HIVE ARCHITECTURE

What is HIVE

Hive is a data warehouse system which is used to analyze structured data. It is built on the top of Hadoop. It was developed by Facebook.

Hive provides the functionality of reading, writing, and managing large datasets residing in distributed storage. It runs SQL like queries called HQL (Hive query language) which gets internally converted to MapReduce jobs.

Using Hive, we can skip the requirement of the traditional approach of writing complex MapReduce programs. Hive supports Data Definition Language (DDL), Data Manipulation Language (DML), and User Defined Functions (UDF).

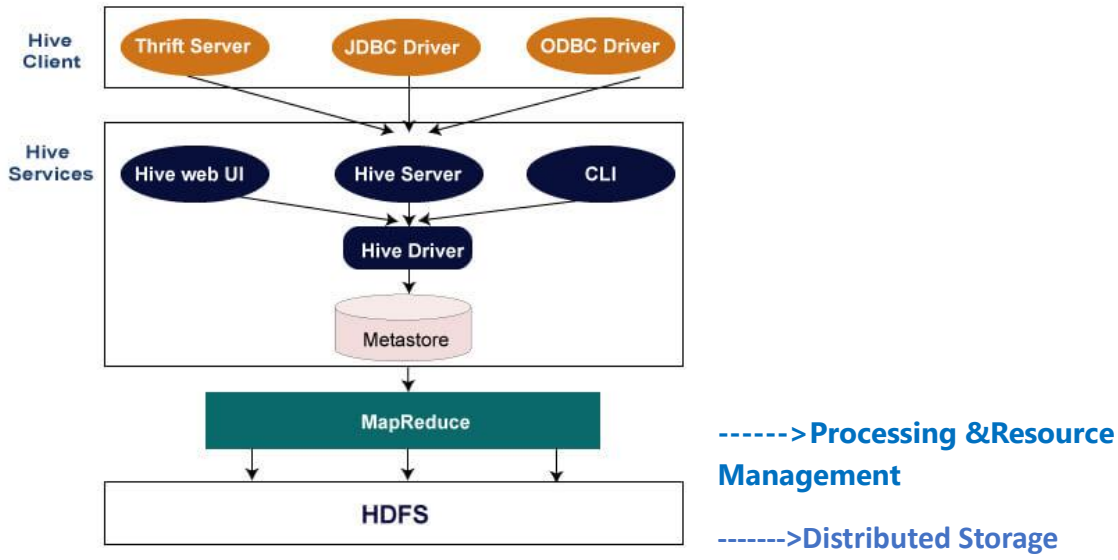
Features of Hive:

- Hive is fast and scalable.
- It provides SQL-like queries (i.e., HQL) that are implicitly transformed to MapReduce or Spark jobs.
- It is capable of analyzing large datasets stored in HDFS.
- It allows different storage types such as plain text, RCFile, and HBase.
- It uses indexing to accelerate queries.
- It can operate on compressed data stored in the Hadoop ecosystem.
- It supports user-defined functions (UDFs) where user can provide its functionality.

Limitations of Hive:

- Hive is not capable of handling real-time data.
- It is not designed for online transaction processing.
- Hive queries contain high latency.
- It doesn't support subqueries.

The major components of Hive and its interaction with the Hadoop is demonstrated in the figure below and all the components are described further:



User Interface (UI) –
As the name describes User interface provide an interface between user and hive. It enables user to submit queries and other operations to the system. Hive web UI, Hive command line, and Hive HD Insight (In windows server) are supported by the user interface.

Hive Client

Hive allows writing applications in various languages, including Java, Python, and C++. It supports different types of clients such as:-

- Thrift Server - It is a cross-language service provider platform that serves the request from all those programming languages that supports Thrift.
- JDBC Driver - It is used to establish a connection between hive and Java applications. The JDBC Driver is present in the class org.apache.hadoop.hive.jdbc.HiveDriver.
- ODBC Driver - It allows the applications that support the ODBC protocol to connect to Hive.

Hive Services

The following are the services provided by Hive: -

- Hive CLI - The Hive CLI (Command Line Interface) is a shell where we can execute Hive queries and commands.
- Hive Web User Interface - The Hive Web UI is just an alternative of Hive CLI. It provides a web-based GUI for executing Hive queries and commands.
- Hive MetaStore - It is a central repository that stores all the structure information of various tables and partitions in the warehouse. It also includes metadata of column and its type information, the serializers and deserializers which is used to read and write data and the corresponding HDFS files where the data is stored.
- Hive Server - It is referred to as Apache Thrift Server. It accepts the request from different clients and provides it to Hive Driver.
- Hive Driver - It receives queries from different sources like web UI, CLI, Thrift, and JDBC/ODBC driver. It transfers the queries to the compiler.
- Hive Compiler - The purpose of the compiler is to parse the query and perform semantic analysis on the different query blocks and expressions. It converts HiveQL statements into MapReduce jobs.
- Hive Execution Engine - Optimizer generates the logical plan in the form of DAG of map-reduce tasks and HDFS tasks. In the end, the execution engine executes the incoming tasks in the order of their dependencies.

Processing and Resource Management:

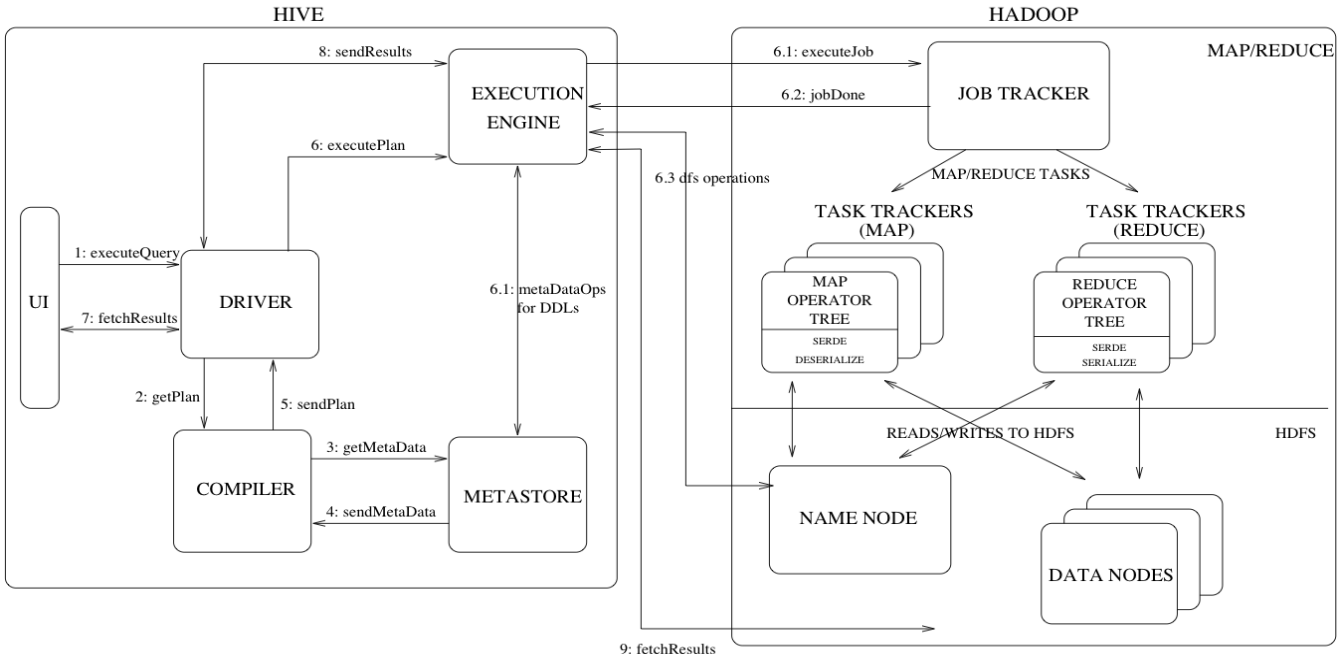
Hive uses a MapReduce framework as a default engine for performing the queries, because of that fact.

MapReduce frameworks are used to write large-scale applications that process a huge quantity of data in parallel on large clusters of commodity hardware. MapReduce tasks can split data into chunks, which are processed by map-reduce jobs.

Distributed Storage: Hive is based on Hadoop, which means that it uses the Hadoop Distributed File System for distributed storage.

Working with Hive:

job execution flow in Hive with Hadoop is demonstrated step by step.



- **Step-1: Execute Query –**
  Interface of the Hive such as Command Line or Web user interface delivers query to the driver to execute. In this, UI calls the execute interface to the driver such as ODBC or JDBC.

- **Step-2: Get Plan –**
  Driver designs a session handle for the query and transfer the query to the compiler to make execution plan. In other words, driver interacts with the compiler.

- **Step-3: Get Metadata –**
  In this, the compiler transfers the metadata request to any database and the compiler gets the necessary metadata from the metastore.

- **Step-4: Send Metadata –**
  Metastore transfers metadata as an acknowledgment to the compiler.

- **Step-5: Send Plan –**
  Compiler communicating with driver with the execution plan made by the compiler to execute the query.

- **Step-6: Execute Plan –**
  Execute plan is sent to the execution engine by the driver.

    - Execute Job
    - Job Done
    - Dfs operation (Metadata Operation)
- **Step-7: Fetch Results –**
  Fetching results from the driver to the user interface (UI).

- **Step-8: Send Results –**
  Result is transferred to the execution engine from the driver. Sending results to Execution engine. When the result is retrieved from data nodes to the execution engine, it returns the result to the driver and to user interface (UI).

Different Modes of Hive:

A hive can operate in two modes based on the number of data nodes in Hadoop.

1. Local mode
2. Map-reduce mode

When using Local mode:

1. We can run Hive in pseudo mode if Hadoop is installed under pseudo mode with one data node.

2. In this mode, we can have a data size of up to one machine as long as it is smaller in terms of physical size.
3. Smaller data sets will be processed rapidly on local machines due to the processing speed of small data sets.

When using Map Reduce mode:

1. In this type of setup, there are multiple data nodes, and data is distributed across different nodes. We use Hive in this scenario
2. It will be able to handle large amounts of data as well as parallel queries in order to execute them in a timely fashion.
3. By turning on this mode, you can increase the performance of data processing by processing large data sets with better performance.

Hive architecture in terms of interview:

Hive architecture (interview point of view)

- When a client submits a query, hive server receives the request, it transfers the request to the hive driver.
- Hive driver internally interacts with the compiler and checks for the syntactical errors.
- If everything looks good then it looks for a hive meta store and checks for the schema availability.
- If everything looks good, then it transfers the results/output to the driver program and it invokes a query execution plan.
- In the query execution plan, it checks it for the order of execution and prioritise/optimize the plan/order.
- This eventually generates a MapReduce job, which internally process the request by Map task and reduces the task by accessing the data set from the HDFS location.