U A'_2
l=2 ==> L=1 et r=0 donc 2^L = 2
A_3= { A(0, 0), A(1, 0) , A(0, 1), A(1, 1), A(2, 0), A(3, 0) , A(2, 1), A(3, 1) }
Ajouter 2^L =2 à la composante r=0 des éléments de A2

Ect...

Hachage linéaire multidimensionnel

Évolution du fichier (vue par niveau)



Hachage linéaire multidimensionnel

Adresses logiques des cases pour l = 5:

Axe 1								
3	10	11	14	15	26	27	30	31
2	8	9	12	13	24	25	28	29
1	2	3	6	7	18	19	22	23
0	0	1	4	5	16	17	20	21
	0	1	2	3				Axe 0

Hachage linéaire multidimensionnel

Projection

- **μ un point dans I^d , I étant un intervalle quelconque**
  (I = [0..v] par exemple)
- **μ_i: i-ème coordonnée de μ, $0 \le i \le d-1$**
- **Projection Π_i: I^d ---> I^d**

Détermine pour μ donné, les coordonnées de la projection de μ le long de l'axe i,

$$\Pi_i(\mu) := (0, 0, \ldots, \mu_i, 0, \ldots 0)$$

$l = L.d + r$

Hachage linéaire multidimensionnel

Fonction de mapping

Une fonction M de I^d (Espace à d dim) dans J (espace à 1 dim)est une fonction de stockage si elle satisfait la contrainte suivante :

$M (H_{l+1}(k)) = M (H_l(k)) + 2^l$
$M(H_l(k)+(0,0,..,2^L, 0,0..)) = M (H_l(k)) + 2^l$

ou bien

$M(i_0, i_1, \ldots,i_{r-1},i_r+2^L, i_{r+1}, \ldots i_{d-1}) = M(i_0, i_1, \ldots, i_r, \ldots i_{d-1}) + 2^l$

où $l = L.d + r$

$l = L.d + r$

Hachage linéaire multidimensionnel

Fonction de Mapping

- **<u>Théorème 1 :</u>** $M (\mu) = \sum_{i=0,d-1} M (\Pi_i (\mu)),$

M est une fonction bijective
Exemple : $M(3, 2) = M(3, 0) + M(0, 2)$
La démo se fait par récurrence

- **<u>Théorème 2 :</u> Si $M(0,0,..,0) = 0$, alors M est une fonction sur la droite I_l ($I_l = \{0, 1, 2, \ldots ,2^{l-1}\}$ avec $l = L.d + r$**

Hachage linéaire multidimensionnel

Fonction de Mapping

<u>Théorème 3 :</u>
M est complètement caractérisée par :
$M(\mu) = \sum_{j=0,d-1}(\sum_{i=0,x_j}(2^{d.i+j} \cdot b_{ij}))$

où

$x_j = \mathrm{Ent}(\mathrm{Log}\ \mu_j)$

et $b_{x_j}j\ b_{x_j-1}j \ldots b_0 j$ est la représentation binaire de μ_j

Exemple **(d=2)**

- **Calcul de M(6, 3)**

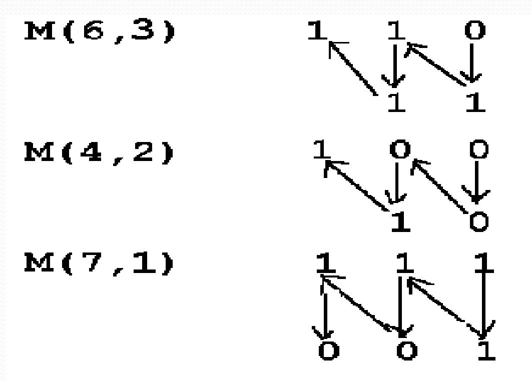
$(6)_2 = 1\,1\,0$ $\quad 0.2^0 + 1.2^2 + 1.2^4$
$(3)_2 = 1\,1$ $\quad 1.2^1 + 1.2^3$
M(6, 3) = 30

- **Calcul de M(4, 2)**

$(4)_2 = 1\,0\,0$ $\quad 0.2^0 + 0.2^2 + 1.2^4$
$(2)_2 = 1\,0$ $\quad 0.2^1 + 1.2^3$
M(4, 2) = 24

Hachage linéaire multidimensionnel

Exemple



l = L.d + r

Hachage linéaire multidimensionnel

Connaissant le niveau l du fichier on peut déterminer M(x, y, z, …)

- **Algorithme :**

  1. ***Déduire L et r***
  2. ***Calcul de M***
  3. ***l := l - 1***
  4. ***Reprendre à partir de 1***

- **Pour la dimension 2 on a les formules :**

  $M(a, b) = M(a, b-2^L) + 2^l$ si $r = 1$

  $= M(a-2^L, b) + 2^l$ si $r = 0$

l = L.d + r

Hachage linéaire multidimensionnel

Connaissant le niveau l du fichier on peut déterminer M(x, y, z, …)

Exemple : Calcul M(3, 3) sachant que l = 3

Si l = 3 ==> L = 1 et r = 1
$M(3, 3) = M(3, 1) + 2^3$

Pour l = 2 ==> L = 1 et r = 0
$M(3, 1) = M(1, 1) + 2^2$

Pour l = 1 ==> L = 0 et r = 1
$M(1, 1) = M(1, 0) + 2^1$

Pour l = 0 ==> L = 0 et r = 0
$M(1, 0) = M(0, 0) + 2^0$

→ $M(3, 3) = 2^3 + 2^2 + 2^1 + 2^0 = 15$

Hachage linéaire multidimensionnel

Fonction inverse

Étant donné une adresse m obtenue par la fonction M, et $b_x m$, $b_{x-1}m$, ...$b_0 m$ sa représentation binaire alors la fonction inverse R peut être exprimée comme :

$$R = (m_0, m_1, ..., m_{d-1})$$

avec

$$m_j = \sum (2^{k \text{ div } d} . b_k m) \text{ , k dans } E_m(j)$$

$E_m(j)$ = { k tel que k = d.i + j et $b_k m$ est <u>dans</u> la représentation binaire de m }

Pour déterminer m_j, on ne considère que les bits de rang k = di + j
j = 0, 1, .., ,d-1
i = 0, 1, ...

Hachage linéaire multidimensionnel

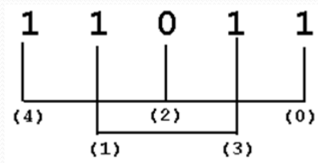
Exemples

- **Pour d = 2**
  - $E_m(0) = \{ 0, 2, 4, ... \}$
  - $E_m(1) = \{ 1, 3, 5, ... \}$
- **Pour d = 3**
  - $E_m(0) = \{ 0, 3, 6, ... \}$
  - $E_m(1) = \{ 1, 4, 7, ... \}$
  - $E_m(2) = \{ 2, 5, 8, ... \}$

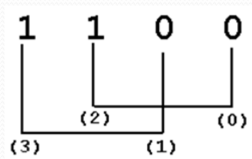
Hachage linéaire multidimensionnel

Exemples

Pour d=2 ; m = 27, puis m= 12



→ (5, 3)



→ (2, 2)

$l = L.d + r$

Hachage linéaire multidimensionnel

Requête exacte

Soit $k = (k_o, k_1, ..k_{d-1})$ et $l = Ld + r$ le nombre de séquences de divisions accomplies, l'algorithme qui suit rend l'adresse logique où l'article est rangé

1. Appliquer $H_l(k) \leftarrow (h_{L+1}(k_o), \ldots, h_{L+1}(k_{r-1}), h_L(k_r), \ldots h_L(k_{d-1}))$

2. $m \leftarrow M(H_l(k)) = \sum_{i=o,\, d-1}(M(\Pi_i(H_l(k)))$

3. Si $m < P$ alors $m := M (H_{l+1}(k))$

(P : prochaine case à éclater)

$l = L.d + r$

Hachage linéaire multidimensionnel

Requête exacte : Exemple

Trouver l'adresse de (30121, 35) en supposant que l = 3 et P = 4

1. H3(k)=(h2(30121), h1(35)) = (1, 1)

2. M(1, 1) = M(1, 0) + M(0, 1)
M(1, 0) = 1; M(0, 1) = 2
m ← 3

3. m < P donc H4(k)=(h2, h2)(k)=(1,3)
M(1, 0) = 1; M(0, 3) = 10
m ← 11

$l = L.d + r$

Hachage linéaire multidimensionnel

Requête partielle

- **Rechercher des clés de la forme (ko, k1, ...,kd-1)**

ki est
-soit une valeur du domaine Di
-soit une valeur non spécifiée ('*').

- **Soit k une requête partielle, on désigne par $b^l(k)$ l'ensemble des cases à rechercher (l = L.d + r)**

$$b^l(k)=\{b^0(k_0),\ldots, b^r(k_r), \ldots b^{d-1}(k_{d-1})\ \}$$

où $b^i(k_i)$ est défini comme suit :

k_i est spécifié et i < r : $b^i(k_i) = h_{L+1}(k_i)$
k_i est spécifié et i ≥r : $b^i(k_i) = h_L(k_i)$

l = L.d + r

Hachage linéaire multidimensionnel

Requête partielle

k_i = * et i # r : $0 \leq b^i(k_i) \leq 2^m - 1$
avec m = L+1 si i < r et m = L si i > r

k_i = * et i=r : $0 \leq b^i(k_i) \leq 2^{L+1} - 1 + Last_i$
Last = P - 1 : dernière case éclatée

$Last_i$ dénote la i-ieme (i = r) coordonnée obtenue par l'application de la fonction inverse R

l = L.d + r

Hachage linéaire multidimensionnel

Requête partielle

Soit la requête $k=(k_0, k_1 \ldots, k_{d-1})$ et l = Ld + r, l'algorithme retourne les adresses des cases où les articles avec les attributs spécifiés sont rangés.

Pour Chaque b dans $b^l(k)$
$m \leftarrow M(b_0, b_1, \ldots, b_{d-1})$
Si k_r est spécifié et m < P
$b_r \leftarrow h_{L+1}(k_r)$
$m \leftarrow M(b_0, b_1, \ldots b_{d-1})$
Fsi
Écrire(m)
Finpour

Hachage linéaire multidimensionnel

Conclusion

- **Très bonnes performances**
- **Pour des divisions non contrôlées avec b=5 et b'=1, le coût d'accès est de 1.09 indépendamment de la dimension et le facteur de chargement est de 67%.**
- **Pour des division contrôlées, même avec un seuil de 90%, pour b=10 et b'=2, le temps d'accès est de 1.74.**