

TD n°1

Techniques de fragmentation et allocation

Introduction

La fragmentation est le processus de décomposition d'une base de donnée logique en un ensemble de "sous" bases de données. Cette décomposition doit être sans perte d'information.

Les règles de fragmentation à respecter sont :

- **La complétude** : pour toute donnée d'une relation R, il existe un fragment R_i de la relation R qui possède cette donnée.
- **La reconstruction** : pour toute relation décomposée en un ensemble de fragments R_i , il existe une opération de reconstruction.
- **La disjonction*** : assure que les fragments d'une relation sont disjoints deux à deux.

Les techniques de **fragmentation dirigée par des requêtes (Query-Driven Partitioning)** vise à optimiser les requêtes les plus fréquentes, avec connaissance préalable des requêtes ainsi que les fréquences d'accès des requêtes.

1 Exemple d'une fragmentation et allocation

Trois universités (U_1, U_2, U_3) ont décidé de mutualiser leurs bibliothèques et leur service de prêts, afin de permettre aux étudiants d'emprunter des ouvrages dans toutes les bibliothèques des universités. Par exemple, un étudiant de U_1 pourra emprunter à la bibliothèque de U_2 .

La gestion commune des bibliothèques et des emprunts est effectuée par une base de données répartie, dont le schéma global est le suivant :

- EMPLOYE (Id_emp, nom, adresse, statut, affectation)
L'attribut *affectation* désigne la bibliothèque où travaille l'employé.
- ETUDIANT (Id_etu, nom, adresse, université, cursus, emprunts)
L'attribut *université* indique l'université où est inscrit l'étudiant.
- OUVRAGE (Id_ouv, titre, éditeur, année, domaine, stock, site)
L'attribut *site* indique la bibliothèque qui gère cet ouvrage. L'attribut *domaine* permet de classer les ouvrages en catégories (physique, maths, info, médecine, etc.). L'attribut *stock* désigne le nombre d'ouvrages restant disponibles au prêt.
- AUTEUR (Id_ouv, nom_auteur)
- PRETS (Id_ouv, Id_etu, date_emprunt, date_retour)

Les relations globales sont fragmentées et réparties sur les différents sites comme suit :

- un employé est affecté à un seul site
- un étudiant est inscrit dans une seule université, mais peut emprunter dans toutes les bibliothèques.
- un ouvrage emprunté dans une bibliothèque est rendu dans la même bibliothèque.
- Le champ *emprunts* de la relation ETUDIANT est utilisé pour limiter le nombre

- d'ouvrages empruntés simultanément par un étudiant sur l'ensemble des bibliothèques.
- chaque université gère ses propres étudiants
- chaque bibliothèque gère son personnel et les ouvrages qu'elle possède.

Exercice 1

Donner la définition des différents fragments de l'exemple précédent en utilisant les opérateurs de l'algèbre relationnelle, ainsi que le schéma d'allocation des fragments.

2 Méthode de fragmentation horizontale

Définition Etant donnée une relation $R(A_1, A_2, \dots, A_n)$ un prédicat simple p_j défini sur R est défini:

$p_j : A_i \theta \text{ valeur}$

avec: $\theta \in \{=, <, >, \leq, \geq, \neq\}$; et $\text{valeur} \in \text{domaine}(A_i)$

Algorithme de fragmentation

Entrée : les prédicats simples de la table R

Soit $Pr = \{p_1, p_2, \dots, p_m\}$ un ensemble de prédicats simples définis sur la relation R, l'ensemble de minterms $M = \{m_1, m_2, \dots, m_z\}$ est défini comme suit:

$M = \{m_i \mid m_i = \bigwedge_{p_j \in Pr} p_j^* \}, 1 \leq i \leq z, 1 \leq j \leq m;$

avec $p_j^* = p_j \text{ or } \neg p_j$

Pour appliquer cet algorithme, il faut suivre les étapes suivantes :

- Génération des minterms m_i
- Elimination des minterms contradictoires

Exercice 2

Considérons une table T avec les attributs A et SA et les prédicats simples suivants:

$A < 10, A > 5, \text{Loc} = \text{"SA"}, \text{Loc} = \text{"SB"}$

Donner les fragments horizontaux de la table T

3 Méthode de fragmentation verticale

Le principe de la méthode est de regrouper les attributs fréquemment utilisés ensemble dans un fragment.

Informations d'entrée :

- $Q = \{q_1, q_2, \dots, q_m\}$: ensemble de requêtes fréquentes
- $R (A_1, A_2, \dots, A_n)$: relation R avec n attributs, et l sites
- Matrice d'usage $m \times n$: $MU_{ij} : 1$ si attribut A_j est référencé par q_i sinon 0
- Matrice d'accès $m \times l$: MF_{ik} : fréquence d'accès de la requête i sur le site k
- Matrice de référence $m \times l$: MR_{ik} : le nombre d'accès aux attributs pour chaque exécution de la requête q_i sur le site k

Matrice d'affinité et son rôle pour construire les fragments :

On calcule la matrice d'affinité entre deux attributs A_i et A_j (MA) qui est définie comme suit:

$$MA_{ij} = \sum \{ k | U_{ki} = 1 \wedge U_{kj} = 1 \} \sum MF_{ki}$$

La matrice d'affinité est utilisée pour guider la génération des fragments verticaux, des attributs ayant une grande affinité seront groupés pour former un fragment vertical.

Exercice 3

Quel est le nombre de fragments verticaux possible pour une relation avec m attributs ?

Soit la relation projets(id,nom,budget,location). On a 3 sites, 4 requêtes SQL :

$r1$: SELECT BUDGET FROM PROJETS WHERE ID = v;

$r2$: SELECT NOM, BUDGET FROM PROJETS;

$r3$: SELECT NOM FROM PROJETS WHERE LOCATION = v;

$r4$: SELECT SUM(BUDGET) FROM PROJETS WHERE LOCATION=v

Calculer les fragments verticaux de la table projets.

4 Méthode d'allocation

L'affectation des fragments sur les sites est décidée en fonction de l'origine prévue des requêtes qui ont servi à la fragmentation. Le but est de placer les fragments pour minimiser les transferts de données entre les sites.

Entrées:

- $F = \{F1, F2, \dots, F_n\}$ ensemble de fragments
- $S = \{S1, S2, \dots, S_m\}$ ensemble de sites
- $Q = \{Q1, Q2, \dots, Q_l\}$ ensemble de requêtes

⇒ Le problème d'allocation consiste à trouver une distribution optimale de F sur S afin d'améliorer les performances (les temps de réponse). La méthode bestfit vise à allouer les fragments de telle sorte à maximiser les traitements locaux.

Exercice 4

Donner une allocation des fragments qui maximise les traitements locaux pour le fragmentation qui suit. Pour cela, pour chaque fragment, calculer le nombre total des accès locaux au fragment en fonction de son site de placement (S1, S2, S3, S4, S5).

Fragments	temps d'accès local (ms)	temps mise à jour local	temps accès distant	temps m.a.j. distant
R1	100	150	500	600
R2	150	200	650	700
R3	200	250	1000	1100

Transactions	sites	fréquence	accès tables (L = lecture, E = écriture)
T1	S1,S4,S5	1	4 accès R1 (3L, 1E) 2 accès R2 (2L)
T2	S2,S4	2	2 accès R1 (2L) 4 accès R3 (3L, 1E)
T3	S3,S5	3	4 accès R2 (3L, 1E) 2 accès R3 (2L)