Interrogation de Polystores hétérogènes multi-modèles à partir de Modèles Unifiés

Léa EL AHDAB[1], André PENINOU[2], Olivier TESTE[2], Imen MEGDICHE[3]

1. Université de Toulouse UT3, IRIT, Toulouse, 118 Route de Narbonne, F-31062 Toulouse, France, lea.el-ahdab@irit.fr
2. Université de Toulouse UT2J, IRIT, Toulouse, 118 Route de Narbonne, F-31062 Toulouse, France, andre.peninou@irit.fr, olivier.teste@irit.fr
3. INU JF Champollion, IRIT, Avenue Georges Pompidou, F-81104 Castres, France, imen.megdiche@irit.fr

Cet article est une synthèse de l'article : El Ahdab, L., Teste, O., Megdiche, I., & Péninou, A. (2023, August). Unified views for querying heterogeneous multi-model polystores. In International Conference on Big Data Analytics and Knowledge Discovery, DAWAK 2023 (pp. 319-324). Cham : Springer Nature Switzerland.

La problématique de cet article porte sur l'interrogation transparente de données distribuées de manière verticale dans des polystores. Systèmes de stockage flexibles, ils associent plusieurs bases de données SQL et NoSQL. Plusieurs solutions se développent en considérant la distribution verticale qui consiste à stocker des classes d'entités dans plusieurs bases de données différentes qui peuvent être de différents modèles. Certaines solutions se concentrent sur la représentation unifiée du polystore et proposent la migration des données vers un modèle unifié (cf. Barret *et al*., 2022). D'autres travaillent sur les opérateurs et utilisent des fonctions externes aux SGBD pour réaliser des jointures entre les systèmes (cf. Kolev *et al.*, 2016). L'hétérogénéité structurelle est peu considérée dans les systèmes NoSQL des polystores, de même que la fragmentation verticale des données qui correspond au stockage d'une classe d'entité dans différentes bases de données. La notion de clé de fragmentation est utilisée : c'est un attribut clé primaire présent dans l'ensemble des fragments identifiés et permettant de reconstruire l'entité fragmentée.

Cet article propose un Framework d'interrogation des polystores composés de bases de données Relationnelles (R) et Document (D). Il propose une représentation unifiée du polystore masquant l'hétérogénéité des modèles de données, la distribution et la fragmentation des données. Il se compose d'une phase de construction qui met en place les outils nécessaires à l'interrogation du polystore et d'une phase d'exploitation qui contient les étapes liées à la réécriture de la requête utilisatrice. *Phase de construction*. Elle se base sur des modèles unifiés par système (R ou D) correspondant à la traduction du modèle E/R du polystore dans ce système.

Chaque classe d'entité correspond à une table (R) ou à une collection (D). Les relations, selon leur cardinalité, peuvent correspondre à une table, à un attribut dans une table/collection (clé étrangère) ou à des attributs imbriqués dans les collections. Le polystore est alors représenté soit sous la forme de tables (R), soit de collections (D). Un dictionnaire de mapping fait le lien entre la position des attributs dans les modèles unifiés et leur position réelle dans le polystore. Un attribut dans un modèle unifié est associé à une table/collection. Dans le polystore, ce même attribut est associé à une base de données et à une table/collection. *Phase d'exploitation*. L'utilisateur écrit une requête *Q* en SQL (opérateurs : σ, π, ⋈) sur le modèle unifié relationnel ou en MongoDB (opérateurs : $match, $project) sur le modèle unifié document. *Q* est analysée algébriquement et le dictionnaire de mapping permet de réécrire *Q* avec la position réelle des attributs (Q_{new}). Des opérations sont ajoutées (jointure relationnelle ou lookup et unwind en document) dans le cas de la reconstruction d'une entité fragmentée ou dans le cas de changement de systèmes (transfert et transformation). Le résultat de l'exécution de Q_{new} est renvoyé à l'utilisateur dans le format du modèle interrogé.

Pour les expérimentations, nous avons utilisé les données du dataset Unibench où les classes d'entités sont distribuées verticalement et certaines sont fragmentées. Plusieurs types de requête sont considérés selon les opérateurs (σ, π, σ+π) et le nombre de tables (mono-table, multi-tables). L'opérateur de jointure est associé aux requêtes multi-tables. Notre évaluation compare le temps de réécriture des requêtes sur les modèles unifiés (relationnel et document) en fonction de plusieurs distributions. Le temps de réécriture moyen des requêtes est d'environ 0.0041 secondes et leur temps d'exécution moyen est d'environ 10 secondes. Les modèles unifiés permettent de cacher la complexité du polystore en le représentant dans un format mono-store. L'utilisateur choisit donc le langage d'écriture de sa requête. La réécriture des requêtes est peu coûteuse par rapport au coût d'exécution qui dépend de la distribution et fragmentation des données. Des structures hétérogènes peuvent apparaître dans les systèmes de stockage document et notre système les prendra en compte dans la réécriture de *Q* à partir de travaux existants.

Remerciements :

Ces travaux ont été menés avec le soutien du Gouvernement Français dans le cadre du programme Territoire d'Innovation, une action du Grand Plan d'Investissement adossé à France 2030, de Toulouse Métropole et du GIS neOCampus.

Bibliographie

Barret N., Manolescu I., Upadhyay P. (2022). Abstra: Toward Generic Abstractions for Data of Any Model, *31st ACM International Conference on Information & Knowledge Management 2022*, Atlanta

Kolev B., Valduriez P., Bondiombouy C., et al. (2016). CloudMdsQL: querying heterogeneous cloud data stores with a common language. *Distributed and parallel databases* vol. 34, p. 463-503.