Chapitre 8

Conclusion

8.1 Bilan et contributions

Nous nous sommes attachés dans cette thèse à proposer des stratégies d'optimisation de performance des entrepôts de données. Ces stratégies, de sélection d'index et de vues matérialisées, exploitent les techniques de fouille de données en général et la recherche de motifs fréquents et la classification non supervisée en particulier. Nous avons également traité le problème de sélection simultanée d'index et de vues. De plus, nous avons étendu ce problème d'optimisation aux entrepôts de données XML. Dans ce cadre, nous avons proposé un index de jointure XML et adapté notre stratégie de sélection de vues au contexte XML.

Notre travail a été guidé par une étude préalable des travaux existants traitant des problèmes de sélection d'index et de vues matérialisées. Chacun de ces travaux a été étudié selon des critères que nous avons jugés pertinents, à savoir, le formalisme utilisé pour mettre en évidence les relations index-index, vue-vue et index-vue, la manière de déterminer l'ensemble d'index ou de vues candidats, la manière de construire la configuration finale d'index ou de vues, l'utilisation des modèles de coût ou l'appel à l'optimiseur de requêtes et la sélection dans un environnement relationnel ou multidimensionnel. Le travail effectué porte essentiellement sur :

- la généricité de nos solutions, qui ne sont pas inhérentes à un SGBD particulier, afin de pouvoir être ultérieurement utilisées et/ou étendues dans n'importe quel autre SGBD ;
- la modularité de nos stratégies, qui sont adaptables en modifiant un ou plusieurs

modules (par exemple, module de calcul du coût, module de fouille de données, module d'analyse syntaxique, etc.) ;
- l'efficacité de nos propositions, ce qui passe par un processus d'expérimentation et de validation impliquant l'implantation de nos solutions et leur utilisation sur des systèmes existants.

Les principales contributions concernant notre stratégie de sélection d'index résident dans les points suivants.

- La construction des index candidats exploite une certaine connaissance en administration et en optimisation des SGBD, intégrée directement dans l'analyseur syntaxique. Cela produit des index candidats pertinents pour la charge.
- La recherche de motifs fréquents fermés est une bonne heuristique pour réduire l'espace de recherche des index candidats. En effet, cette heuristique cible les attributs fréquemment utilisés ensemble qui sont des bons générateurs des index candidats.
- L'algorithme de recherche des motifs fréquents génère des index mono et multi-attributs à la volée. Cela évite de passer par un processus à plusieurs itérations dans lequel sont créés les index mono-attributs à la première itération, les index à deux attributs à la deuxième itération et ainsi de suite pour les index de taille supérieure.
- Notre stratégie de sélection d'index prend en compte en plus des B-arbre d'autres techniques d'indexation telles que les index *bitmap* de jointure utiles pour réduire les requêtes comportant plusieurs jointures et opérant sur des données volumineuses, typiques dans l'environnement des entrepôts de données.
- Notre stratégie de sélection d'index est modulaire et peut être améliorée ou adaptée en changeant le module adéquat. Par exemple, si nous voulons prendre une nouvelle technique d'indexation, il suffit d'ajouter au module des modèles de coût le mode de calcul des coût correspondant à cette technique et de l'intégrer dans le module de génération d'index à partir des motifs fréquents fermés.
- Notre algorithme de construction de la configuration finale d'index prend en compte l'interaction existant entre les index, au contraire des algorithmes assimilant cette construction au problème du sac à dos ou utilisant les algorithmes génétiques.

Notre stratégie de sélection de vues matérialisées présente quant à elle les caractéristiques novatrices suivantes.

- La construction des vues candidates exploite une certaine connaissance en administration et en optimisation des SGBD, intégrée directement dans l'analyseur syntaxique. Cela produit des vues candidates pertinentes pour les requêtes de la charge.
- Notre stratégie de sélection de vues matérialisées utilise la classification non supervisée afin de construire l'ensemble des vues candidates. Nous avons montré que la classification est une bonne heuristique pour réduire l'espace de recherche des vues candidates.
- Les mesures de similarité et de dissimilarité introduites dans le processus de classification permettent de capturer les relations qui peuvent exister entre les vues candidates.
- La fusion des vues candidates réalisée au niveau des classes obtenues à partir de la classification non supervisée est moins coûteuse que la fusion réalisée directement sur les vues dérivées directement des requêtes de la charge.
- Notre algorithme de construction de la configuration finale de vues matérialisées prend en compte l'interaction existant entre les vues, contrairement aux algorithmes assimilant cette construction au problème du sac à dos ou utilisant les algorithmes génétiques.

Dans notre stratégie de sélection simultanée d'index et de vues matérialisées, nous avons introduit la notion de bénéfice de matérialisation et d'indexation. Ces bénéfice sont introduits dans le but de prendre en compte les interactions index-vues.

Pour terminer, nous avons posé le problème d'optimisation des entrepôts de données XML et avons proposé deux solutions à ce problème. La première concerne un nouvel index XML qui permet de réduire le coût des jointures. La deuxième consiste à adapter notre stratégie de sélection de vues matérialisées se basant sur la classification non supervisée au contexte XML.

8.2 Perspectives de recherche

Le travail réalisé dans cette thèse ouvre diverses perspectives de recherche.

Les stratégies que nous avons proposées s'appliquent sur une charge extraite du système pendant une période de temps jugée suffisante par l'administrateur. Nous sommes alors dans un cas d'optimisation statiques. Il serait donc intéressant de rendre nos stratégies dynamiques et incrémentales. Nous pensons que dans le cas de la sélection d'index, la

recherche incrémentale des motifs fréquents [VMGM02] et celle des motifs fréquents séquentiels [SS05] pourrait garantir l'aspect incrémental et dynamique, respectivement. En effet, cette recherche peut déterminer les motifs fréquents à extraire d'une charge qui évolue dans le temps. Dans le cas de la classification non supervisée, les travaux traitant du problème de classification non supervisée dynamique [ZE99] ou rendant le processus de classification incrémental [JMF99] pourraient être exploités.

Comme nous venons de le préciser, nos stratégies d'optimisation s'appliquent dans un cas statique. La charge sur laquelle est effectuée l'optimisation peut devenir obsolète au bout d'un temps donné. Lorsque cela arrive, il faut resélectionner les index et les vues matérialisées. Nous pensons que les travaux traitant de la détection de sessions basés sur le calcul d'entropie [YHA05] pourraient s'appliquer pour déterminer le moment où il faut lancer la resélection des index et des vues.

Dans les travaux effectués dans cette thèse, nous nous sommes par ailleurs positionnés dans le cas où l'administrateur est le seul utilisateur qui effectue les tâches d'optimisation. L'administrateur optimise alors les requêtes adressées au système par tous les utilisateurs confondus. Or, les besoins définis par différents profils d'utilisateurs (dans le cas des systèmes multi-utilisateurs) sont différents. Il serait donc plus pertinent d'appliquer nos stratégies sur des groupes de requêtes définis par les utilisateurs identifiés dans chaque profil.

Dans cette thèse, nous nous sommes également restreints à utiliser seulement les index et les vues matérialisées comme mécanismes d'optimisation des performances. Or, d'autres mécanismes peuvent être intégrés ou couplés avec les index et les vues. Nous citons à titre d'exemple la gestion de cache, le regroupement et le partitionnement [BBM05].

Finalement, tout le long de cette thèse, nous avons expérimenté nos stratégies sur un entrepôt de données test (cf. Chapitre 4, Section 4.6) contenant des données synthétiques. Nous envisageons d'étendre ces tests sur le banc d'essais d'entrepôts de données DWEB [DBB05] conçu au sein de notre laboratoire. Nous allons également mettre en œuvre nos stratégies sur la plateforme CLAPI, ce qui constituait l'un des objectifs de ce travail. Rappelons que CLAPI est une base de données où sont entreposés des corpus de langue française parlée en interaction. Ces corpus sont composés de données multimédias et XML. Cela offre un terrain d'application sur une échelle réelle des travaux effectués dans cette thèse.