Hive

→ Apache hive is an open source data warehousing software used for reading, writing, quering and managing large amount of data set files that are stored directly in either Apache hadoop distributed file system (HDFS) or other storage systems such as Apache Hbase.

→ Data analysts often use hive for analysis quering large amount of unstructured data and generate data summaries.

→ It stores schema in databases and processes data into Hdfs.

→ It is designed for OLAP.

→ It provies SQL like language for quering called HQL (or) HiveQL.

→ It is developed by facbook and now maintained as a apache project.

→ Hive support variety of storage formats:-

- *Textfile for plain text.
- *Sequencefile for binary key value pair.
- *RC file stores colums of tables in a recorded column format.

* Hive table structure consist of rows and column, where:-
  - rows represents some records or particular entity details.
  - columns represents various attributes or characteristics for each row.

Hive Use cases:-

→ Exploratory analysis of HDFS data:
Data can be queried, transformed and exported to analytical tools.

→ Extracts or feed data to reporting systems, dashboards, or data repositories such as HBase.

→ Combining external structured data to data that already exist in HDFS

Hive Architecture & Work flow:-

→ Hive mainly consists of 4-main components):-

i) Hive client:-

→ Interface for users to interact with hive throught (Thrift, JDBC, ODBC)

ii) Hive Services:-

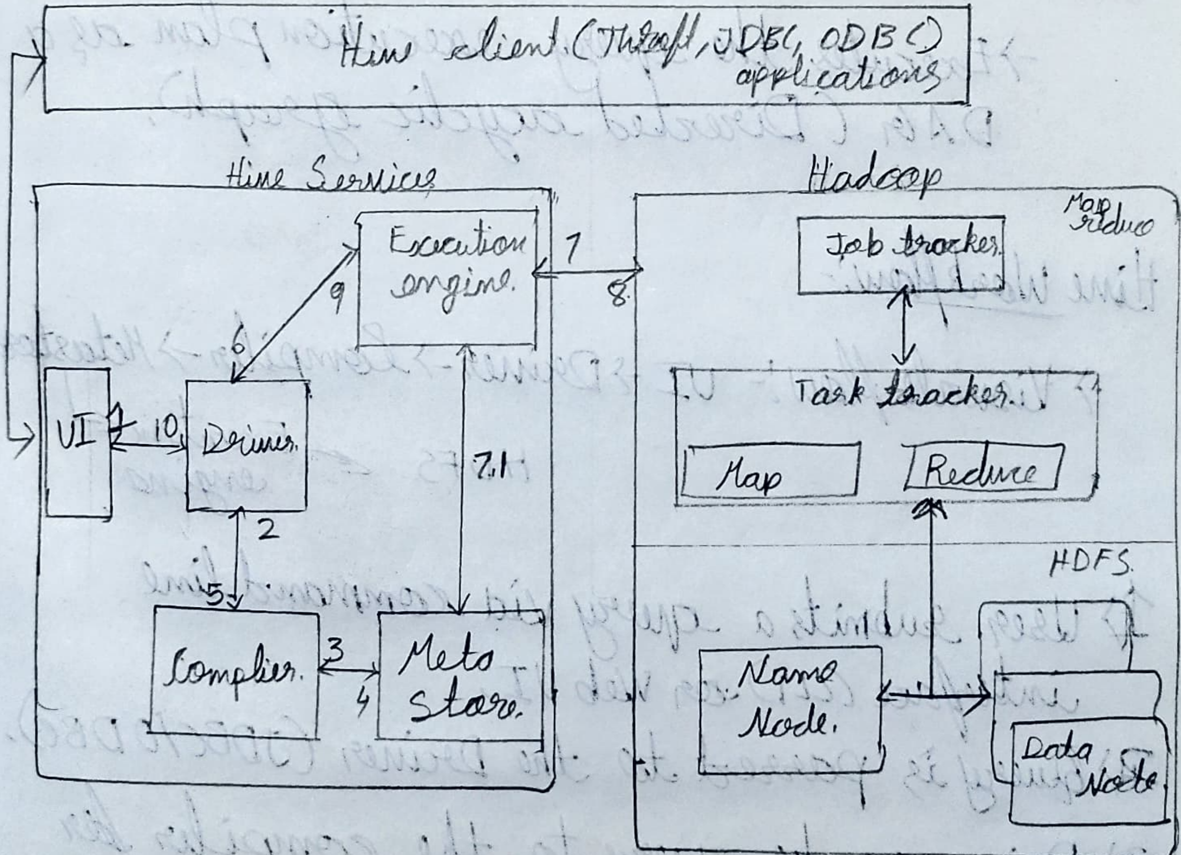
→ Includes Baseline (CLI), Server for query management & Metastore, etc.

iii > Processing & Resource Management:-

-> Uses Mapreduce for query execution & Yarn for resource Management.

iv) Distributed storage:-

-> Data is stored on HDFS for scalability and fault tolerance.



i> UI:-

->Allows users to interact with Hive

->Types: CLI, Web UI, JDBC/ODBC.

ii > Driver:

-> Manages query execution flow using JDB/ODBC.

-> Implements session handles and API s.

iii) **Compiler:**

→ Parses queries, perform semantic analysis & generates execution plan using Metadata.

iv) **Metastore:**

→ Stores tables metadata, partition info, and HDFS file mappings.

v) **Execution engine:-**

→ Executes the query execution plan as a DAG (Directed acyclic graph).

Hive Workflow:-

→ Visuals flow:- UI → Driver → Compiler → Metastore
HDFS ← Execution engine

1) User submits a query via command line interface (CLI) or Web UI.

2) Query is passed to the Driver (JDBC/ODBC).

3) Driver sends query to the compiler for parsing & plan creation.

4) Compiler requests metadata from the Metastore.

5) Metastore returns the required metadata to the compiler.

6) Compiler generates the execution plan & sends it to the Driver.

7> Driver forwards the plan to the execution engine.

8> Execution engine runs the job (Mapreduce/spark)

9> Results are retrieved from HDFS.

10> Results are sent to the Driver & then back to the User Interface (UI).

Advantages of Hive:

1> Handles Big Data.
2> Scalability
3> Fault tolerance.
4> Cost-effective (open-source).
5> Familiarity with SQL.

Disadvantages of Hive:-

1> Not for OLTP
2> Not ideal for small Data.
3> Performance is slower compared to spark.
4> Limited SQL functions.