

Master :

Mathématiques appliquées de la science de données

HIVE

Présenté par :

Salma El Goude
Touayba BAISSA
Meryeme ERRAJRAJI
Mustapha OUSALM

Sous la direction de :

Prof. Charaf HAMIDI

Année Universitaire : 2023-2024

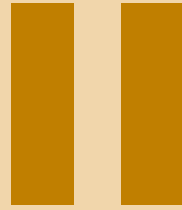


on t- on

PLAN



Introduction



**Fonctionna-
lités de
Hive**



**Hive en
pratique**



**Avantages
et
limites**



Conclusion

Qu'est-ce que Hive ?

Hive est un système de gestion de base de données distribué qui fait partie intégrante de l'écosystème Hadoop. Il a été spécifiquement conçu pour faciliter et accélérer les requêtes sur de grands ensembles de données, notamment celles stockées dans des systèmes de fichiers distribués comme le Hadoop Distributed File System (HDFS).

Qu'est-ce que Hadoop ?

Hadoop est un framework logiciel open source utilisé pour le stockage et le traitement distribué de grands ensembles de données sur des clusters de machines. Il offre un système de fichiers distribué (HDFS) pour le stockage et un modèle de programmation (MapReduce) pour le traitement parallèle des données. Hadoop permet aux entreprises de gérer et d'analyser efficacement d'énormes volumes de données, offrant ainsi des solutions évolutives et rentables pour le big data.

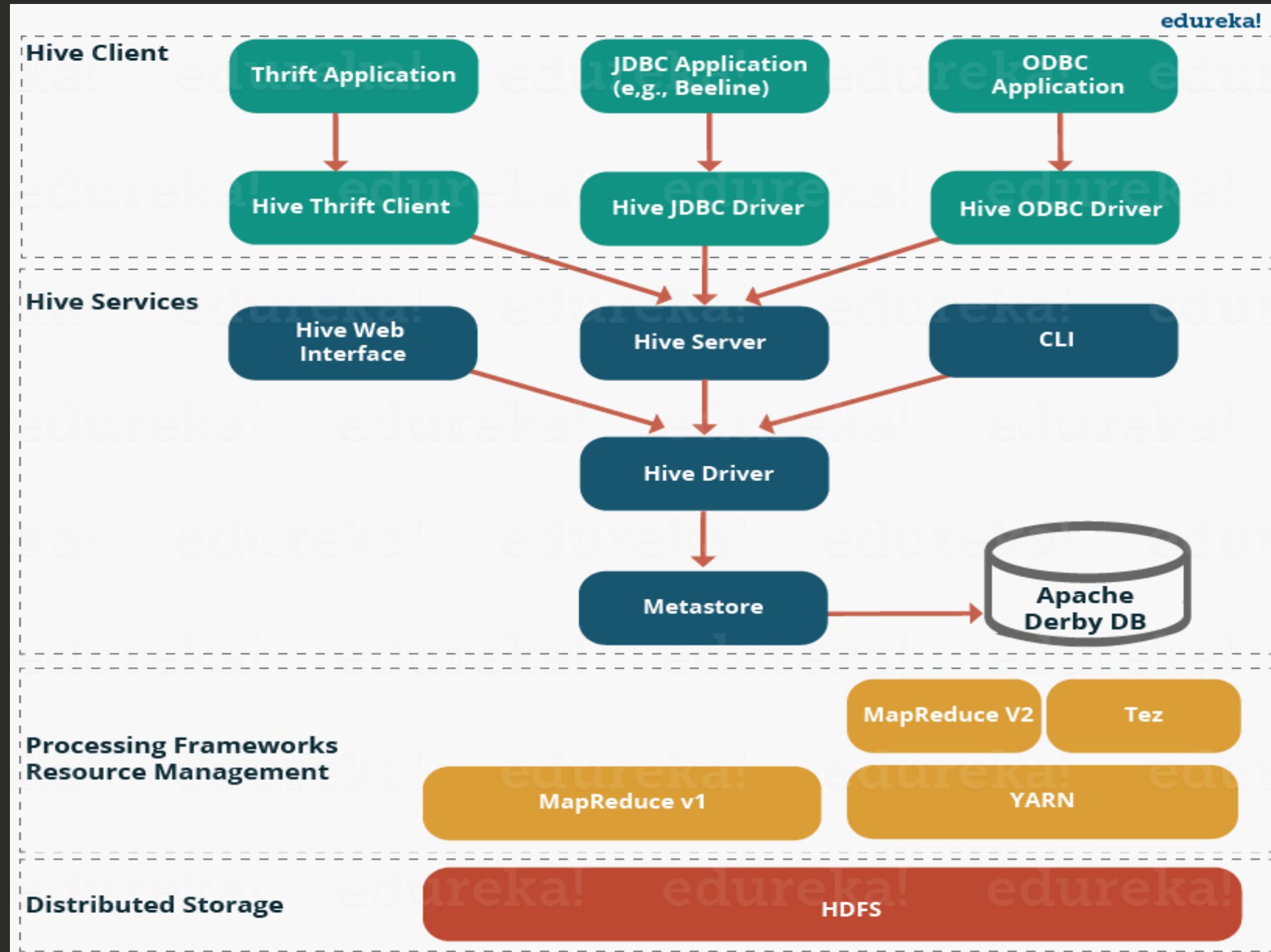
Qu'est-ce que HDFS ?

Hadoop Distributed File System (HDFS) : est un système de fichiers distribué conçu pour fonctionner sur du matériel informatique standard et peu coûteux. Il offre un accès haute performance aux données pour les applications Hadoop, avec une grande tolérance aux pannes et une capacité à gérer de très grands ensembles de données. HDFS divise les fichiers en blocs de données et les distribue à travers les nœuds d'un cluster, permettant ainsi un traitement des données efficace et parallèle.

Problématique :

1. Quels sont les principaux avantages de l'utilisation de Hive pour l'analyse de données à grande échelle par rapport aux systèmes de gestion de base de données traditionnels ?
2. Comment Hive gère-t-il les défis liés à la performance des requêtes complexes ?

Architecture de Hive :



Fonctionnement de Hive :

Fonctionnalités

Apache Hive prend en charge l'analyse de grands ensembles de données stockés non seulement dans HDFS d'Hadoop mais aussi dans des systèmes de fichiers compatibles tels qu'Amazon S3, Google Cloud Storage et Alluxio.

Hive permet l'exploitation des données compressées stockées dans l'écosystème Hadoop à l'aide d'algorithmes.

Hive peut être facilement intégré avec Apache Spark, en effet il dispose d'une bibliothèque appelée **Apache Hive Warehouse Connector (HWC)** permettant de travailler plus facilement avec Spark. Elle prend en charge certaines tâches comme le déplacement de données entre les DataFrames Spark et les tables Hive.

Il fournit un langage de requête de type SQL appelé **HiveQL** et permet de convertir de manière transparente les requêtes en jobs Apache Spark, MapReduce et Apache Tez.

Le Metastore de Hive, agissant comme un catalogue centralisé, stocke toutes les informations sur les tables Hive, y compris leur structure (colonnes, partitions) et leur localisation dans HDFS, permettant ainsi à Hive de comprendre la structure des données et de traiter efficacement les requêtes.

Définition de HQL :

Hive Query Language (HiveQL) est un langage de requête pour Hive qui permet d'analyser et de traiter des données structurées dans un Meta-store. Il est très similaire à SQL et hautement évolutif. Il réutilise des concepts familiers du monde des bases de données relationnelles, tels que les tables, les lignes, les colonnes et les schémas, pour faciliter l'apprentissage

Comparaison entre HQL et SQL :

Caractéristiques	SQL	HiveQL
Commandes de mise à jour	UPDATE, DELETE, INSERT	Seulement INSERT
Types de données	Cinq types de base	Plusieurs types supplémentaires
MapReduce	Non Supporté	Supporté
Insertions multi-tables	Non Supporté	Supporté
Créer une table à partir d'une sélection	Non Supporté	Supporté
Commande SELECT	Ne supporte pas	Supporte SORT BY et LIMIT
Jointures	Moins de types supportés	Plusieurs types supportés
Sous-requêtes	Supporté	Limité
Vues	Peuvent être mises à jour	Lecture seule

LES FORMATS PRIS EN CHARGE PAR HIVE :

Hive prend en charge plusieurs formats de fichiers pour stocker les données. Parmi les formats supportés nous avons :

1

TEXTFILE :

C'est le format le plus simple, les données sont stockées sous forme de lignes de texte brut. Il peut être utilisé avec des séparateurs de champs comme la virgule (CSV) ou la tabulation (TSV).

2

SEQUENCEFILE :

Ce format stocke les données sous forme de paires clé-valeur. Il est plus flexible que le TEXTFILE mais aussi plus complexe à traiter.

3

ORC (Optimized Row Columnar) :

C'est un format de stockage de données en colonnes optimisé pour les requêtes analytiques. Il est efficace pour compresser les données et permet des lectures plus rapides.

4

RCFile (Record Columnar) :

Autre format de stockage en colonnes, il est moins répandu que l'ORC mais offre des fonctionnalités similaires.

La difference entre Hive et SGBD traditionnel :

Hive

SGBD traditionnel

1

Utilisation

est utilisé pour maintenir l'entrepôt de données.

sont utilisés pour maintenir la base de données.

2

Langage de requête

utilise le langage de requête Hive (HQL).

utilisent le langage de requête structuré (SQL).

3

Schéma

offre une flexibilité de schéma

le schéma est fixe

4

Données stockées

peut stocker à la fois des données normalisées et dénormalisées

stockent des données normalisées

5

Partitionnement

supporte le partitionnement automatique

ne supportent pas le partitionnement

Fonctionnalités

Approche NoSQL de Hive :

Hive adopte une approche NoSQL en fournissant une flexibilité de schéma, ce qui est avantageux lorsqu'on travaille avec des données non structurées ou semi-structurées. Cette approche permet aux utilisateurs de définir la structure des données lors de l'exécution de la requête, contrairement aux SGBD traditionnels qui nécessitent un schéma bien défini à l'avance. De plus, Hive est bien adapté pour le traitement par lots, où de grands ensembles de données sont traités à intervalles programmés.

Exemple de requêtes :

Créer une base de données :

```
CREATE DATABASE ma_base_de_donnees;
```

Utiliser une base de données :

```
USE ma_base_de_donnees;
```

Créer une table :

```
CREATE TABLE ma_table (  
  colonne1 TYPE,  
  colonne2 TYPE,  
  ...  
);
```

Insérer des données dans une table :

```
INSERT INTO ma_table VALUES (valeur1, valeur2, ...);
```

Exemple de requêtes :

supprimer les données d'une table :

```
DELETE FROM ma_table WHERE condition;
```

Supprimer une table :

```
DROP TABLE ma_table;
```

Supprimer une base de données :

```
DROP DATABASE ma_base_de_donnees;
```

Créer une table externe :


```
CREATE EXTERNAL TABLE ma_table_external (  
    colonne1 TYPE,  
    colonne2 TYPE,  
    ...  
)  
LOCATION '/chemin/vers/les/donnees';
```


Application :

```
CREATE TABLE employee (  
  id INT COMMENT 'Identifiant unique de l'employé',  
  nom VARCHAR(50) COMMENT 'Nom de l'employé',  
  prenom VARCHAR(50) COMMENT 'Prénom de l'employé',  
  date_naissance DATE COMMENT 'Date de naissance de l'employé',  
  departement VARCHAR(50) COMMENT 'Département de l'employé',  
  salaire DECIMAL(10,2) COMMENT 'Salaire de l'employé')  
COMMENT 'Table stockant les informations des employés'  
ROW FORMAT DELIMITED;  
FIELDS TERMINATED BY ;  
LINES TERMINATED BY '\n';
```

IV

Avantages de Hive :



Facilité d'utilisation : Hive utilise un langage similaire au SQL, ce qui le rend accessible aux utilisateurs déjà familiers avec les bases de données relationnelles.

Évolutivité : Hive permet de gérer efficacement des ensembles de données volumineuses stockées sur des systèmes de stockage distribués comme Hadoop.


Requêtes SQL simplifiées : Grâce à son langage de type SQL (HiveQL), vous pouvez interroger et analyser vos données sans avoir à programmer en Java, un atout pour les analystes.

Exploration de données facilitée : La structure des tables en Hive permet d'explorer et d'agréger facilement vos données pour en extraire des informations précieuses.

Open source : Étant un logiciel open source, Hive est gratuit et sa communauté active en assure un développement continu

IV

Limites de Hive :



Traitement par lots (batch processing): Hive n'est pas conçu pour le traitement de transactions en ligne (OLTP - Online Transaction Processing). Il excelle par contre dans le traitement analytique de données (OLAP - Online Analytical Processing), analysant des jeux de données volumineux pour en extraire des tendances.

Pas de mises à jour ou suppressions directes: Contrairement à une base de données relationnelle classique, Hive ne permet pas de modifier directement des lignes de données existantes. On peut cependant ajouter de nouvelles données ou remplacer des partitions entières de tables.

Latence élevée: L'exécution des requêtes Hive peut prendre plus de temps que sur des bases de données optimisées pour les requêtes ad-hoc. Ceci est dû à son architecture basée sur le traitement par lots.

Support limité des sous-requêtes: Les requêtes imbriquées (subqueries) sont peu prises en charge par HiveQL, le langage de requête utilisé par Hive.

Conclusion :

En conclusion, Hive joue un rôle essentiel dans l'écosystème des technologies de données. En tant que système de requêtes basé sur Hadoop, Hive permet aux utilisateurs d'extraire des informations précieuses à partir de grandes quantités de données. Il offre une interface familière aux utilisateurs habitués au SQL, tout en offrant la puissance et la flexibilité nécessaires pour gérer des ensembles de données de grande taille.

perspectives

En regardant vers l'avenir, nous pouvons nous attendre à ce que Hive continue d'évoluer pour répondre aux besoins changeants du monde des données. Avec l'augmentation constante du volume de données générées, la demande pour des outils comme Hive ne fera que croître.

Les prochaines versions de Hive promettent d'apporter des améliorations en termes de performances, de fonctionnalités et d'accessibilité.

MERCI

POUR VOTRE ATTENTION