

Entrepôts de données sur grilles de calcul

Pascal Wehrle, Maryvonne Miquel, Anne Tchounikine
LIRIS UMR 5205
INSA, Campus de la Doua,
Bâtiment Blaise Pascal (501), 20, avenue Albert Einstein
69621 VILLEURBANNE CEDEX
<prenom>.<nom>@insa-lyon.fr
<http://liris.cnrs.fr>

1 Introduction

L'objectif d'un entrepôt de données est de mettre à disposition des outils d'aide à la décision à partir de grands volumes de données produits par des systèmes d'informations de production (Inmon 1992). Les « dimensions » du modèle multidimensionnel représentent les axes d'analyse et sont hiérarchisées en niveaux de détail. Les données sont modélisées sous forme d'hypercubes navigables grâce aux outils OLAP (On Line Analytical Processing). La structure interne classique d'un entrepôt de données est celle du schéma en étoile, introduit par Kimball (Kimball 1996). Celui-ci est constitué d'une table de faits centrale contenant les données les plus détaillées de l'entrepôt, appelées « faits » ou « mesures ». Celles-ci sont associées via des clés étrangères à des tables de dimension accueillant les données concernant les axes d'analyse. Afin d'améliorer les temps de réponse aux requêtes, des agrégats comme par exemple la somme ou la moyenne sur les faits sont pré-calculés au sein de l'entrepôt.

Les besoins croissants en termes de capacité de traitement et de stockage causés par la conception et l'exploitation d'entrepôts de données de plus en plus complexes et volumineux par exemple dans le secteur geno-médical (Brunie et al. 2003) favorisent l'utilisation de systèmes distribués puissants. Le concept récent des grilles de calcul fournit une approche décentralisée à la construction d'infrastructures à hautes performances efficaces, économiques et extensibles dont les principes de base sont exposés par Foster (Foster 2003). Leurs services de gestion et d'information mettent à disposition un accès transparent à un grand nombre de ressources hétérogènes distantes dans le but d'offrir à l'utilisateur une qualité de service « non triviale ».

Le modèle d'architecture proposé dans cet article a pour objectif l'intégration d'un entrepôt de données sur une infrastructure de grille de calcul. Les avantages principaux d'un déploiement de grands volumes de données détaillées et de leurs agrégats sur une grille sont les possibilités de traitement et d'accès parallèles, de stockage et d'échange décentralisés des données ou résultats de requêtes. Du côté utilisateur, l'objectif est de proposer un service d'entrepôt aux spécialistes connectés à différents points d'accès de la grille.

2 Identification et fragmentation des données

L'entrepôt de données doit être entièrement réparti parmi les noeuds de la grille afin de s'adapter à l'infrastructure de grille de calcul et de permettre une gestion et un accès décentralisé. Pour faciliter la recherche et l'échange de données entre noeuds de la grille nous introduisons une méthode d'identification unique et globale des données de l'entrepôt.

Des identifiants uniques pour les données les plus détaillées sont facilement trouvés grâce aux membres de dimension directement associés aux faits. Il est important pour une gestion efficace des ensembles de données de pouvoir ordonner et comparer ces identifiants.

2.1 Chunks de base et chunks agrégés

Nous introduisons une relation d'ordre total sur l'ensemble M_{i0} de membres de dimension au niveau le plus détaillé (niveau 0) pour chaque dimension D_i . Nous supposons qu'un tel ordre peut être trouvé pour chaque dimension. Cet ordre total est propagé à travers la hiérarchie entière et l'on obtient une notion d'intervalle $R_{ij} = [m_i, m_s]$ sur les M_{ij} pour tous les niveaux hiérarchiques j . Ces propriétés facilitent le choix d'une unité de base pour les données détaillées et agrégées. La notion de « chunk » de données est originaire de systèmes de mémoires cache de résultats de requêtes coté client proposés par Deshpande et al. (Deshpande et al. 1998) et Kalnis et al. (Kalnis et al. 2002). Nous définissons :

Définition 1 – Un **chunk de base** comme une combinaison unique de membres de dimensions de niveau de hiérarchie le plus bas (niveau 0) pour chaque dimension, et des faits associés.

Définition 2 – Un **chunk agrégé** comme une combinaison unique de membres de dimensions de niveau de hiérarchie i où $i > 0$ pour au moins une dimension, et des faits agrégés associés.

On peut identifier un chunk par le tuple $c = \langle \langle m_1, \dots, m_n \rangle, f_1, \dots, f_q, [L_a] \rangle$ où m_i est un membre de la dimension D_i , f_i est un fait ou un fait agrégé, et L_a la liste des fonctions d'agrégation pour les chunks agrégés.

2.2 Application au partitionnement horizontal

Le partitionnement horizontal d'une base de données relationnelle comme le décrit Özsu (Özsu 1991) consiste à diviser une table de grande taille en plusieurs ensembles de données adaptées à répondre aux requêtes d'un certain groupe d'utilisateurs. Bellatreche et al. (Bellatreche et al. 1999) élargit cette approche au partitionnement de schémas en étoile, créant ainsi plusieurs instances appelées « fragments » du schéma en étoile. Les cuboïdes OLAP agrégés peuvent également être fragmentés selon cette méthode.

Définition 3 – **bloc de chunks**: Un bloc de chunks multidimensionnel B est un sous-ensemble contigu des données de l'entrepôt. Il est représenté par un intervalle R_i pour chaque dimension et contient tous les chunks $c = \langle \langle m_1, \dots, m_n \rangle, f_1, \dots, f_q, [L_a] \rangle$ pour lesquels $m_i \in R_i$.

Définition 4 – **fragment horizontal**: Un fragment horizontal $HF = \{B_1, \dots, B_p\}$ d'un schéma en étoile ou d'un cuboïde OLAP est un ensemble contenant des blocs de chunks avec une liste de niveaux hiérarchiques commune. Ces fragments sont identiques aux fragments du schéma en étoile produits par la méthode originale et incluent en plus les données agrégées des cuboïdes OLAP de l'entrepôt.

3 Indexation de l'entrepôt de données

Une structure d'indexation spécialisée permet de connaître les données qui sont localement disponibles. Tous les noeuds de grille participant à l'exploitation de l'entrepôt de données hébergent une instance de cette structure et doivent disposer de l'ensemble des informations sur les instances de dimensions et leurs hiérarchies. Nous utilisons une indexation spatiale basée sur des arbres X (Berchtold et al. 1996) combinée avec une structure de treillis.

3.1 Indexation des fragments de l'entrepôt

Le treillis de cuboïdes forme un graphe où un noeud représente chaque combinaison de niveaux hiérarchiques et les arêtes une opération d'agrégation sur une dimension d'un niveau hiérarchique vers son parent direct. Les fragments matérialisés sur un noeud de la grille sont d'abord regroupés en fragments correspondants aux listes de niveaux hiérarchiques

communes. Pour chaque liste, un noeud de treillis est créé et relié aux autres par les arêtes appropriées. Le résultat est une instance partielle du treillis maximum permettant un accès rapide aux catégories de données de l'entrepôt disponibles localement.

Les blocs de chunks contenus dans un fragment sont décrits par des intervalles sur des ensembles de membres de dimension M_i pour chaque dimension D_i . Nous assimilons les blocs de chunks à des hyperrectangles dans un espace de données multidimensionnel formé par le produit cartésien $M_1 \times \dots \times M_n$ des ensembles de membres de dimension. Les arbres X utilisés comme structure d'indexation spatiale limitent le nombre de chevauchements entre sous-arbres et sont donc particulièrement bien adaptés aux espaces de données avec un grand nombre de dimensions.

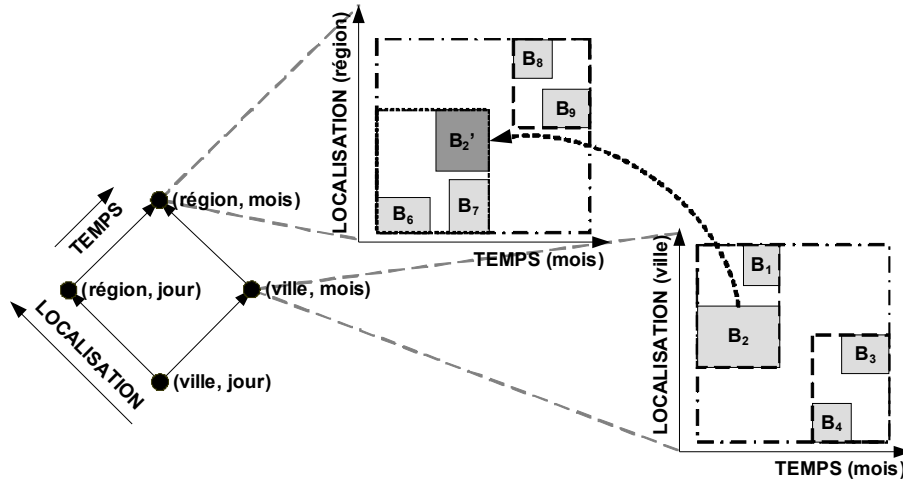


FIG 1 – structure d'indexation spécialisée pour les fragments de l'entrepôt réparti

3.2 Intégration de données calculables et traitement des requêtes

Une fois les fragments matérialisés sur site indexés, il est également possible d'intégrer des informations sur les agrégats calculables à partir de données matérialisées localement. Un agrégat ne peut être calculé qu'à partir de données indexées par un noeud du treillis de niveau hiérarchique inférieur. Le mécanisme permettant de déterminer si un bloc de chunks est calculable à partir de données localement matérialisées est basé sur un compteur virtuel introduit par Deshpande (Deshpande 2000) qui contient pour chaque chunk le nombre de chemins dans le treillis par lesquels ce chunk peut être calculé.

Le traitement des requêtes OLAP consiste à construire des plans d'exécution en associant les résultats fournis par notre structure d'indexation avec les capacités des services de grilles. La grille doit alors exécuter des tâches de chargement et de calcul sur les données matérialisées localement et rechercher les données manquantes sur d'autres nœuds de la grille via un service de requête distribué comme celui proposé par Smith et al. (Smith et al. 2002).

4 Conclusion

Nous avons introduit un modèle de données pour la gestion d'un entrepôt de données réparti adapté à la nature décentralisée et hétérogène des grilles de données. Ce modèle est inspiré par les méthodes existantes pour la gestion d'entrepôts de données répartis. L'unité élémentaire de données appelée « chunk » a été introduite pour former une base pour

l'indexation globale sur l'ensemble de la grille. Les chunks de données détaillées et agrégées permettent de constituer des blocs contigus et des fragments entiers de l'entrepôt de données. Notre structure d'indexation spécialisée associe une structure de treillis pour l'indexation des fragments avec une indexation spatiale en arbre X qui gère l'espace multidimensionnel interne aux différents fragments. En plus des données matérialisées localement, cette structure intègre les agrégats calculables à partir de celles-ci. Le résultat est une méthode efficace pour l'évaluation de requêtes OLAP qui profite de la répartition des données et du potentiel de calcul d'agrégation à la demande.

Références

- Bellatreche L., Karlapalem K. et Mohania M. (1999), OLAP Query Processing for Partitioned Data Warehouses, 1999 International Symposium on Database Applications in Non-Traditional Environments (DANTE '99), 1999, IEEE Computer Society, pp 35-42
- Berchtold S., Keim D. A. et Kriegel H.-P. (1996), The X-tree : An Index Structure for High-Dimensional Data, VLDB'96, Proceedings of 22th International Conference on Very Large Data Bases, 1996, Morgan Kaufmann Publishers/Elsevier Science, pp 28-39
- Brunie L., Miquel M., Pierson J.-M., Tchounikine A., Dhaenens C., Melab N., Talbi E.-G., Hameurlain A. et Morvan F. (2003), Information grids: managing and mining semantic data in a grid infrastructure; open issues and application to geno-medical data, 14th International Workshop on Database and Expert Systems Applications (DEXA'03), 2003, IEEE Computer Society, pp 509-516
- Deshpande P. M., Ramasamy K., Shukla A. et Naughton J. F. (1998), Caching multidimensional queries using chunks, Proceedings of SIGMOD 1998, ACM SIGMOD International Conference on Management of Data, 1998, ACM Press, pp 259-270
- Deshpande P. et Naughton J. F. (2000), Aggregate Aware Caching for Multi-Dimensional Queries, Proceedings of Advances in Database Technology - EDBT 2000, 7th International Conference on Extending Database Technology, 2000, Springer, pp 167-182
- Foster I. et Kesselman C. (2003), The Grid 2: Blueprint for a New Computing Infrastructure, Morgan Kauffmann, 2003, ISBN 1-55860-933-4
- Inmon W.H. (1992), Building the Data Warehouse, John Wiley&Sons, 1992, ASIN: 0471569607
- Kalnis P., Ng W. S., Ooi B. C., Papadias D. et Tan K.-L. (2002), An adaptive peer-to-peer network for distributed caching of OLAP results, Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data, 2002, ACM Press, pp 25-36
- Kimball R. (1996), The Data Warehouse Toolkit, John Wiley&Sons, 1996
- Özsu M. T. et Valduriez P. (1991), Principles of distributed database systems, Prentice Hall, 1991, ISBN 0-13-715681-2
- Smith J., Gounaris A., Watson P., Paton N. W., Fernandes A. A. A. et Sakellariou R. (2002), Distributed Query Processing on the Grid, Proceedings of Grid Computing - GRID 2002, Third International Workshop, 2002, Springer, pp 279-290