1. Objective

Apache Hive is an open source data warehouse system built on top of the **Hadoop** 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 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

Hive compiler converts the queries written in HiveQL into MapReduce jobs so that Hadoop developers do not need to worry much about the complex programming code beyond the processing and they can focus on the business problem. The three important functions performed by Hive include - data summarization, data querying and data analysis. Apache Hive is extensively used by data scientists and data analysts for

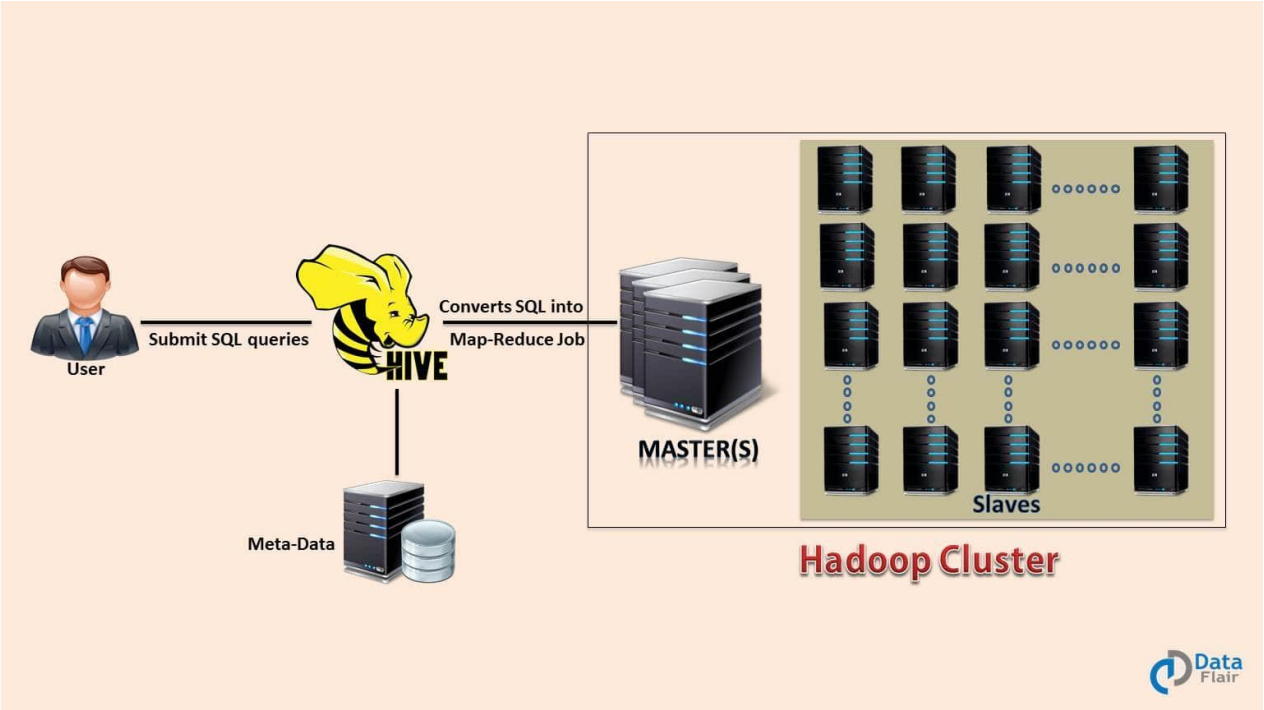
data exploration, building data pipelines and for processing ad-hoc queries.

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.

8. 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.

9. 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.

10: Hive Data Manipulation Language

A. Loading files into tables

B. Inserting data into Hive tables from queries

C. Inserting data into dynamic partitions

D. Writing data into files from queries

E. Enabling transactions in Hive

F. Inserting values into tables from SQL

G. Updating data

H. Deleting data

a: Loading files into tables

Loading data into a Hive table is one of the variants of inserting data into a Hive table. In this method, the entire file is copied/moved to a directory that corresponds to Hive tables. If the table is partitioned, then data is loaded into partitions one at a time. The general syntax of loading the data into a table is as follows:

```
LOAD DATA [LOCAL] INPATH 'filepath' [OVERWRITE] INTO TABLE tablename
[PARTITION (partcol1=val1, partcol2=val2 ...)]
```

Where:

- `[LOCAL]`: This is an optional clause. If this clause is specified, the preceding command will look for the file in the local filesystem. The command will follow the file path in the local filesystem.
- `FILEPATH`: This is the path where files reside either in the local filesystem or HDFS.
- `[OVERWRITE]`: Is an optional clause. If this clause is specified, the data in the table or partition is deleted and new data is loaded based on the file path in the statement.
- `tablename`: This is the name of the table.
- `[PARTITION (partcol1=val1, partcol2=val2 ...)]`: This is an optional clause for partitioned tables.

Note: If You Omit The Overwrite Clause While Creating A Hive Table, what Happens To File Which Are New And Files Which Already Exist?

Answer :
The new incoming files are just added to the target directory and the existing files are simply overwritten. Other files whose name does not match any of the incoming files will continue to exist.
If you add the OVERWRITE clause then all the existing data in the directory will be deleted before new data is written.

b: Inserting data into Hive tables from queries

This is another variant of inserting data into a Hive table. Data can be appended into a Hive table that already contains data. Data can also be overwritten in the Hive table. Data can also be inserted into multiple tables through a single statement only. The general format of inserting data into a table from queries is as follows:

Syntax

```
INSERT OVERWRITE TABLE tablename [PARTITION (partcol1=val1, partcol2=val2 ...) [IF NOT EXISTS]]
select select_statement FROM from_statement;
```

Where:

- `tablename`: This is the name of the table
- `OVERWRITE`: This is used to overwrite existing data in the table
- `[PARTITION (partcol1=val1]`: This option is used when data needs to be inserted into a partitioned table
- `[IF NOT EXISTS]`: This is an optional clause

The second syntax of inserting the data into a Hive table is as follows:

```
INSERT INTO TABLE tablename [PARTITION (partcol1=val1, partcol2=val2 ...)] select

select_statement FROM from_statement;
```

12: Show the table properties

TBLPROPERTIES

The TBLPROPERTIES clause allows you to tag the table definition with your own metadata key/value pairs. Some predefined table properties also exist, such as last_modified_user and last_modified_time which are automatically added and managed by Hive.

The TBLPROPERTIES clause is used to add the creator name while creating a table.

Other predefined table properties include:

- TBLPROPERTIES ("comment"="*table_comment*")
- TBLPROPERTIES ("hbase.table.name"="*table_name*")
- LPROPERTIES ("immutable"="true") or ("immutable"="false") in release 0.13.0+ (HIVE-6406)
- TBLPROPERTIES ("orc.compress"="ZLIB") or ("orc.compress"="SNAPPY") or ("orc.compress"="NONE") and other ORC properties – see ORC Files.
- TBLPROPERTIES ("transactional"="true") or ("transactional"="false") in release 0.14.0+, the default is "false" – see Hive Transactions.
- TBLPROPERTIES ("NO_AUTO_COMPACTION"="true") or ("NO_AUTO_COMPACTION"="false"), the default is "false" – see Hive Transactions.

- TBLPROPERTIES ("auto.purge"="true") or ("auto.purge"="false") in release 1.2.0+ (HIVE-9118) – see Drop Table and Drop Partitions.

For more information Click Here

Note: There is no way you can delete the DBPROPERTY.

The TBLPROPERTIES is added like –

```
TBLPROPERTIES('creator'= 'Joan')
```

```
SHOW TBLPROPERTIES tblname;
```

Example

Suppose one file "abc.csc" contain 3 lines of headers that we do not want to include in our Hive query. To skip header lines from our tables in Hive we can set a table property that will allow us to skip the header lines.

```
CREATE EXTERNAL TABLE userdata (
name STRING,
job STRING,
dob STRING,
id INT,
salary INT)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ' ' STORED AS TEXTFILE
LOCATION '/user/data'
TBLPROPERTIES("skip.header.line.count"="3");
```

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.

FAQ

Question: : What is the difference between LIKE and RLIKE operators in Hive?

Ans: The LIKE operator behaves the same way as the regular SQL operators used in select queries. Example –

street_name like '%Chi'

But the RLIKE operator uses more advance regular expressions which are available in java

Example – street_name RLIKE '.*(Chi|Oho).*' which will select any word which has either chi or oho in it.

:

Question: What kind of dataware house application is suitable for Hive?

Answer: Hive is not a full database. The design constraints and limitations of Hadoop and HDFS impose limits on what Hive can do. Hive is most suited for data warehouse applications, where

1) Relatively static data is analyzed,

2) Fast response times are not required, and

3) When the data is not changing rapidly.

Hive doesn't provide crucial features required for OLTP, Online Transaction Processing. It's closer to being an OLAP tool, Online Analytic Processing. So, Hive is best suited for data warehouse applications, where a large data set is maintained and mined for insights, reports, etc.

Question: What is the maximum size of string data type supported by Hive?

Answer: Maximum size is 2 GB.

Question Describe REVERSE function in Hive with example?

Answer: REVERSE function will reverse the characters in a string.

Question: LOWER or LCASE function in Hive with example?

Answer: LOWER or LCASE function will convert the input string to lower case characters.

Question: UPPER or UCASE function in Hive with example?

Answer: UPPER or UCASE function will convert the input string to upper case characters.

Question: Rename a table in Hive – How to do it?

Answer: Using ALTER command, we can rename a table in Hive.

ALTER TABLE hive_table_name RENAME TO new_name;

Question: Difference between order by and sort by in hive?

Answer: SORT BY will sort the data within each reducer. You can use any number of reducers for SORT BY operation.

ORDER BY will sort all of the data together, which has to pass through one reducer. Thus, ORDER BY in hive uses single reducer.

ORDER BY guarantees total order in the output while SORT BY only guarantees ordering of the rows within a reducer. If there is more than one reducer, SORT BY may give partially ordered final results

Question: RLIKE in Hive?

Answer: RLIKE (Right-Like) is a special function in Hive where if any substring of A matches with B then it evaluates to true. It also obeys Java regular expression pattern. Users don't need to put % symbol for a simple match in RLIKE.

Question : Can a partition be archived? What are the advantages and Disadvantages?

Answer: Yes. A partition can be archived. Advantage is it decreases the number of files stored in namenode and the archived file can be queried using hive. The disadvantage is it will cause less efficient query and does not offer any space savings.

Question: Does Hive support record level Insert, delete or update?

Answer: Hive does not provide record-level update, insert, or delete. Henceforth, Hive does not provide transactions too. However, users can go with CASE statements and built in functions of Hive to satisfy the above DML operations. Thus, a complex update query in a RDBMS may need many lines of code in Hive.

Questions: What is the functionality of Query Processor in Apached Hive ?

Ans: This component implements the processing framework for converting SQL to a graph of map/reduce jobs and the execution time framework to run those jobs in the order of dependencies.

Questions. Which classes are used by the Hive to Read and Write HDFS Files

Ans : Following classes are used by Hive to read and write HDFS files

•TextInputFormat/HiveIgnoreKeyTextOutputFormat: These 2 classes read/write data in plain text file format.

•SequenceFileInputFormat/SequenceFileOutputFormat: These 2 classes read/write data in hadoop SequenceFile format.

Question: What is the functionality of Query Processor in Apached Hive ?

Ans: This component implements the processing framework for converting SQL to a graph of map/reduce jobs and the execution time framework to run those jobs in the order of dependencies.

Question: What is Hive Metastore?

Ans : Hive metastore is a database that stores metadata about your Hive tables (eg. table name, column names and types, table location, storage handler being used, number of buckets in the table, sorting columns if any, partition columns if any, etc.). When you create a table, this metastore gets updated with the information related to the new table which gets queried when you issue queries on that table.

Question: Explain about the different types of join in Hive.

Answer: HiveQL has 4 different types of joins – JOIN- Similar to Outer Join in SQL

FULL OUTER JOIN – Combines the records of both the left and right outer tables that fulfil the join condition.

LEFT OUTER JOIN- All the rows from the left table are returned even if there are no matches in the right table.

RIGHT OUTER JOIN-All the rows from the right table are returned even if there are no matches in the left table.