



Pr. Ghazouani mohamed

7 décembre 2024

## Introduction



Apache Hive est un système d'entrepôt de données open source. Il permet d'interroger et d'analyser des ensembles de données volumineux (Big Data) stockés dans des fichiers Hadoop.

- Utilisé pour les **entrepôt de données** / base de données décisionnelle.
- Hive propose aussi une fonction de stockage distribué et permet d'accéder à des fichiers stockés dans HDFS (ou dans d'autres systèmes comme Apache HBase).
- En terme de langage, **Hive** propose **HiveQL**, un langage déclaratif, similaire à SQL
- Application : (Text mining, Log analysis (pig est plus utilisé), BI, Document indexing, Predictive modeling, etc.. )

7 décembre 2024

## Introduction



Apache Hive est un datawarehouse pour Hadoop. Il a été créé par Facebook pour devenir par la suite un projet Apache open source. Il ne s'agit pas d'une base de données relationnelle ni d'un datawarehouse classique.

### Si Hive n'est pas une base de données ni un datawarehouse, qu'est-ce donc alors ?

Il s'agit d'un système qui maintient des métadonnées décrivant les données stockées dans HDFS. Il utilise une base de données relationnelle appelée **metastore** (c'est une base de données embarquée Apache Derby par défaut) pour assurer la persistance des métadonnées. Ainsi, une table dans Hive est composée essentiellement :

- D'un schéma stocké dans le **metastore**,
- Des métadonnées décrivant les données stockées dans HDFS.

Avec les données du metastore, Hive permet de manipuler les données comme si elles étaient persistées dans des tables (au sens d'un système de gestion de base de données classique) et de les interroger avec son langage HiveQL.

Pr. M. Ghazouani

3

7 décembre 2024

## Introduction



En général, plus le modèle de données se complexifie, plus l'écriture d'un job MapReduce qui les manipule devient fastidieuse. Si nous prenons le simple exemple du Word count que nous trouvons sur la documentation officielle de Hadoop, l'implémentation Java fait une centaine de lignes environ avec :

- 15 lignes pour le mapper,
- 12 lignes pour le reducer,
- 35 lignes pour le setup ainsi que les méthodes utilitaires pour le parsing des données en entrées,
- 30 lignes pour le main.

Tout ça pour un Word Count avec MapReduce !

Afin de faciliter l'analyse de données stockées dans HDFS sans passer par la complexité de MapReduce, certains frameworks comme Pig, Hive sont apparus. Ils permettent de **convertir les requêtes HiveQL en jobs MapReduce**.

Pr. M. Ghazouani

4

7 décembre 2024

## Hive supporte les clauses SQL standards



D'un point de vue syntaxe, Hive supporte les clauses SQL standards :

```
INSERT INTO
SELECT
FROM ... JOIN .... ON
WHERE
GROUP BY
HAVING
ORDER BY
LIMIT
```

Ainsi que les commandes de définition de structure (DDL) :

```
CREATE / ALTER / DROP TABLE / DATABASE
```

Pr. M. Ghazouani

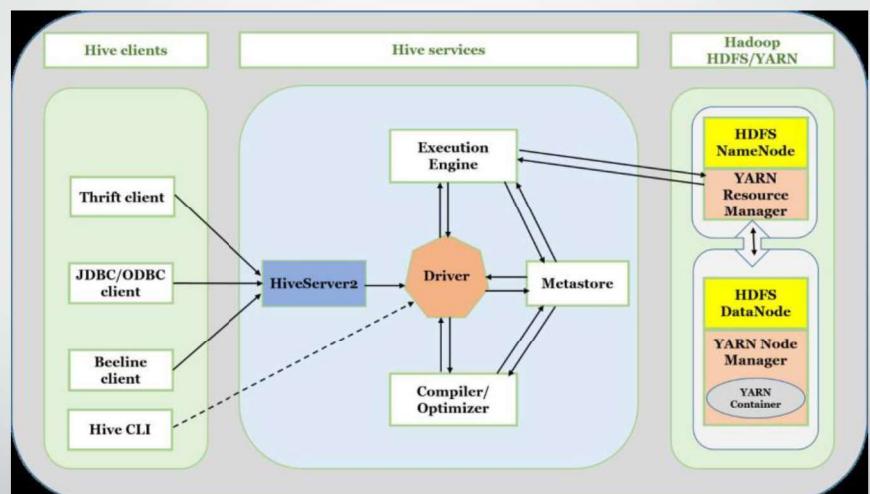
5

7 décembre 2024

## Les composantes de l'architecture Hive

L'architecture de HIVE est constituée de trois principales couches :

- 1) - la couche de requêtage (Hive Clients)
- 2) - la couche de traitements des requêtes (Hive Services)
- 3)- la couche d'exécution et de Stockage (HADOOP HDFS/YARN).



Pr. M. Ghazouani

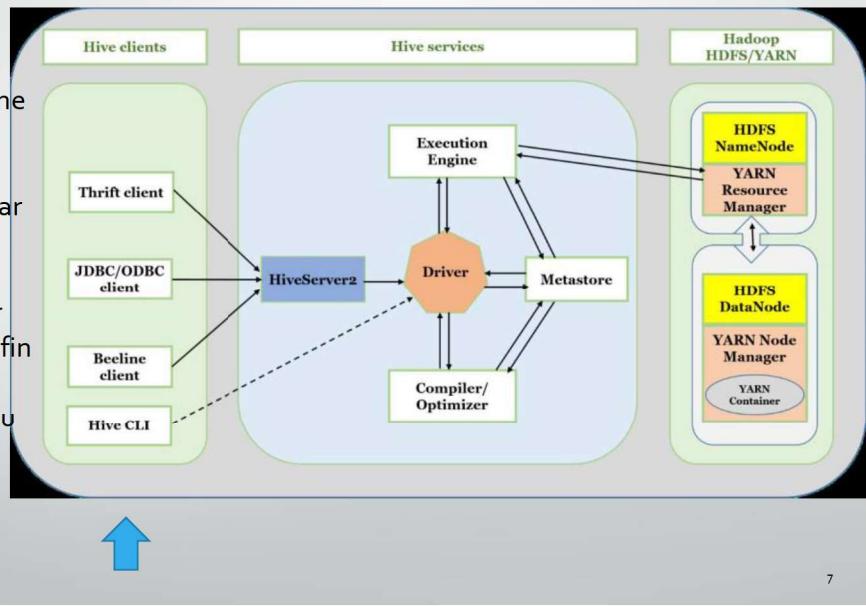
6

7 décembre 2024

## Les composantes de l'architecture Hive

### Clients directement intégrés à Hive:

- Hive CLI (Command Line Interface): permet d'interroger Hive en passant directement par le metastore et HDFS.
- Hive CLI permet aux utilisateurs d'exécuter des requêtes HiveQL afin de créer, modifier et supprimer des tables ou bases de données.



Pr. M. Ghazouani

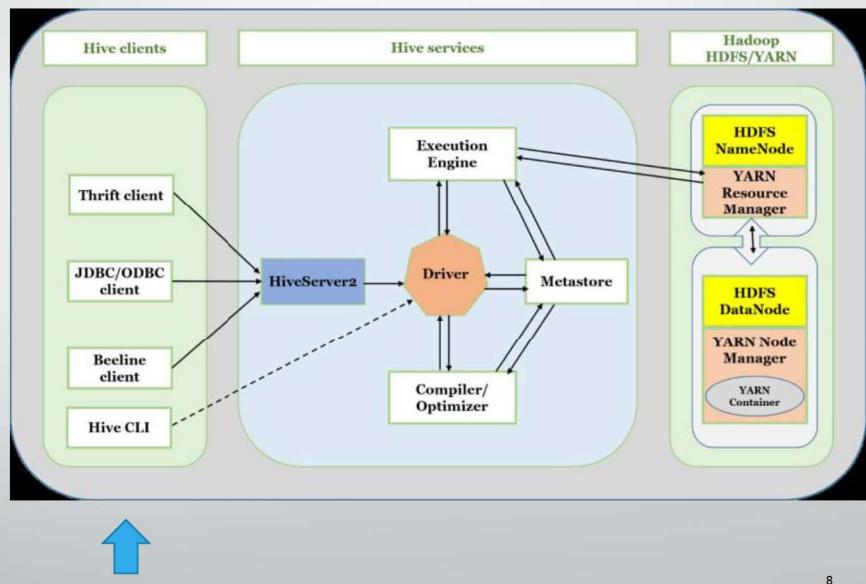
7

7 décembre 2024

## Les composantes de l'architecture Hive

### Clients directement intégrés à Hive:

- Beeline : est une interface en ligne de commande (CLI) basée sur JDBC utilisée pour interagir avec les bases de données Hive.
- Elle offre un moyen plus moderne et robuste d'exécuter des commandes SQL et de se connecter à Hive, en remplacement de l'ancienne CLI de Hive.



Pr. M. Ghazouani

8

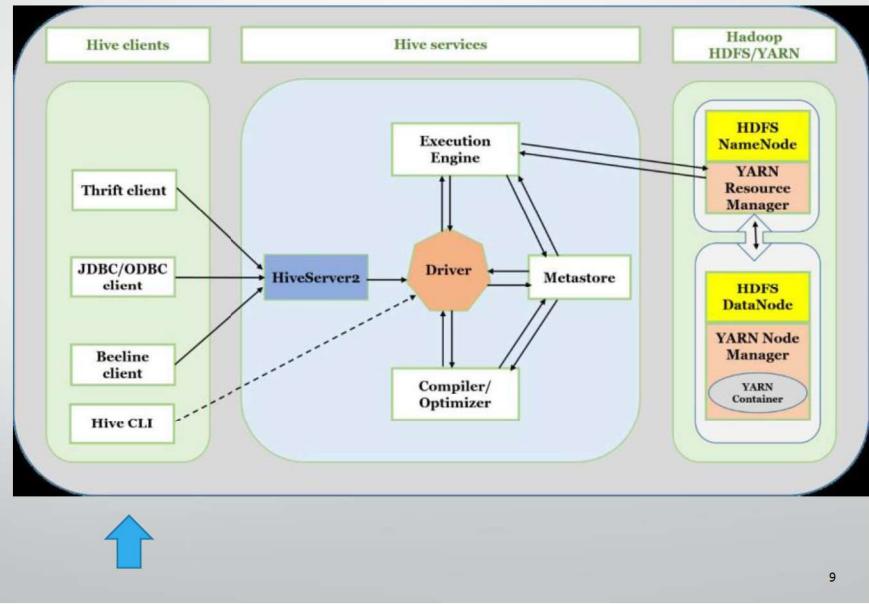
7 décembre 2024

## Les composantes de l'architecture Hive

### Clients externes:

JDBC/ODBC : ici, il s'agit de toute application externe permettant d'interroger Hive par l'intermédiaire d'un connecteur (pilote) spécifique.

Parmi ces types de client, on peut citer par exemple : SQL Workbench, Ms Excel.



Pr. M. Ghazouani

9

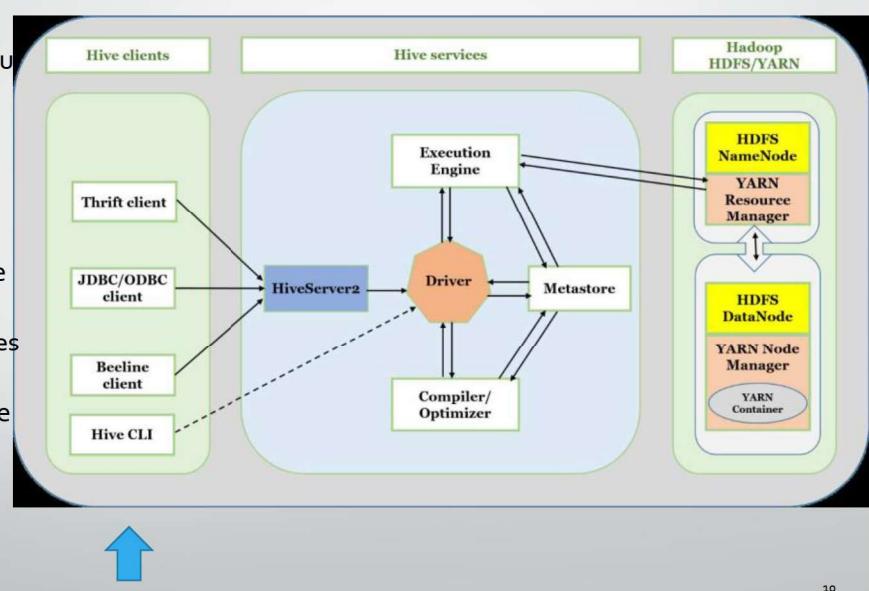
7 décembre 2024

## Les composantes de l'architecture Hive

### 1. Les clients Thrift :

Si vous voulez écrire du code (par exemple en Java, Python ou PHP) pour envoyer ces requêtes, vous avez besoin d'un client.

**Thrift** est un protocole qui sert de pont entre Hive et les programmes écrits dans d'autres langages. Il transforme les requêtes en un format que Hive comprend.



Pr. M. Ghazouani

10

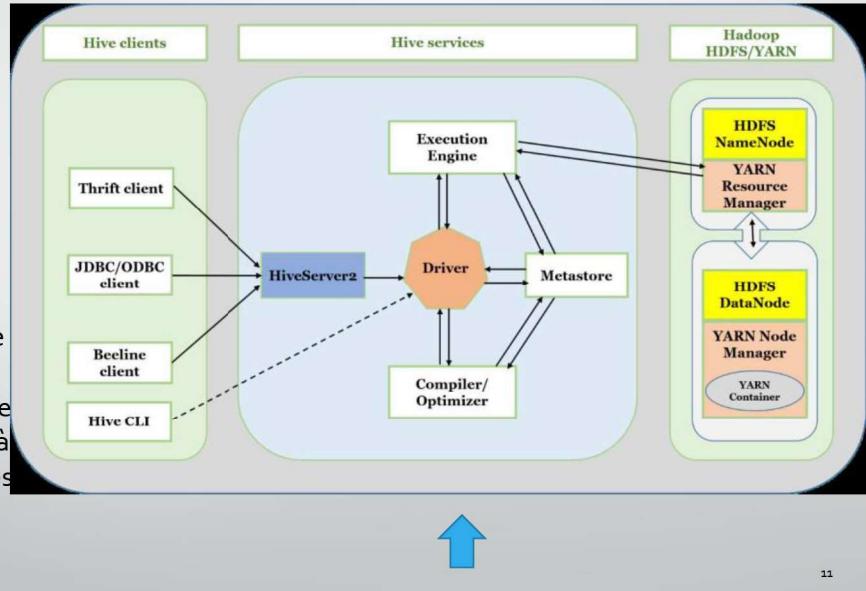
7 décembre 2024

## Les composantes de l'architecture Hive

HiveServer2 est le service à travers lequel les clients Hive soumettent leurs requêtes à Hive. HiveServer2 est bâti sur du Thrift et est parfois appelé Thrift Server. Le Thrift Server est une interface qui agit comme une passerelle permettant aux clients de soumettre des requêtes à Hive dans divers langages de programmation.

Pr. M. Ghazouani

11



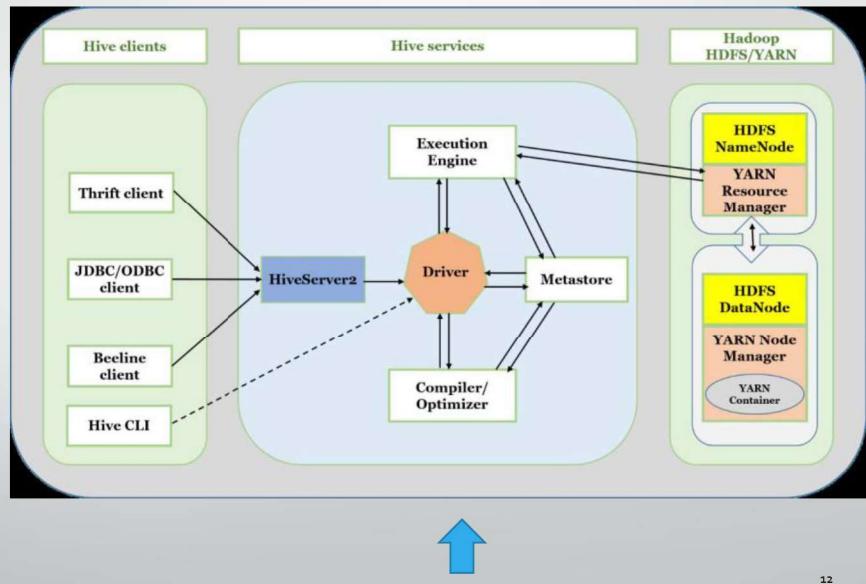
7 décembre 2024

## Les composantes de l'architecture Hive

Le metastore stocke les métadonnées, c'est-à-dire les informations sur la structure des tables, les schémas, les types de données, les emplacements des données, et les partitions.

Pr. M. Ghazouani

12



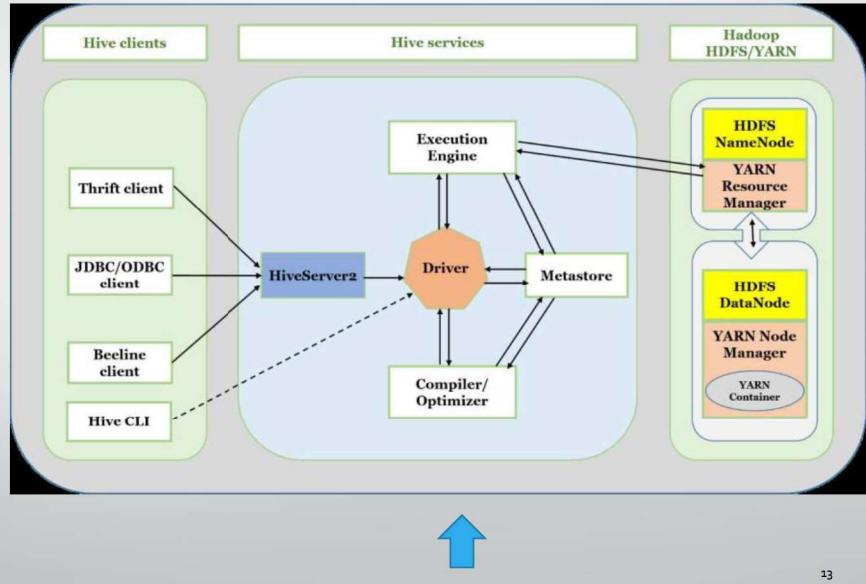
7 décembre 2024

## Les composantes de l'architecture Hive

- Le compiler est le service qui réceptionne les requêtes envoyées par le Driver.
- Une fois que la requête est reçue, le Compiler effectue une analyse sémantique de la requête, vérifie et valide la syntaxe des requêtes.
- Ensuite, il se base sur les métadonnées stockées dans le metastore pour générer un plan d'exécution de la requête.

Pr. M. Ghazouani

13



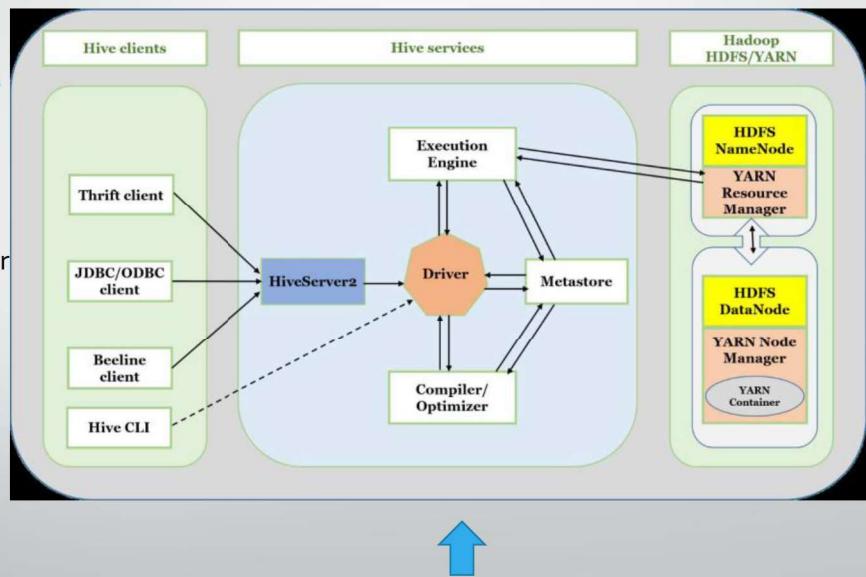
7 décembre 2024

## Les composantes de l'architecture Hive

- L'Optimizer est un service complémentaire qui effectue des opérations de transformation sur le plan d'exécution et subdivise les tâches pour plus d'efficacité et de performance lors de l'exécution des tâches.

Pr. M. Ghazouani

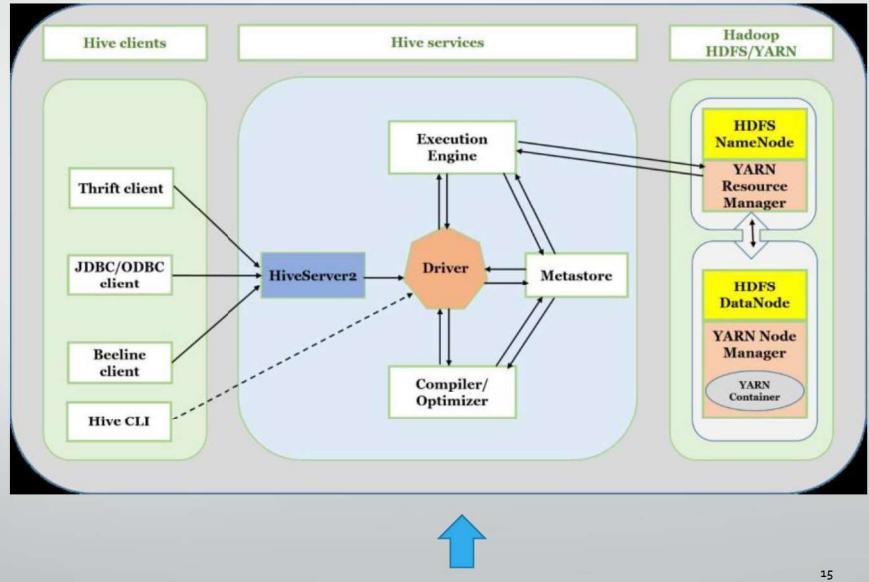
14



7 décembre 2024

## Les composantes de l'architecture Hive

- Le moteur d'exécution est l'interface entre Hive services et la couche HADOOP YARN/HDFS.
- Le moteur d'exécution reçoit le plan d'exécution et déroule les actions selon l'ordre défini dans le plan.
- L'exécution du plan prend place dans des containers mis à disposition par HADOOP YARN.



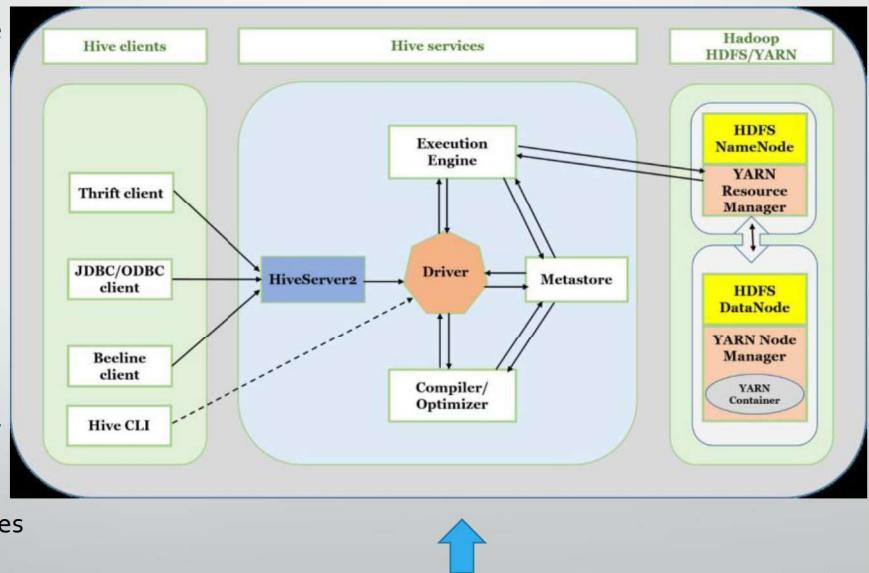
Pr. M. Ghazouani

15

7 décembre 2024

## Les composantes de l'architecture Hive

- Le driver est la composante centrale à l'architecture de Hive.
- Il reçoit les instructions HiveQL soumises par l'utilisateur via les Hive clients.
- Il crée les sessions Hive en vue de traiter la requête.
- Il envoie la requête au Compiler/Optimizer pour apporter des optimisations.
- Il transmet le plan d'exécution à l'Execution Engine pour la réalisation des opérations.



Pr. M. Ghazouani

16

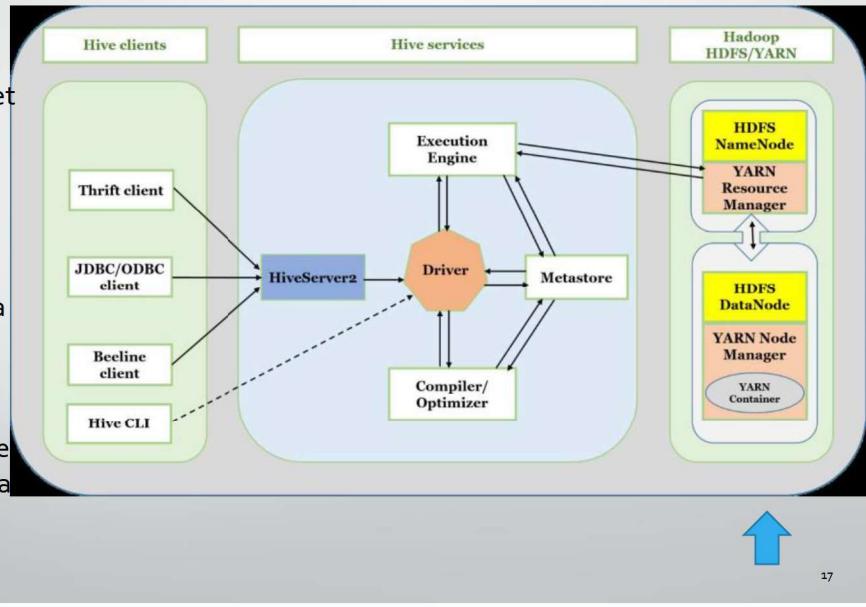
7 décembre 2024

## Les composantes de l'architecture Hive

- YARN se charge de la gestion des ressources et de la coordination des différentes étapes de l'exécution des jobs.
- Pour cela, il crée les containers et réunit les ressources nécessaires à l'exécution des tâches.
- Hive étant une surcouche de HADOOP, il utilise naturellement le FileSystem HDFS pour la lecture et le stockage des données.

Pr. M. Ghazouani

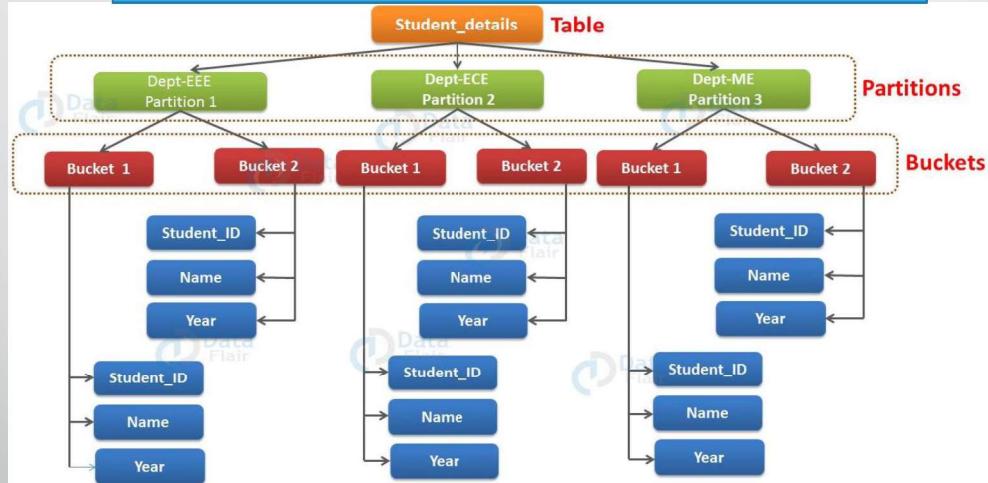
17



7 décembre 2024



## Gestion des partitions



Hive est un système d'entrepôt de données open source construit sur Hadoop pour interroger et analyser de grands ensembles de données stockés dans des fichiers Hadoop. Il traite les données structurées et semi-structurées dans Hadoop. Les données dans Apache Hive peuvent être classées en: Table, Partition et Bucket

Pr. M. Ghazouani

18

7 décembre 2024



Pr. M. Ghazouani

19

## Installation apache-hive-1.2.2



7 décembre 2024

## Database Tables, partitions et requêtes dans Hive



Créer et décrire une base de données avec des métadonnées

Avec Hive, lorsque vous créez une base de données, il est facile d'attribuer des métadonnées utiles à une base de données en tant que description, auteur et bien d'autres options. Essayons quelques-unes de ces options ici:

```
CREATE DATABASE IF NOT EXISTS
masterBigdata
COMMENT "Study BigData at FSBM"
LOCATION '/opt/hive/warehouse/jd_db'
with DBPROPERTIES ('createdby'='shubham',
'createdfor'='JournalDev');
```

Maintenant que la base de données est créée, nous pouvons voir les informations de métadonnées en **décrivant** la base de données:

```
DESCRIBE DATABASE masterBigdata;
```

Par défaut elle est créée sur /user/hive/warehouse

Pr. M. Ghazouani

20

7 décembre 2024

# Database Tables, partitions et requêtes dans Hive



## Modification de la base de données

Les métadonnées affectées à la base de données ne sont pas permanentes. Nous pouvons les changer avec une simple commande Alter Database avec la syntaxe suivante:

```
ALTER (DATABASE) database_name SET
DBPROPERTIES
(property_name=property_value, ...);
```

Nous pouvons également modifier le propriétaire d'une base de données Hive avec une commande similaire:

```
ALTER (DATABASE) database_name SET OWNER [USER|ROLE]
user_or_role;
```

Par exemple :

```
ALTER DATABASE masterBigdata
SET OWNER ROLE admin;
```

Pr. M. Ghazouani

21

7 décembre 2024

# Database Tables, partitions et requêtes dans Hive



## Afficher toutes les bases de données

Tout comme SQL, nous pouvons afficher toutes les bases de données qui existent dans Hive jusqu'à présent:

```
SHOW DATABASES;
```

## Utilisation d'une base de données

Lorsque nous voulons exécuter des commandes DDL dans une base de données particulière, nous devons la sélectionner à l'aide de la commande suivante:

```
USE masterBigdata;
```

Une fois que nous utilisons une base de données spécifique, ce n'est qu'alors que nous pouvons y exécuter des commandes liées aux tables dans la base de données.

Pr. M. Ghazouani

22

7 décembre 2024

# Database Tables, partitions et requêtes dans Hive



## Tables et chargement des données dans Hive

Une table dans Hive permet d'associer une structure à des données non structurées dans HDFS. La création d'une table dans Hive est similaire à la création d'une table dans un RDBMS et s'effectue avec la commande **CREATE TABLE**.

Il existe dans Hive deux types de tables :

- 1) **Managed table,**
- 2) **External table.**

Dans Hive, une Managed table est similaire à une table au sens RDBMS. La **différence entre une Managed table et une External table est la gestion des données lorsque la table est supprimée.**

La syntaxe générale à utiliser pour créer une nouvelle table:

```
CREATE TABLE [IF NOT EXISTS] [db_name.]table_name -
[(col_name data_type [COMMENT col_comment], ...)]
[COMMENT table_comment]
[LOCATION hdfs_path]
```

Pr. M. Ghazouani

23

7 décembre 2024

# Database Tables, partitions et requêtes dans Hive



## Hive Managed table

La suppression d'une Managed table entraîne la suppression des métadonnées ainsi que les données dans HDFS (pour notre exemple, les données sont stockées dans HDFS par défaut sous /user/hive/warehouse/product).

```
hive > CREATE TABLE product (
  productId INT,
  productName STRING,
  productCategory STRING,
  valuationDate TIMESTAMP,
  validTillDate TIMESTAMP)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ',';
```

Pr. M. Ghazouani

24

7 décembre 2024

# Database Tables, partitions et requêtes dans Hive



## Hive External table

La suppression d'une **External table** entraîne uniquement la **suppression des métadonnées**. C'est bien pratique, une External table est un moyen de protéger les données contre les commandes **drop** accidentelles.

Hive permet aussi de spécifier l'emplacement de stockage de données dans HDFS et ne pas se limiter à l'emplacement de stockage par défaut. Ceci est possible en ajoutant la clause **LOCATION** lors de la création table.

```
hive > CREATE EXTERNAL TABLE product-ext (
    productId INT,
    productName STRING,
    productCategory STRING,
    valuationDate TIMESTAMP,
    validTillDate TIMESTAMP)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
LOCATION '/user/BigDataLab/Hive_part1/products' ;
```

Pr. M. Ghazouani

25

7 décembre 2024

# Database Tables, partitions et requêtes dans Hive



## Tronquer toutes les données du table

Nous pouvons facilement vider une table Hive en exécutant une simple commande tronquer:

```
TRUNCATE TABLE db_name.table_name;
```

## Supprimer la table

Si nous souhaitons supprimer une table entière avec ses données, nous pouvons simplement la supprimer par :

```
DROP TABLE [IF EXISTS] table_name [PURGE];
```

Si PURGE est utilisé, les données ne peuvent pas être récupérées

Pr. M. Ghazouani

26

7 décembre 2024

# Database Tables, partitions et requêtes dans Hive



## Commande Hive DML

L'insertion de données dans une table Hive est également facile. Nous pouvons utiliser la commande Insérer suivante:

```
INSERT INTO TABLE masterBigdata.lessons
VALUES (20353, 'Installing Hive on Ubuntu',
'toutdeveloppement.com/20353/install-apache-hive-ubuntu-hql-
queries'), (20358, 'Installing Hadoop on Ubuntu',
'toutdeveloppement.com/20358/install-hadoop-on-ubuntu');
```

Nous pouvons voir les données que nous avons insérées comme:

```
SELECT * FROM masterBigdata.lessons;
```

Ou nous pouvons limiter les données à une seule ligne:

```
SELECT * FROM masterBigdata.lessons LIMIT 1;
```

Pr. M. Ghazouani

27

7 décembre 2024

# Database Tables, partitions et requêtes dans Hive



## Chargement des données dans une table Hive

Un des points forts de Hive réside en sa **capacité d'associer des métadonnées aux données**. La source de ces données peut être le système de fichiers ou HDFS. Pour cela, il faut utiliser la commande **LOAD DATA**.

Pr. M. Ghazouani

28

7 décembre 2024

# Database Tables, partitions et requêtes dans Hive



## Chargement des données dans une table Hive

### Pour une Managed table :

Les données sont déplacées dans un sous répertoire de /user/hive/warehouse qui est le répertoire racine par défaut. Le mot clé **LOCAL** signifie que le fichier d'entrée est dans le système de fichiers local. Si LOCAL est omis, il s'agit d'un fichier d'input dans HDFS.

Le mot clé **OVERWRITE** signifie que les données (si elles existent) dans la table product seront supprimées. Si OVERWRITE est omis, les données seront ajoutées (mode append) aux données existantes.

```
hive> LOAD DATA LOCAL INPATH '/tmp/product.txt' OVERWRITE
      INTO TABLE product ;
```

Pr. M. Ghazouani

29

7 décembre 2024

# Database Tables, partitions et requêtes dans Hive



## Chargement des données dans une table Hive

### Pour une External table :

les données sont déplacées dans le répertoire spécifié dans la clause LOCATION de la définition de la table. Pour cet exemple, le fichier d'entrée product-ext.txt sera déplacé dans /user/BigDataLab/Hive\_part1/products puisqu'il s'agit de l'emplacement défini lors de la création de la table product-ext (crée dans slide 16).

```
hive> LOAD DATA INPATH '/user/training/product-ext.txt' INTO
      TABLE product-ext ;
```

Pr. M. Ghazouani

30

7 décembre 2024

## Database Tables, partitions et requêtes dans Hive



### Gestion des partitions

Partitionner une table dans Hive implique une séparation des fichiers selon la colonne (ou les colonnes) définissant la clé de partition. Le **partitionnement peut améliorer les performances des requêtes HiveQL**.

La définition d'une partition est similaire à sa définition en SQL :

```
CREATE TABLE product_partitioned (
    productId INT,
    productName STRING,
    valuationDate TIMESTAMP,
    validTillDate TIMESTAMP) partitioned by (productType STRING);
```

Dans HDFS, nous aurons une structure de données en sous-répertoire par productType sous /user/hive/warehouse/product\_partitioned :

|  |   |
|--|---|
| <pre>/user/hive/warehouse/product_partitioned     /productType=bond/     /productType=future/     /productType=forward/     /productType=option/</pre> | <i>Structure de données<br/>partitionnées dans HDFS</i> |
|--|---|

Pr. M. Ghazouani

31

7 décembre 2024

# Fin de la partie 6

Pr. M. Ghazouani

32