

Relations complexes au sein du système Hive

Matthieu Pilven^{*,**} Stefanie Scherzinger^{*}
Laurent d’Orazio^{**}

^{*}OTH Regensburg, Regensburg, Germany
stefanie.scherzinger@oth-regensburg.de,

^{**}Univ Rennes, CNRS, IRISA, France
laurent.dorazio@irisa.fr

Résumé. Dans cet article, nous abordons le problème de la modélisation de données pour les entrepôts au sein du système Hive. Ce moteur SQL sur Hadoop bien connu permet des relations avec des valeurs complexes, avec différents types d’attribut non nécessairement atomiques. A l’aide d’une étude empirique, nous analysons des dossiers de sources libres qui contiennent des déclarations de schémas Hive. Nous examinons dans quelles limites les utilisateurs créent des relations à valeur complexe et par conséquent leurs requêtes sur les valeurs complexes. Comprendre comment les utilisateurs conçoivent leurs modèles de données Hive, en particulier quelles fonctionnalités ils utilisent, devrait nous aider à prendre de bonnes décisions de conception pour la création d’un banc d’essais réaliste pour les moteurs SQL sur Hadoop, et choisir quels opérateurs de requêtes utiliser pour une bonne optimisation.

1 Introduction

A l’origine un entrepôt de données interne à Facebook Thusoo et al. (2009), le moteur SQL sur Hadoop Hive est désormais un projet officiel Apache. Après seulement une décennie de développement, Apache Hive figure parmi les 10 premiers systèmes de bases de données relationnelles sur le site Web DB-Engines Ranking ¹, aux côtés des bases de données de DB2 ou Oracle. Pourtant, il s’agit de produits commerciaux dont l’historique de développement est jusqu’à quatre fois plus long. Une partie de ce succès est due au fait que travailler avec Hive est très familier :

- Les utilisateurs de Hive interagissent facilement avec un modèle de données relationnel, mais les données brutes sont généralement stockées dans le système de fichiers distribués de Hadoop (HDFS).
- Bien que le langage de requête HiveQL ressemble étroitement au dialecte SQL de MySQL (White, 2009), les requêtes sont exécutées sous forme de tâches MapReduce évolutives.

De plus, Les relations Hive n’ont pas forcément besoin d’être en première forme normale, où toutes les valeurs d’attributs sont atomiques. Au contraire, le modèle de données permet

1. <https://db-engines.com/de/ranking/relational+dbsms>, de Juin 2019.