Entrepôt de données et analyse en ligne

M2 SIAD - TD

21 octobre 2009

Le problème consiste en l'étude d'un algorithme de sélection de cuboïdes à matérialiser dans le cas où l'indexation des cuboïdes est prise en compte, et où la charge de l'entrepôt en terme de requêtes OLAP est connue.

Ce TD s'inspire de l'article *Index Selection for OLAP* de Gupta, Harinarayan, Rajaraman et Ullman, ICDE 1997.

1 Modèle de coût

Dans ce qui suit, on appelle structure un cuboïde ou un index. Les cuboïdes seront représentés par un ensemble d'attributs indiquant le niveau de granularité. Pour simplifier, on ne considère que des indexes classique mono-attribut, et on ne tient pas compte des indexes sur les mesures. Les indexes seront représentés par un nom d'attribut.

Coût de stockage On considère que le coût de stockage des structures est le suivant :

- coût de stockage d'un cuboïde V = le nombre de tuples de V,
- coût de stockage d'un index pour l'attribut A sur un cuboïde V = le nombre de tuples de V

Requêtes Le charge de l'entrepôt est donnée sous la forme d'un ensemble de requêtes. Pour simplifier, on ne considère que des requêtes avec groupement, agrégation, et condition de sélection portant sur un unique attribut. Les requêtes sont donc représentées de la manière suivante : $\pi_{A_1,\dots A_n}(\sigma_{A_i})$ où $A_1, \dots A_n$ représente les attributs de groupement et A_i est un attribut pour lequel est exprimé une condition de sélection.

Coût d'évaluation On considère que le coût d'évaluation d'une requête sur le cuboïde V avec une sélection portant sur l'attribut A est le suivant :

- sans index sur l'attribut A = taille de V
- avec un index sur l'attribut A = taille de V / taille du cuboïde A

2 Algorithme de sélection de structures

L'algorithme choisit un ensemble de structures (cuboïdes et/ou indexes) à matérialiser, la condition d'arrêt étant une contrainte sur l'espace disque occupé par la matérialisation de ces structures.

Le choix est guidé par un calcul de bénéfice, qui utilise la fonction suivante :

bénéfice(S,E) calcule le bénéfice qu'il y a à matérialiser la structure S étant donnée un ensemble E de structure matérialisé : bénéfice(S,E) = C(E) - C(E $\cup$ S)

où C est une fonction de calcul de coût, telle que pour un ensemble de structure E, C(E) est calculé comme suit :

1. résultat = 0
2. pour chaque requête q de la charge de l'entrepôt faire
   (a) trouver c le coût minimum de l'évaluation de q sur les structures matérialisées E
   (b) résultat = résultat + c
3. retourner résultat

L'algorithme est donc le suivant :

1. au départ, seul le cuboïde de granularité la plus fine est matérialisé et les étapes suivantes sont répétées jusqu'à saturation de l'espace de stockage
2. pour tout cuboïde non matérialisé et pour tout index non matérialisé d'un cuboïde matérialisé faire
   (a) calculer le bénéfice de matérialisation de chaque structure
   (b) matérialiser la structure de bénéfice le plus important

3 Questions

On considère la table de faits de schéma *ventes(produit, fournisseur, client, montant)* détaillant des montants de vente de produits (P) de différents fournisseurs (F) à des clients (C). Pour simplifier, on considère qu'il n'y a pas de hiérarchie et que chaque dimension est donc composée d'un seul niveau correspondant à l'attribut présent dans la table de faits.

On estime la taille des cuboïdes aux valeurs suivantes (en million de tuples, on ne tiendra pas compte du cuboïde de dimension 0 contenant un seul tuple) :

cuboïde	taille	cuboïde	taille
PFC	6	P	0,2
PF	0,8	F	0,1
PC	6	C	0,01
FC	6		

On considère que la charge de l'entrepôt, en terme de requêtes, est constituée par l'ensemble suivant :

numéro	requête	numéro	requête
1	$\pi_{FC}(\sigma_F)$	4	$\pi_F(\sigma_C)$
2	$\pi_P(\sigma_C)$	5	$\pi_C(\sigma_P)$
3	$\pi_C(\sigma_F)$	6	$\pi_C(\sigma_C)$

1. Indépendamment de la charge, quels sont les cuboïdes que l'on pourrait matérialiser, et quels sont les indexes que l'ont pourrait matérialiser ? Sur quel(s) cuboïde(s) et avec quel(s) index(es) pourrait être évaluée chaque requête de la charge ?
2. Considérons le cas où seul le cuboïde PFC et tous ses indexes sont matérialisés :
   (a) quel est la taille de l'espace de stockage nécessaire ?
   (b) Quel est le coût d'évaluation de chaque requête de la charge ?
3. Dérouler l'algorithme sur l'entrepôt décrit précédemment en supposant que l'espace maximum de stockage est de 14 millions de tuples.
4. Soient les requêtes suivantes, exprimées en SQL sur la table ventes :
   - SELECT fournisseur, client, SUM(montant) FROM ventes WHERE fournisseur='sas' OR fournisseur='ibm';
   - SELECT produit, fournisseur, client, SUM(montant) FROM ventes WHERE produit='processeur' OR produit='mémoire';
   - SELECT produit, fournisseur, SUM(montant) FROM ventes WHERE client='JYA';
   - SELECT client, SUM(montant) FROM ventes WHERE client='JYA' OR client='PM';

   (a) Ces requêtes font-elles partie de la charge de l'entrepôt ?
   (b) En supposant un cube pour lequel ventes est la table des faits et chaque dimension produit, fournisseur, client possède un niveau le plus détaillé et un niveau regroupant toutes les valeurs du niveau le plus détaillé, donner l'expression MDX de chacune de ces requêtes.
   (c) Quel est leur coût d'évaluation respectif si l'on a matérialisé les cuboïdes et indexes trouvés par l'algorithme à la question précédente ?
5. Que faudrait-il changer au modèle si les indexes considérés étaient des indexes bitmap ?