



**HAL**  
open science

# Optimisation de requêtes dans les systèmes d'intégration de données

Abdeslem Belghoul

## ► To cite this version:

Abdeslem Belghoul. Optimisation de requêtes dans les systèmes d'intégration de données. BDA 2014 : Gestion de données - principes, technologies et applications, Oct 2014, Autrans, France. pp.24-25. hal-01169948

**HAL Id: hal-01169948**

**<https://inria.hal.science/hal-01169948>**

Submitted on 30 Jun 2015

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial - NoDerivatives 4.0 International License

# Optimisation de requêtes dans les systèmes d'intégration de données

BELGHOUL Abdeslem  
Université Blaise Pascal, LIMOS, CNRS  
Clermont-Ferrand, France  
belghoul@isima.fr

## ABSTRACT

Le concept d'intégration de données est conçu initialement pour fédérer et unifier l'accès aux diverses sources de données. La grande partie des travaux menés dans ce domaine a porté sur le traitement des problèmes d'architecture des systèmes d'intégration de données et de réécriture de requêtes (GAV, LAV, GLAV, etc.). Une réécriture de requêtes peut être vue comme un plan logique de requête auquel plusieurs plans physiques d'exécution peuvent correspondre. Si le problème de réécriture de requêtes a été intensivement étudié dans la littérature, la génération de plans physiques à partir des réécritures est encore mal maîtrisée. Nous nous intéressons, dans notre travail de recherche, au problème d'optimisation pour l'évaluation de requêtes dans les systèmes d'intégration de larges volumes de données.

## 1. INTRODUCTION

Le concept d'intégration de données est conçu pour fédérer et unifier l'accès à des sources de données hétérogènes. Les différents travaux de recherche menés dans ce domaine [11, 8, 4, 10] ont proposé les principaux composants de l'architecture des systèmes d'intégration de données  $I$ . Ces systèmes sont constitués d'un schéma global  $G$ , d'un ensemble de schémas de sources de données  $S$  et d'un ensemble de liens (mapping)  $M$  reliant les relations du schéma  $G$  aux relations des schémas de  $S$ .

Pour la définition des liens  $M$ , deux principales approches ont été proposées [8, 4]. La première, appelée *Global As View (GAV)*, consiste à décrire chaque élément du schéma global  $G$  à travers les schémas des sources de données de  $S$ . Quant à la deuxième approche, appelée *Local As View (LAV)*, consiste à définir les éléments des sources de données de  $S$  à travers les relations du schéma global  $G$ . Le mécanisme le plus utilisé pour matérialiser l'ensemble des liens  $M$  est celui des vues (*views*).

La difficulté dans les systèmes d'intégration de données réside dans l'évaluation de requêtes en utilisant les liens de  $M$ . Cette difficulté se trouve dans l'identification des vues

pertinentes et la génération de l'ensemble des plans logiques possibles contenant le maximum de réponse [5, 6, 9].

## 2. PROBLÉMATIQUE

La problématique générale abordée a trait à l'optimisation de requêtes dans les systèmes d'intégration de larges volumes de données. Il est constaté que la grande partie des travaux menés dans ce domaine a traité les problèmes liés à l'architecture des systèmes (GAV, LAV, GLAV, etc.) et la réécriture de requêtes [11, 8, 4, 10, 5]. Cette réécriture, peut être perçue comme un plan logique de requête, pouvant être exécutée concrètement de plusieurs manières correspondant à différents plans physiques possibles. Ces derniers dépendent de plusieurs paramètres, tels que, l'optimiseur, les méthodes d'accès aux données, les algorithmes de jointures implémentés, etc.

Notre travail de recherche vise, d'une part, la définition d'une approche permettant de générer les différents plans physiques possibles, et d'autre part, l'élaboration d'un modèle de coût permettant de choisir le plan physique efficient. Les paramètres importants dudit modèle devraient être identifiés dans le cadre de ce travail de thèse.

## 3. ANALYSE DES PREMIERS RÉSULTATS

Dans l'objectif d'effectuer une première analyse sur le fonctionnement de système d'intégration de données et l'identification des paramètres impactant les performances d'exécution de requêtes, une plate-forme expérimentale a été mise en place, selon l'approche (GAV). Elle est composée de trois machines virtuelles munies de système de gestion de base de données (SGBD) dont deux hébergeant les données et la troisième jouant le rôle de médiateur.

Notre travail est inscrit dans le cadre du projet intitulé "Petasky" (<http://com.isima.fr/Petasky>), qui est défini et soutenu dans le cadre du défi "Grandes masses de données scientifiques" de la mission interdisciplinarité du CNRS-France. Le jeu de données utilisé provient du projet "Large Synoptic Survey Telescope" (LSST : <http://www.lsst.org/lst/>). Ce jeu de données est d'une taille estimée à 90 Go et d'un nombre d'enregistrements estimé à 170 millions répartie sur deux tables dans deux serveurs. Un protocole de tests a été élaboré afin d'étudier le fonctionnement du système d'intégration et d'évaluer les performances d'exécution de requêtes de jointure.

L'analyse des premiers résultats fait ressortir les éléments suivants :

- Le médiateur décompose la requête en un ensemble de

(c) 2014, Copyright is with the authors. Published in the Proceedings of the BDA 2014 Conference (October 14, 2014, Grenoble-Autrans, France). Distribution of this paper is permitted under the terms of the Creative Commons license CC-by-nc-nd 4.0.

(c) 2014, Droits restant aux auteurs. Publié dans les actes de la conférence BDA 2014 (14 octobre 2014, Grenoble-Autrans, France). Redistribution de cet article autorisée selon les termes de la licence Creative Commons CC-by-nc-nd 4.0.

BDA 14 octobre 2014, Grenoble-Autrans, France.

sous-requêtes qui sont envoyées aux sources de données concernées. Aussi, il pousse la sélection et la projection au niveau des sources distantes. Il collecte les réponses et applique un algorithme de jointure.

- La variation des paramètres liés à l'espace mémoire du SGBD, utilisé par le médiateur, a permis d'observer des changements dans les performances d'exécution sans le changement de l'algorithme de jointure appliqué. Aussi, il a été observé que le médiateur n'utilise pas le cache mémoire du SGBD.
- La variation des paramètres liés à la couche réseaux du SGBD, au niveau du médiateur, a permis de constater des changements dans les performances d'exécution.
- Le médiateur est sensible à la façon de définir les liens entre les relations du schéma global et les relations des sources distantes. Les cas suivants ont été observés :
  - Dans une liaison de type une vue (relation globale) pour une et une seule relation de source distante, le médiateur interroge la deuxième source distante autant de fois que le nombre d'enregistrements de la première source. Dans cas, l'algorithme de jointure appliqué est appelé bind-join qui affiche une exécution non optimale de la requête.
  - Dans une liaison de type une vue (relation globale) composée au moins deux relations distantes et l'autre composé d'une seule relation distante, le médiateur interroge la deuxième source distante autant de fois que le nombre d'enregistrements de l'autre source. Ce plan d'exécution est le même que le plan précédent.
  - Dans une liaison de type une vue reliée à au moins à deux relations distantes, le médiateur matérialise les données de la première relation globale sur l'espace mémoire de travail et commence à lire et faire la jointure des enregistrements de la deuxième relation globale en appliquant le Hash-Join algorithme.
  - Dans une liaison de type une vue pour une et une seule relation et les vues résidant dans le même site, le médiateur pousse la jointure vers le site contenant les deux relations qui aura le choix de définir le plan physique d'exécution de la requête.

#### 4. CONCLUSIONS ET PERSPECTIVES :

Des premiers résultats obtenus, il convient de dire que les systèmes d'intégration actuels ne sont pas adaptés à la gestion de large volumes de données. Le mode d'exécution d'une requête de jointure dans un environnement distribué est plus complexe que l'exécution d'une requête de jointure en local. En effet, l'optimisation de l'exécution d'une requête de jointure en local repose sur le choix de l'ordonnement des relations, la méthode d'accès et l'algorithme de jointure à appliquer [1]. Quant à l'exécution dans un système d'intégration de données, en plus des éléments d'optimisation d'une requête de jointure en local, elle dépend notamment de l'environnement distribué, à titre d'exemple :

- A quel moment faut-il réaliser la jointure au niveau du médiateur ou au niveau du site distant ?

- Quel type d'algorithme de jointure et de méthode d'accès faut-il appliquer ?
- Quel type de fonction objective faut-il appliquer ?

Dans ce contexte, notre travail de recherche vise l'étude des différentes techniques d'optimisation de requêtes distribuées [3, 7]. Les auteurs de [2] considèrent que la gestion et l'évaluation de requêtes dans les systèmes d'intégration de larges volumes de données est un défi.

Pour notre travail qui s'appuie sur [3, 7, 2], il s'agit : (i) d'identifier les différents opérateurs physiques utiles pour la construction de l'espace des solutions, (ii) d'identifier les paramètres impactant les performances d'exécution d'un plan physique de requête et l'élaboration d'un modèle de coût adapté, et (iii) la définition de stratégie d'optimisation qui s'appuie sur les éléments précédents pour générer des plans physiques avec des coûts d'exécution raisonnables.

#### 5. REFERENCES

- [1] S. Chaudhuri. An overview of query optimization in relational systems. In *In PODS*, pages 34–43, 1998.
- [2] X. L. Dong and D. Srivastava. Big data integration. *Proc. VLDB Endow.*, 6(11):1188–1189, Aug. 2013.
- [3] L. M. Haas, D. Kossmann, E. L. Wimmers, and J. Yang. Optimizing queries across diverse data sources. In *Proceedings of the 23rd International Conference on Very Large Data Bases, VLDB '97*, pages 276–285, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc.
- [4] A. Halevy, A. Rajaraman, and J. Ordille. Data integration: The teenage years. In *Proceedings of the 32nd International Conference on Very Large Data Bases, VLDB '06*, pages 9–16. VLDB Endowment, 2006.
- [5] A. Y. Halevy. Answering queries using views: A survey. *The VLDB Journal*, 10(4):270–294, Dec. 2001.
- [6] T. Kirk, A. Y. Levy, Y. Sagiv, and D. Srivastava. The information manifold. In *In Proceedings of the AAAI 1995 Spring Symp. on Information Gathering from Heterogeneous, Distributed Environments*, pages 85–91.
- [7] D. Kossmann. The state of the art in distributed query processing. *ACM Comput. Surv.*, 32(4):422–469, Dec. 2000.
- [8] M. Lenzerini. Data integration: A theoretical perspective. In *Proceedings of the Twenty-first ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS '02*, pages 233–246, New York, NY, USA, 2002. ACM.
- [9] A. Y. Levy, A. Rajaraman, and J. J. Ordille. Querying heterogeneous information sources using source descriptions. In *Proceedings of the 22th International Conference on Very Large Data Bases, VLDB '96*, pages 251–262, San Francisco, CA, USA, 1996. Morgan Kaufmann Publishers Inc.
- [10] J. D. Ullman. Information integration using logical views. In *Proceedings of the 6th International Conference on Database Theory, ICDT '97*, pages 19–40, London, UK, UK, 1997. Springer-Verlag.
- [11] G. Wiederhold. Mediation in information systems. *ACM Comput. Surv.*, 27(2):265–267, June 1995.