



N° d'ordre NNT : 2019LYSE2099

## THESE de DOCTORAT DE L'UNIVERSITÉ DE LYON

Opérée au sein de

L'UNIVERSITÉ LUMIÈRE LYON 2

**École Doctorale : ED 512 Informatique et Mathématiques**

Discipline : Informatique

Soutenue publiquement le 21 novembre 2019, par :

**Yassine RAMDANE**

---

### **Big Data Warehouse :**

*Modèle de distribution de données dans des cubes à la volée.*

---

Devant le jury composé de :

Salima HASSAS, Professeure des universités, Université Claude Bernard Lyon 1, Présidente

François PINET, Directeur de Recherches, Inst. Nat. Rech.en Sces et Tech. Environmt et Agricult., Rapporteur

Olivier TESTE, Professeur des universités, Université Toulouse 2, Rapporteur

Vargas-Solar GENOVEVA, Chargé de recherche, CNRS, Examineur

Abdelmouneem REZGUI, Maître de conférences, Illinois State University, Examineur

Omar BOUSSAID, Professeur des universités, Université Lumière Lyon 2, Directeur de thèse

Nadia KABACHI, Maître de conférences, Université Claude Bernard Lyon 1, Co Directrice

## Contrat de diffusion

Ce document est diffusé sous le contrat *Creative Commons* « [Paternité – pas d'utilisation commerciale – pas de modification](#) » : vous êtes libre de le reproduire, de le distribuer et de le communiquer au public à condition d'en mentionner le nom de l'auteur et de ne pas le modifier, le transformer, l'adapter ni l'utiliser à des fins commerciales.

## Résumé substantiel de la thèse

Le traitement des données massives devient un grand défi pour les entreprises, au même moment *Hadoop* s'impose comme un standard pour le traitement Big Data. Aujourd'hui, plusieurs grandes entreprises, comme Facebook, Yahoo, Twitter, etc, utilisent la plateforme *Hadoop* pour gérer et traiter leurs données massives. Dans le cadre de notre thèse, nous travaillons à améliorer les performances des entrepôts de données volumineux distribuées sur un cluster Hadoop. Plus particulièrement, nous focalisons sur l'optimisation des requêtes OLAP afin d'analyser les entrepôts de données distribuées sur un cluster Hadoop.

Pour améliorer le temps d'exécution des requêtes OLAP, exécutées sur un système distribué comme *Hadoop* écosystème ou *Spark*, plusieurs techniques de partitionnement et de distribution des entrepôts de données massives ont été proposées. Les requêtes OLAP sont généralement des requêtes coûteuses qui nécessitent beaucoup de temps pour être exécutées sur des Entrepôts de Données Distribuées (EDD) massives. Dans les EDD massives, l'amélioration du traitement d'une requête OLAP est une tâche ardue. Une requête OLAP est composée de plusieurs clauses et prend généralement la forme "Select... Function ()... From...Where... Join-Predicates ...Filters... GROUP BY...". Chaque clause peut être exécutée dans la phase *map* ou dans la phase *reduce*, et chaque opération peut être exécutée avec un ou plusieurs cycles MapReduce ou stages de Spark, avec une quantité considérable de données transférée entre les nœuds.

Pour optimiser le traitement des opérations d'une requêtes OLAP, nous avons proposées différents techniques statiques et dynamiques de Partitionnement et d'Equilibrage des Charges de données (PECh). La base de ces techniques est la fragmentation horizontale. Nous avons proposées quatre contributions. Les approches proposées s'articulent autour de quatre axes principaux:

1. Proposition d'une approche basée sur les données ou sur le schéma de l'EDD, en utilisant une technique statique de de PECh des données, pour équilibrer les données d'entrées fractionnées d'un EDD massives sur un *cluster* de nœuds homogènes.

2. Proposition d'une approche basée sur une charge de requêtes, en utilisant une technique statique de PECh de données, pour gérer les métadonnées des tables d'un EDD massives et pour éviter de charger des blocs de données HDFS inutiles.
3. Proposition d'une approche hybride (combine entre la stratégie basée sur les données et celle qui est basée sur la charge de requêtes) pour construire une nouvelle conception physique d'un EDD massives sur *Hadoop*.
4. Proposition d'une approche basée sur les données, en utilisant une technique dynamique de PECh de données, pour équilibrer les charges de reducers d'une requête OLAP en cours d'exécution sur un cluster de nœuds homogènes, à l'aide d'un système multi-agents (SMA).

Dans nos contributions, nous prenons en compte plusieurs paramètres tels que la taille de l'EDD, la distribution des valeurs des clés primaires et des clés étrangères des tables de faits et de dimensions, ainsi que les caractéristiques physiques des nœuds du *cluster* (nombre de cœurs de processeur, taille de la mémoire, etc.). Nous utilisons ces paramètres pour créer un bon schéma de partitionnement de distribution d'un EDD massives sur un cluster Hadoop, et pour améliorer les performances du système distribué, ce qui peut améliorer le temps d'exécution de la charge de travail OLAP.

L'objectif de la première contribution est d'optimiser le fonctionnement de la jointure en étoile. Cette opération est la plus coûteuse dans une requête OLAP et peut créer un embouteillage dans le réseau en raison de la quantité de données transférés dans la phase de *shuffle*. Habituellement, cette opération nécessite plusieurs itérations MapReduce, ce qui peut considérablement ralentir le traitement de la requête. Nous proposons une heuristique dans laquelle l'optimiseur du système de traitement des requêtes SQL peut exécuter cette opération en une seule itération *MapReduce* ou *Spark stage*, sans la phase de *shuffle*. Nous montrons également comment améliorer cette opération sans tenir compte de la charge de travail utilisée.

L'objectif de la deuxième proposition est d'éviter de charger des blocs HDFS inutiles lors de l'analyse de tables d'un EDD massives et d'éviter d'obtenir de grosses partitions de certaines tables, ce qui peut perturber le traitement parallèle. Dans cette contribution, nous

maitrisons le nombre de partitions créées pour chaque table ainsi que leurs tailles. Nous proposons une technique de PECh de données, en se basant sur la charge de requêtes utilisée, à savoir l'utilisation d'une charge de travail stable. Dans cette approche, nous combinons les techniques de partitionnement et de *bucketing*.

Dans la troisième proposition, nous combinons les deux premières approches (modèle basé sur les données et modèle basé sur la charge de travail). Nous proposons donc une nouvelle conception physique d'un EDD massives, dans laquelle le système peut exécuter l'opération de jointure en étoile dans une seule étape Spark, et il peut ignorer l'analyse de blocs HDFS non pertinents de certaines tables de l'EDD.

La quatrième contribution est différente des précédentes. Dans cette approche, nous proposons une nouvelle technique dynamique de PECh de données, en utilisant un SMA, pour équilibrer à la volée les charges de *reducers*. L'objectif principal de cette solution est d'optimiser la clause Group by d'une requête OLAP et les fonctions d'agrégations. La clé principale de cette proposition est que nous n'apportons aucune modification au mécanisme par défaut du paradigme MapReduce comme les approches de l'état de l'art.

Pour évaluer nos contributions, nous avons utilisés le banc d'essai TPC-DS, en utilisant différent taille d'un EDD en schéma en étoile dont le nombre d'enregistrements de la table des faits peut atteindre plus de 28 milliards. Nous avons utilisé un *cluster* Hadoop-YARN composé entre 5 et 15 nœuds, le moteur d'exécution *Spark*, le système *Hive* ainsi que la librairie *Ray*. Les résultats expérimentaux montrent que nos approches surpassent plusieurs approches existantes en terme du temps d'exécution d'une requête OLAP.