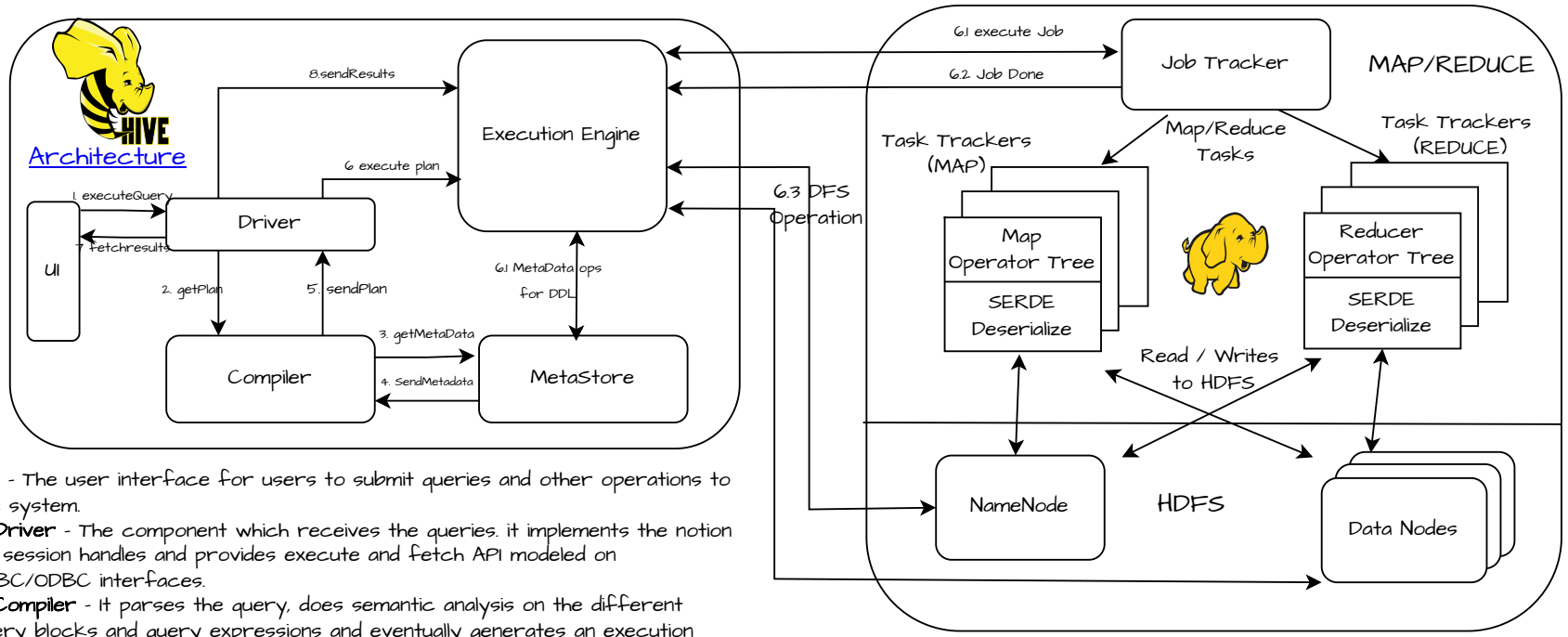
Hive is a data warehousing infrastructure based on Apache Hadoop. it facilitates reading, writing, and managing large datasets residing in distrubuted storage and queried using SQL syntax.

[Interview Questions Data Flair]

[More Interview Questions]



1. UI - The user interface for users to submit queries and other operations to the system.

2. Driver - The component which receives the queries. it implements the notion of session handles and provides execute and fetch API modeled on JDBC/ODBC interfaces.

3. Compiler - It parses the query, does semantic analysis on the different query blocks and query expressions and eventually generates an execution plan with the help of table and partition metadata looked up from the metastore

4. MetaStore - it stores all the structure information of the various tables and partitions in the warehouse including column and column type information the serializers and deserializers necessary to read and write data and the corresponding HDFS files where the data is stored.

5. Execution Engine - It executes the execution plan created by the compiler. The plan is DAG of stages. The Execution engine manages the dependencies b/w these different of the plan and executes these stages on the appropriate system components.

DDL(Data Definition Language) operations

Creating Tables

hive > CREATE TABLE pokes (foo INT, bar STRING)

Browsing through tables

hive > SHOW TABLES;

hive > DESCRIBE <table_name>

Altering & Dropping tables

hive > ALTER TABLE events RENAME to <name>;

hive > ALTER TABLE pokes ADD COLUMNS (new_col INT);

hive > ALTER TABLE invites REPLACE COLUMNS (foo INT);

// only keeps the first column

hive > DROP TABLE <table_name>;

DML (Data Manipulation Language) Operations

Loading data from flat files into HIVE

hive > LOAD DATA LOCAL INPATH './example/files/xf.txt'
>OVERWRITE INTO TABLE pokes;

OVERWRITE signifies that existing data in the table is deleted.
if OVERWRITE is ommited,
datafiles are appended to existing data sets.

Notes:

* No verification of data against the schema is performed by the load command
* If the file exists in HDFS, it is moved into Hive-Controller file system namespace.
* we should create the directory before trying to create tabels via HIVE

hive > LOAD DATA LOCAL INPATH '/example path/'
>OVERWRITE INTO pokes PARTITION (ds='a');

In the above statement, the TABLE pokes must be created as partitioned by the key ds for this command to work

1. What is Hive?

-> Hive allows users to read, write and manage large amount of data using SQL. Hive is built on top of Apache Hadoop, which is open-source framework used to efficiently store and process large datasets. Hive is built on top of Apache Hadoop, which is an open source framework used to efficiently store and process large datasets. Hive is closely integrated with Hadoop, and is designed to work quickly on large amount of data. Hive's unique ability is its ability to query large datasets, leveraging Apache Tez or MapReduce, with a SQL-like interface.

2. Hive MetaStore?

-> The hive metastore is simply a relational database. It stores metadata related to the tables schemes you create query big data stored in HDFS. When you create a new Hive table, the information related to the scheme(col names, datatypes) is stored in the Hive MetaStore relational database.

MetaStore provides data abstraction, and data discovery.. Hive accomplishes both these features by providing a metaadata repositary that is tightly integrated with the Hive processing system so that data and metadata are in sync The metaStore can be configured to be used in a couple ways: remote and embedded. In remote mode Thrift Service. In embeded mode, the Hive client directly connects to an underlying metastore using JDBC. This mode is useful because it avoids another system that needs to be maintained and monitored. The Thrift provides an interface to manipulate and query Hiive metadata. Thrift provides bindings in many popular languages and third party tools.

3. Types of Data Loads in Hive?

-> * Edge Node load (edge node to Hive * HDFS to Hive

4. Types of tables in Hive?

-> **Manged Tables.**

In a managed Table, both the table's data and schema are managed by Hive. The data will be located in a folder named after the table within the Hive data Warehouse. which is essentially a file in HDFS. The location is user-configurable when Hive is installed. By managed or controlled we mean that if you drop a managed table Hive will delete both the schema and the files which have the table data. Default location is

/user/hive/warehouse/

-> External Tables.

An external table is one where only the table schema is controlled by Hive. In most cases the user will set up the folder location within HDFS and copy the data file(s). When an external table is deleted, Hive will only delete the schema associated with the table. The data files are not affected

MANAGED TABLE	EXTERNAL TABLE
* Hive manages the data and the Schema	* Hive doesnot manage the data it only handles the schema
* If the table or part of the table is dropped the data corresponding to that will also be deleted	* Dropping the table doesn't delete the data, but it does clear the metadata of that table
* Hive stores the data into its warehouse directory	* Hive stores the data in the Location specified during execution (generally not in warehouse)
* It provides ACID/transactional support (Atomicity, consistency, Isolation, Durability)	* It doesn't provide ACID/transactional support
* Used for Temporary tables, intermediate results, and tables where data is managed by Hive only	* Tables to point to the existing data, tables shared with other systems

Feature	Managed Table	External Table
Data Ownership	Hive	External System
Data location	Hive Warehouse	User-specified
Lifecycle Management	Hive	External system
Schema Evolution	Supported	Limited
Data Deletion	Deleted with table	Not Deleted with table

5. Hive Warehouse Directory?

-> The Hive warehouse directory is the default storage location where Hive stores Data for Managed tables, It's typically located at /user/hive/warehouse in HDFS When you create a managed table in Hive, it automatically created subdirectory for that table within the warehouse directory.

6. HIVE vs SQL?

HIVE	SQL
* Hive is not a Database as it doesn't hold the physical data. The Actual data is stored in HDFS and not in Hive. Hive is just a tool working on top of Hadoop/HDFS to query data as per user demand.	* SQL is based on relational Database model & it uses a pure database in which the physical data actually resides in tables & adheres to all the RDBMS and ACID rules.
* ELT(Extract Load Transform)	* ETL(Extract Transform Load)
* Query Time Parsing (Schema on road)	* Load time parsing (Schema on Write)
* Good serialized Data support	* No Serialized Data support
* Good for large Data	* Good for small Data
* OLAP(online Analytical processing) Historical Data for analysis and consistency	* OLTP (online Transaction processing) Current, real-time data for Efficiency
* Limited ACID support disabled by default	* Fully ACID compliant
* Support for Collection Datatypes(array, maps, structs)	* Limited to basic Data types (generally)
* Multiple execution engines (Tez, MR, Spark)	* Single Execution engine
* Unstructured / SemiStructured Data	* Structured Data.

7. SORT BY vs ORDER BY?

->The ORDER BY sorts the data it uses one reducer to sort the final output. If the number of rows in the output is too large, the single reducer could take a very long time to finish.

->The SORT BY uses the columns to sort the rows before feeding the rows to a reducer. The sort order will be dependent on the column types. If the columns is of numeric type, then the sort order is numeric order, if the columns is of string type, then the sort order is lexicographical order. SORT BY is more suitable for large datasets it uses multiple reducers partially sorting array at the data nodes, where as the ORDER BY only uses one reducer for large data ORDER BY can be slow

8, Default delimitter in Hive ? -> ^A

9. Schema Evolution in Hive?

-> Schema Evolution is supported in Hive provided the table is created with Avro File format as we have **.avsc** file manages the schema. Avro is a data serialization system that provides a compact, fast, binary data format. It supports schema evolution. meaning you can change the schema of data over time without breaking compatibility with older data. It uses the **.avsc** file to define the schema structure of the data. Hive supports Avro as a storage format and can leverage its schema evolution capabilities.

10. Partitions in Hive?

-> Partitioning is a technique to divide a large table into smaller, more manageable subsets based on one or more columns. These columns are called partition columns. Each partition is stored as a seperate directory under the table's location. The column is choosed on which has the lowest cardinalcolumns (low variations) are chosen for partitioning.

* Static Partitioning: Partitions are created manually before data loading. Suitable for large datasets where partition values are know in advance.Provides better control over data organization.
* Dynamic Partitioning: partitions are created automatically during data load. Suitable for scenarios where partition values are generated dynamically

11. What is Bucketing?

-> Bucketing is a data organization technique that divides a table into smaller, more manageable units called buckets. Bucketing uses a hashing function to distribute data evenly across buckets. A Hash function is applied to the columns to generate a hash value. The hash value is used to determine the bucket which the data will be stored based on the hash value.

Feature	Partitioning	Bucketing
* Division	Column values	Hash Function
* HDFS structure	Seperate directories	inside partion as files
* Uses	Data is Diverse and Cardinal	Uniform Datasets
* Benefits	Reduced Data Scan	Better Joins & faster Random Access

12. Default Partition size?
-> Hive create default partition of 100. To increase the number of partitions
"hive.exec.max.dynamic.partitions"
13. DML in Hive?
-> DML(Data Manipulation language) in hive is possible when the table is of orc fileformat, should be partitioned and bucketed, and should be set transactional to be true.
14. What is MSCK repair in Hive?
-> The MSCK REPAIR TABLE command is used to manually add partitions that are added or removed from the file system, such as HDFS or S3, but are not present in the metastore.
* This task assumes that the table is external table that stores partitions outside the warehouse. You remove one of the partition directories on the file system. This action renders the metastore inconsistent with the file system. You repair the discrepancy manually to synchronize the metastore with the file system such as HDFS.
* When you perform the MSCK repair the metastore would be updated then any other table which uses the same data doesn't have to perform the partitions it can use the partitions already existing picked up from the metastore
15. Performance Tuning in Hive?
* Vectorization : query performance of all operations like scans, aggregations, filters and joins, by performing them in batches of 1024 rows at once instead of a single row each time.
* Parallel Execution : A specific job may contain multiple stages which may not be completely iterdependent. Instead of running a single stage at one instance executing these non-interdependent stages in parallel can drastically reduce the run time.
* Partition & Bucketing & File Formats
* Optimize JOINS : For larger Datasets then it is not advisable to just use normal joins, there are many other JOINS like Map Join, Bucket Join, etc,.
* Map Join : It determines the smaller table and the table itslef is loaded into the memory of each mapper, the loaded data is mapped into a hash-table & joins are performed with the bigger table
* TEZ / Spark Engine, ORC file format.
* Cost-Based Optimizer(CBO) generates more efficient query plans based on the statistics about the data(column and table level)
16 Sort Merget Bucket Join?
->Both tables must have the same bucketed column and should have the same number of buckets. Tables are then sorted on the bucketed column within each bucket. The mapper reads the corresponding buckets from both tables the joined data is then passed to the reducer for further processing.
17. Hive Indexing?
-> Hive indexing creates a seperate table that contains information about the indexed columns and pointers to the corresponding rows in the main table. When a query involves an indexed column, Hive can leverage the index to efficiently locate relevant data, reducing the amount of data scanned. The table is a sorted list-based table(Compact Index) with key as the column value and Value as the pointer to the corresponding row in the main table for efficient lookups other types of indexing is Bitmap, Bitarray
18. Hive View?
-> A view allows a query to be saved and treated like a table. It is a virtual table that presents a result set based on the result of the query. It doesn't physicallystore data like a table, it's just a logical representation of data.
19. Limitations of Hive?
-> * Not suitable for Unstructured (data skew) or small Data
* High Latency(MR) & Limited real-time processing
* Limited DML Operations (designed primarily for Reads)
20. Rename Table & change column Data type?
-> Rename

```
ALTER TABLE db.test RENAME TO db.rename;|
```

Data type changes:

```
ALTER TABLE tableA CHANGE <col_name> BIGINT;
```

21. SERDE in Hive?

-> SERDE is the acronym for Serializer/Deserializer. Hive uses the SerDe interface for IO. The interface handles both serialization and deserialization. A SerDe allows Hive to read in data from a table, and write it back out to HDFS in any custom format. If we want to we can use any or our own SerDe.

22. UDF in Hive?

-> User Defined Function(UDF) in Hive is custom function which we can create to perform a specific operation on data that is not available in Hive's built-in functions. To extend Hive's functionality, allowing us to process data in ways that are specific to our application requirements.

23. Hive in shell?

-> We can use the sql or hql scripts and execute them with the command

hive -f my_script.sql

with direct shellscripting file with .sh extension we can use this command in the file and run that file

hive -e "query" (it needs to be in a single line)

24. Thrift server?

-> Metastore provides a Thrift Interface to manipulate and query Hive Metadata. Thrift provides bindings in many popular languages. Third party tools can use this interface to integrate Hive metadata into other business metadata repositaries.

-> In remote mode Thrift Service.The Thrift provides an interface to manipulate and query Hiive metadata.

Thrift provides bindings in many popular languages and third party tools.

practice Questions

1. Create a external tables of different file formats?

```
CREATE EXTERNAL TABLE table_name (
column1 data_type,
column2 data_type,
...
)
ROW FORMAT DELIMITED -- or SERDE(schema format)
FIELDS TERMINATED BY ',' -- for text files
STORED AS FILE_FORMAT -- e.g., PARQUET, AVRO, ORC, TEXTFILE
TBLPROPERTIES('avro.schema.url'='hdfs:///path/to/your/schema.avsc') --  only for avro if .avsc file exists
LOCATION '/path/to/directory/';
```

2. Improve the query performance by creating paritioned tables in the Hive metastore.

```
-> create external table < table_name > (cols.....)
   partitioned by (cols to be partitioned)
   stored as file format
   location 'location';
```

3. creating a hive table from a avro data file and an avro schema

```
create external table orders_avro (
        order_id int,
        order_date string,
        order_customer_id int,
        order_status string
)
stored as avro
location '/user/cloudera/output/retail_db/orders_avro'
tblproperties('avro.schema.url'='/user/cloudera/output/retail_db/schema/orders.avsc');
```

4. Enable dynamic partition

```
set hive.exec.dynamic.partition.mode=nonstrict
```

1. What is Hive
2. Hive architecture
3. Hive Metastore
4.types of data load in Hive
5. Types of tables and differences
6. Hive warehouse
7. Hive vs SQL
8. Sort By vs Order By
9. Default delimiter of Hive
10. Schema evolution in hive
11. Partitions and its type
12. Bucketing
- Partitioning vs Bucketing
13. Default partition size
14. DML in Hive?
15. MSCK repair in HIVE
16. Performace tuning in HIVE
17. sort merget bucket join
18. Indexing
19. Hive views
20. Limitations of Hive
21. Rename table in Hive
22. Change datatype of column in Hive
23. Serde in hive
24. UDFS in hive
25. hive Unix run
26. Thrift server
27. OLTP VS OLAP
28. types of MetaStore
29. Collection Data types
30. UDF AND UDAF
31.