Chapitre 1

Introduction

1.1 Contexte scientifique

L'utilisation courante de bases de données requiert un administrateur qui a pour rôle principal la conception logique ou physique des bases de données, la gestion de l'espace de stockage, le réglage de performance (*tuning*), etc. Avec le déploiement à grande échelle des systèmes de gestion de bases de données, minimiser la fonction d'administration est devenu indispensable [WMHZ02].

L'une des tâches importantes d'un administrateur est la sélection de structures physiques (index, vues matérialisées, etc.) et de politiques (de gestion du cache, de regroupement, de partitionnement, etc.) appropriées susceptibles d'améliorer les performances du système en minimisant les temps d'accès aux données [FST88]. L'optimisation automatique de ces tâches donne lieu à des systèmes dits auto-administratifs. Ces systèmes ont pour objectif de s'administrer et de s'adapter eux-mêmes, automatiquement, sans perte ou même avec un gain de performance.

Depuis quelques années, l'idée d'utiliser les techniques de fouille de données (*data mining*) pour extraire des connaissances utiles des données elles-mêmes pour leur administration est avancée [Cha98]. Cependant, peu de travaux ont été entrepris dans cette optique. C'est pourquoi nous étudions dans cette thèse l'utilisation des techniques de fouille de données pour l'optimisation automatique des performances des entrepôts de données. Nous étudions en particulier le problème de sélection d'index et de vues matérialisées dans les entrepôts

de données relationnels et XML. Ces problèmes d'optimisation sont NP-complets [Com78, Gup99] . De ce fait, il n'existe pas d'algorithme qui propose une solution optimale en un temps raisonnable.

1.2 Contexte d'application

Cette thèse a été financée dans le cadre de l'ACI TTT (Action Concertée Incitative "Terrains, Techniques, Théories"), fruit d'une collaboration entre les laboratoires ICAR (Interactions, Corpus, Apprentissages, Représentations) de l'Université Lyon 2, ERIC (Équipe de Recherche en Ingénierie des Connaissances) de l'Université Lyon 2 et RIM (Réseaux, Information, Multimédia) de l'École Nationale Supérieure des Mines de Saint-Étienne.

L'objectif de ce projet était de rechercher de nouvelles méthodes d'exploration des données en linguistique, plus particulièrement, des corpus de français parlé en interaction, et mettre en place une plateforme logicielle multimédia accessible sur le Web. Le projet a donné lieu à la réalisation de l'application CLAPI (Corpus de LAngue Parlée en Interaction) [ABBD03], qui permet :

- la gestion d'un nombre important de corpus/unités documentaires (enregistrements audios et vidéos, transcriptions[1], etc.) ;
- l'interrogation par des requêtes portant sur les descripteurs et sur le contenu des transcriptions, enrichies par des balises (recherche sur des chaînes de caractères, mais aussi sur la temporalité des phénomènes balisés) ;
- la consultation et le téléchargement des corpus gérés par des droits d'accès sécurisés.

L'application CLAPI est exploitée principalement par des non-informaticiens. Il est donc indispensable de garantir les performances du système de façon automatique ou semi-automatique. Cette base offre un terrain d'application des travaux effectués dans cette thèse. En effet, les requêtes définies par les utilisateurs peuvent être coûteuses car les objets entreposés dans CLAPI sont volumineux. Nos stratégies de sélection d'index et de vues matérialisées peuvent donc être appliquées. De plus, les transcriptions sont stockées en XML. Nos stratégies d'optimisation dans le contexte XML peuvent donc être également exploitées.

[1]Une transcription est une reproduction écrite du signal audio ou vidéo selon une convention de transcription.

1.3 Objectifs et contributions

Notre objectif principal consiste à fournir des stratégies qui permettent d'optimiser les performances des entrepôts de données. Le cœur de ces stratégies repose sur des techniques de fouille de données. Ces techniques sont employées comme des heuristiques qui aident à réduire la complexité des problèmes de sélection d'index et de vues matérialisées. Ces structures permettent un accès direct aux données et jouent un rôle particulièrement important dans les bases de données décisionnelles (BDD) telles que les entrepôts de données, qui présentent une volumétrie très importante et sont interrogés par des requêtes complexes.

Notre travail s'articule donc autour de trois axes principaux :

- le développement d'une stratégie pour la sélection d'index,
- la proposition d'une stratégie de sélection de vues matérialisées,
- la sélection simultanée d'index et de vues matérialisées.

Plusieurs travaux de recherche ont traité le problème de sélection d'index et de vues matérialisées. Cependant, ces travaux ne prennent pas en compte les connaissances (métadonnées, statistiques, charge de requêtes appliquée au système, usage des attributs de l'entrepôt de données dans ces requêtes, etc.) qui peuvent être extraites de la charge afin de réduire la complexité du problème de sélection et de cibler les index et les vues candidats les plus pertinents. Les stratégies que nous proposons intègrent ces connaissances dans le processus d'optimisation. En effet, notre stratégie de sélection d'index exploite la recherche des motifs fréquents fermés afin de cibler l'ensemble des index candidats. Notre intuition est que l'utilité d'un index donné est fortement corrélée avec la fréquence d'utilisation des attributs associés à cet index dans l'ensemble des requêtes de la charge. Notre stratégie de sélection de vues matérialisées quant à elle utilise la classification non supervisée afin de construire un ensemble des vues candidates. L'idée d'utiliser la classification est motivée par le fait que plusieurs requêtes ayant une syntaxe similaire sont susceptibles d'être résolues à partir d'une vue matérialisée dont la syntaxe est également proche de celle des requêtes.

Il nous a par ailleurs paru intéressant d'adapter ces stratégies au contexte des entrepôts de données XML. Actuellement, les applications décisionnelles exploitent de plus en plus de données hétérogènes et provenant de sources variées. Dans ce contexte, XML peut aider grandement à l'intégration et à l'entreposage de données en vue de fouille ou d'analyse en

ligne. Cependant, les requêtes décisionnelles sont généralement complexes du fait qu'elles impliquent de nombreuses jointures et agrégations. Par ailleurs, les systèmes natifs XML présentent des performances médiocres quand le volume des données est important ou que les requêtes sont complexes. Il est donc crucial lors de la construction d'un entrepôt de données XML de garantir les performances des requêtes qui l'exploiteront.

1.4 Organisation de la thèse

La reste de ce mémoire est organisé comme suit. Le Chapitre 2 présente les principales techniques d'indexation utilisées dans les systèmes relationnels et natifs XML, ainsi que le principe de la matérialisation des vues. Ce chapitre ne prétend pas faire une présentation exhaustive de toutes les techniques d'indexation et de matérialisation de vues, ce qui sortirait du cadre de nos travaux. Nous nous attachons cependant à décrire en détail le fonctionnement de ces techniques dans le processus d'optimisation des requêtes, puisque nos stratégies d'optimisation reposent sur cette connaissance.

Le Chapitre 3 traite des problèmes de sélection d'index et de vues matérialisées, ainsi que de la sélection simultanée de ces structures. Nous y détaillons aussi les principaux travaux proposés pour résoudre ces problèmes. Nous proposons également une classification de ces travaux selon divers critères que nous avons jugés pertinents.

Le Chapitre 4 présente notre stratégie de sélection d'index basée sur la recherche de motifs fréquents fermés. Nous abordons en détail les motivations qui nous ont poussées à choisir cette technique de fouille de données, présentons les étapes de notre stratégie, ainsi que les résultats expérimentaux que nous avons obtenus.

Le Chapitre 5 développe point par point notre stratégie de sélection de vues matérialisées basée sur la classification non supervisée. Nous abordons également ici les motivations qui nous ont amenées à choisir la classification comme heuristique permettant de réduire la complexité du problème de sélection, ainsi que nos résultats expérimentaux.

Le Chapitre 6 expose notre stratégie de sélection simultanée d'index et de vues matérialisées. Nous y montrons que la sélection simultanée offre de meilleures performances que la sélection isolée des index et des vues.

Le Chapitre 7 traite du problème de l'optimisation des performances des entrepôts de

données XML. Nous présentons notre index de jointure XML, qui améliore les performances des requêtes XQuery grâce au précalcul des jointures, ainsi que l'adaptation de notre stratégie de sélection de vues matérialisées au contexte XML.

Le Chapitre 8 conclut ce mémoire en dressant le bilan de nos contributions et présente les perspectives de recherche qui en découlent.