Apache Hive Introduction & Architecture

What is Hive?

Hive is a component in Hadoop Stack. It is an open-source data warehouse tool that runs on top of Hadoop. It was developed by Facebook and later it is donated to the Apache foundation. It reads, writes, and manages big data tables stored in HDFS or other data sources.

Hive doesn't offer insert, delete and update operations but it is used to perform analytics, mining, and report generation on the large data warehouse. Hive uses Hive query language similar to SQL. Most of the syntax is similar to the MySQL database. It is used for OLAP (Online Analytical Processing) purposes.

Why we need Hive?

In the year 2006, Facebook was generating 10 GB of data per day and in 2007 its data increased by 1 TB per day. After few days, it is generating 15 TB of data per day. Initially, Facebook is using the Scribe server, Oracle database, and Python scripts for processing large data sets. As Facebook started gathering data then they shifted to Hadoop as its key tool for data analysis and processing.

Facebook is using Hadoop for managing its big data and facing problems for ETL operations because for each small operation they need to write the Java programs. They need a lot of Java resources that are difficult to find and Java is not easy to learn. So Facebook developed Hive which uses SQL-like syntaxes that are easy to learn and write. Hive makes it easy for people who know SQL just like other RDBMS tools.

Hive Features

The following are the features of the Hive.

1. It is a Data Warehousing tool.
2. It is used for enterprise data wrangling.
3. It uses the SQL-like language HiveQL or HQL. HQL is a non-procedural and declaration language.
4. It is used for OLAP operations. (OLAP (Online Analytical Processing) operations are a set of data manipulation techniques used in data warehouses and multidimensional databases to facilitate complex analytical queries and reporting.)
5. It increases productivity by reducing 100 lines of Java code into 4 lines of HQL queries.
6. It supports Table, Partition, and Bucket data structures.
7. It is built on top of Hadoop Distributed File System (HDFS)
8. Hive supports Tez, Spark, and MapReduce.

Hive Architecture

1. Shell/CLI: It is an interactive interface for writing queries.
2. Driver: Handle session, fetch and execute the operation
3. Compiler: Parse, Plan and optimize the code.
4. Execution: In this phase, MapReduce jobs are submitted to Hadoop. and jobs get executed.
5. Metastore: Meta Store is a central repository that stores the metadata. It keeps all the details about tables, partitions, and buckets.

Hive Query Language(HQL) Practice

Starting Hive

```
[cloudera@quickstart ~]$ hive
hive>
```

Database

`hive> show databases;`

```
hive> create database emp;
hive> use emp;
```

Create a database

`CREATE DATABASE students;`

Use the created database

`USE`

Tables

Add your dataset to cloudera or the hive server if using docker.

```
student_id,name,age,course
1,Alice,20,Math
2,Bob,22,History
3,Carol,21,Math
4,David,23,Science
5,Eve,20,History
```

You can name the above file as students.csv

```
CREATE TABLE student_table (
  student_id INT,
  name STRING,
  age INT,
  course STRING
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE
TBLPROPERTIES ("skip.header.line.count"="1");
```

Hive offers a variety of storage formats beyond TEXTFILE, each tailored to specific advantages and use cases. Here are some commonly employed options:

1. **ORC (Optimized Row Columnar):** ORC is exceptionally well-suited for handling vast datasets efficiently. Its columnar storage design optimizes compression and read performance, making it a top choice for large-scale data processing.
2. **Parquet:** Parquet is a versatile format known for its efficient storage and processing of columnar data. It's particularly effective for analytics workloads, providing both compression benefits and schema evolution capabilities.
3. **AVRO:** AVRO is a flexible and schema-centric format, making it a great choice when schema evolution or compatibility is crucial. It supports complex data types and is often used in scenarios requiring schema flexibility.

By leveraging these storage formats in Hive, you can tailor your data storage to the specific requirements of your data processing tasks, optimizing performance and efficiency as needed.

Load Data into the Table

```
LOAD DATA LOCAL INPATH '/home/cloudera/Desktop/hive/student.csv' INTO TABLE
student_table;
```

`SELECT * FROM student_table;`

Query to find average age for each course

```
SELECT course, AVG(age) AS average_age
FROM student_table
GROUP BY course;
```
