1. Objective

Apache Hive is an open source data warehouse system built on top of **Hadoop**Haused for querying and analyzing large datasets stored in Hadoop files. It process structured and semi-structured data in Hadoop.

This Apache Hive tutorial explains basics of Apache Hive & Hive history in great details. In this hive tutorial, we will learn about the need for a hive and its characteristics. This Hive guide also covers internals of Hive architecture, Hive Features and Drawbacks of Apache Hive.

2. What is Hive?

Apache Hive is an open source data warehouse system built on top of Hadoop Haused for querying and analyzing large datasets stored in Hadoop files.

Initially, you have to write complex **Map-Reduce** jobs, but now with the help of Hive, you just need to submit merely **SQL** queries. Hive is mainly targeted towards users who are comfortable with SQL. Hive use language called **HiveQL** (HQL), which is similar to SQL. HiveQL automatically translates SQL-like queries into MapReduce jobs.

Hive abstracts the complexity of Hadoop. The main thing to notice is that there is no need to learn java for Hive.

The Hive generally runs on your workstation and converts your SQL query into a series of jobs for execution on a **Hadoop cluster**. Apache Hive organizes data into tables. This provides a means for attaching the structure to data stored in **HDFS**.

3. History of Hive

Data Infrastructure Team at Facebook developed Hive. Apache Hive is also one of the technologies that are being used to address the requirements at Facebook. It is very popular with all the users internally at Facebook. It is being used to run thousands of jobs on the cluster with hundreds of users, for a wide variety of applications.

Apache Hive-Hadoop cluster at Facebook stores more than 2PB of raw data. It regularly loads 15 TB of data on a daily basis.

Now it is being used and developed by a number of companies like Amazon, IBM, Yahoo, Netflix, Financial Industry Regulatory Authority (FINRA) and many others.

4. Why Apache Hive?

Let's us now discuss the need of Hive-

Facebook had faced a lot of challenges before implementation of Apache Hive. Challenges like the size of data being generated increased or exploded, making it very difficult to handle them. The traditional **RDBMS** could not handle the pressure. As a result, Facebook was looking out for better options. To overcome this problem, Facebook initially tried using **MapReduce**. But it has difficulty in programming and mandatory knowledge in SQL, making it an impractical solution. Hence, Apache Hive allowed them to overcome the challenges they were facing.

With Apache Hive, they are now able to perform the following:

- Schema flexibility and evolution
- Tables can be portioned and bucketed
- Apache Hive tables are defined directly in the HDFS
- JDBC/ODBC drivers are available

Apache Hive saves developers from writing complex Hadoop MapReduce jobs for ad-hoc requirements. Hence, hive provides summarization, analysis, and query of data. Hive is very fast and scalable. It is highly extensible. Since Apache Hive is similar to SQL, hence it becomes very easy for the SQL developers to learn and implement Hive Queries.

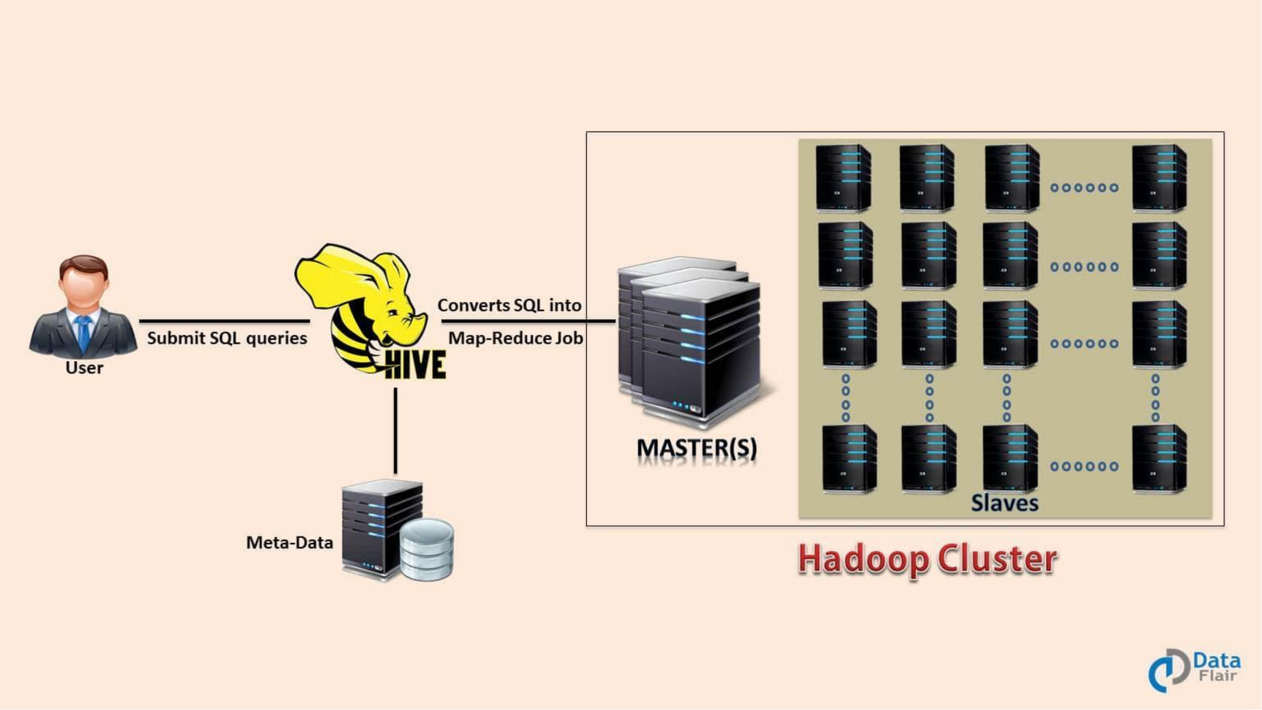
Hive reduces the complexity of MapReduce by providing an interface where the user can submit SQL queries. So, now business analysts can play with **Big Data** using Apache Hive and generate insights. It also provides file access on various data stores like HDFS and HBase. The most important feature of Apache Hive is that to learn Hive we don't have to learn Java.

5. Hive Architecture

After the introduction to Apache Hive, Now we are going to discuss the major component of Hive Architecture. The Apache Hive components are-

- **Metastore –** It stores metadata for each of the tables like their schema and location. Hive also includes the partition metadata. This helps the driver to track the progress of various data sets distributed over the cluster. It stores the data in a traditional RDBMS format. Hive metadata helps the driver to keep a track of the data and it is highly crucial. Backup server regularly replicates the data which it can retrieve in case of data loss.
- **Driver –** It acts like a controller which receives the HiveQL statements. The driver starts the execution of statement by creating sessions. It monitors the life cycle and progress of the execution. Driver stores the necessary metadata generated during the execution of a HiveQL statement. It also acts as a collection point of data or query result obtained after the Reduce operation.
- **Compiler –** It performs the compilation of the HiveQL query. This converts the query to an execution plan. The plan contains the tasks. It also contains steps needed to be performed by the MapReduce to get the output as translated by the query. The compiler in Hive converts the query to an **Abstract Syntax Tree (AST)**. First, check for compatibility and compile time errors, then converts the AST to a **Directed Acyclic Graph (DAG).**
- **Optimizer –** It performs various transformations on the execution plan to provide optimized DAG. It aggregates the transformations together, such as converting a pipeline of joins to a single join, for better performance. The optimizer can also split the tasks, such as applying a transformation on data before a reduce operation, to provide better performance.

- **Executor –** Once compilation and optimization complete, the executor executes the tasks. Executor takes care of pipelining the tasks.
- **CLI, UI, and Thrift Server –** CLI (command-line interface) provide a user interface for an external user to interact with Hive. Thrift server in Hive allows external clients to interact with Hive over a network, similar to the JDBC or ODBC protocols.



6. Hive Shell

The shell is the primary way with the help of which we interact with the Hive; we can issue our commands or queries in HiveQL inside the Hive shell. Hive Shell is almost similar to MySQL Shell. It is the command line interface for Hive. In Hive Shell users can run HQL queries. HiveQL is also case-insensitive (except for string comparisons) same as SQL.

We can run the Hive Shell in two modes which are: Non-Interactive mode and Interactive mode

- **Hive in Non-Interactive mode –** Hive Shell can be run in the non-interactive mode, with -f option we can specify the location of a file which contains HQL queries. For example- hive -f my-script.q
- **Hive in Interactive mode –** Hive Shell can also be run in the non-interactive mode. In this mode, we directly need to go to the hive shell and run the queries there. In hive shell, we can submit required queries manually and get the result. For example- $bin/hive, go to hive shell.

7. Features of Hive

There are so many features of Apache Hive. Let’s discuss them one by one-

- Hive provides data summarization, query, and analysis in much easier manner.
- Hive supports external tables which make it possible to process data without actually storing in HDFS.
- Apache Hive fits the low-level interface requirement of Hadoop perfectly.
- It also supports partitioning of data at the level of tables to improve performance.
- Hive has a rule based optimizer for optimizing logical plans.
- It is scalable, familiar, and extensible.
- Using HiveQL doesn’t require any knowledge of programming language, Knowledge of basic SQL query is enough.
- We can easily process structured data in Hadoop using Hive.
- Querying in Hive is very simple as it is similar to SQL.
- We can also run Ad-hoc queries for the data analysis using Hive.

8. Limitation of Hive

Hive has the following limitations-

- Apache does not offer real-time queries and row level updates.

- Hive also provides acceptable latency for interactive data browsing.
- It is not good for online transaction processing.
- Latency for Apache Hive queries is generally very high.

9. Conclusion

In Conclusion, Hive is a Data Warehousing package built on top of Hadoop used for data analysis. Hive also uses a language called **HiveQL** (HQL) which automatically translates SQL-like queries into MapReduce jobs. We have also learned various components of Hive like meta store, optimizer etc.