**#Assignment\_23.3**

**Q1-Explain hive architecture and components in brief.**

**SOLUTION:**



Primarily The diagram represents CLI (Command Line Interface), JDBC/ODBC and Web GUI (Web Graphical User Interface).This represents when user comes with CLI (Hive Terminal) it directly connected to Hive Drivers, When User comes with JDBC/ODBC (JDBC Program) at that time by using API (Thrift Server) it connected to Hive driver and when the user comes with Web GUI (Ambari server) it directly connected to Hive Driver

The hive driver receives the query from user and send it to Hadoop architecture. The Hadoop architecture uses name node, data node, task tracker and job tracker to receive query and divide work.

Typical query flows through the system:

**Step 1:-**The UI calls the execute interface to the Driver

**Step 2:-**The Driver creates a session handle for the query and sends the query to the compiler to generate an execution plan

**Step 3&4:-**The compiler needs the metadata so send a request for getMetaData and receives the sendMetaData request from MetaStore.

**Step 5:-**This metadata is used to type check the expressions in the query tree as well as to prune partitions based on query predicates. The plan generated by the compiler is a DAG of stages with each stage being either a map/reduce job, a metadata operation or an operation on HDFS. For map/reduce stages, the plan contains map operator trees (operator trees that are executed on the mappers) and a reduce operator tree (for operations that need reducers).

**Step 6:-**The execution engine submits these stages to appropriate components (steps 6, 6.1, 6.2 and 6.3). In each task (mapper/reducer) the deserializer associated with the table or intermediate outputs is used to read the rows from HDFS files and these are passed through the associated operator tree. Once the output generate it is written to a temporary HDFS file though the serializer. The temporary files are used to provide the subsequent map/reduce stages of the plan. For DML operations the final temporary file is moved to the table’s location

**Step 7&8&9:-**For queries, the contents of the temporary file are read by the execution engine directly from HDFS as part of the fetch call from the Driver

**HIVE COMPONENTS**

**UI: -**UI means User Interface, It allows users to submit their queries and other operations to the system.

**Megastore: -**It stores all the information related to structure of the various tables and partitions that are present in warehouse including column and column type information, the serializers and deserializers necessary to read and write data and the respective HDFS files where the data is stored.

**Compiler: -**It is used to parses the query, does semantic analysis is performed on different query blocks and query expressions and that generates an execution plan with the help of the table and partition metadata looked up from the metastore.

**Execution Engine: -**The execution plan created by compiler is executed by executive engine. The plan is a DAG of stages. The execution engine manages the dependencies between various stages of the store plan and execute them on correct system component.

**Driver: -**Driver receives the queries from UI. This component implements the notion of session handles and provides execute and fetch APIs modeled on JDBC/ODBC interfaces.