Dépendances fonctionnelles et matérialisation partielle des cubes de données

Eve Garnaud, Sofian Maabout, Mohamed Mosbah

LaBRI. Université de Bordeaux
351, cours de la Libération, 33405 Talence
{garnaud, maabout, mosbah}@labri.fr

Résumé. La sélection de vues à matérialiser dans des entrepôts de données de plus en plus volumineux est une nécessité. Dans cet article, nous montrons qu'il existe un lien très étroit entre recherche des cuboïdes à matérialiser dans un cube de données afin d'optimiser les traitements et les dépendances fonctionnelles sur celui-ci. La contrainte que nous imposons sur les vues que l'on matérialise ne porte pas sur une borne d'espace de stockage à ne pas dépasser comme c'est le cas dans la plupart des travaux relatifs, mais elle porte sur le facteur de performance f que celles-ci vérifient. Nous tentons cependant d'utiliser le moins d'espace mémoire pour atteindre cet objectif. Nous caractérisons formellement toute solution optimale (en terme d'espace mémoire) répondant à ce critère. On prouve que ce problème est NP-difficile et on démontre l'efficacité de nos algorithmes gloutons pour répondre à ce problème en respectant la contrainte de performance fixée par l'utilisateur.

1 Introduction

De nos jours, la quantité d'information stockée par les entreprises est de plus en plus importante ce qui rend le travail d'analyse et d'exploitation de ces informations souvent long et complexe. Une interface sous forme de cube de données permet d'optimiser ces traitements puisqu'il est possible de choisir de ne matérialiser qu'une partie des cuboïdes qui le composent. Le problème est de savoir quelle est la meilleure partie à stocker pour répondre, dans un temps minimum, à toutes les requêtes.

Nous développons donc une technique basée sur les dépendances fonctionnelles pour trouver la meilleure solution à ce problème. Cette solution est optimale dans le sens où, étant donné un facteur de performance $f \geq 1$, on peut répondre à toutes les requêtes posées sans que le coût de calcul de toutes ces requêtes ne dépasse de plus de f fois leur coût de calcul si tous les cuboïdes étaient matérialisés. Il serait trop couteux en terme d'espace mémoire et de coût de maintenance de stocker tous les cuboïdes, c'est pourquoi une sélection s'impose.

Nous prouvons que ce problème de sélection est NP-difficile, cependant, nous sommes en mesure de caractériser précisément, grâce aux dépendances fonctionnelles, les sous-ensembles de cuboïdes solutions et cela, quel que soit le f fixé par l'utilisateur. Cette caractérisation nous assure que la solution apportée est bien de taille minimale par rapport à une autre sélection qui respecterait elle aussi la contrainte f.