



Working of Hive

Step 1: executeQuery: The user interface calls the execute interface to the driver.

Step 2: getPlan: The driver accepts the query, creates a session handle for the query, and passes the query to the compiler for generating the execution plan.

Step 3: getMetaData: The compiler sends the metadata request to the metastore.

Step 4: sendMetaData: The metastore sends the metadata to the compiler.

The compiler uses this metadata for performing type-checking and semantic analysis on the expressions in the query tree. The compiler then generates the execution plan (Directed acyclic Graph). For Map Reduce jobs, the plan contains map operator trees (operator trees which are executed on mapper) and reduce operator tree (operator trees which are executed on reducer).

Step 5: sendPlan: The compiler then sends the generated execution plan to the driver.

Step 6: executePlan: After receiving the execution plan from compiler, driver sends the execution plan to the execution engine for executing the plan.

Step 7: submit job to MapReduce: The execution engine then sends these stages of DAG to appropriate components.

For each task, either mapper or reducer, the deserializer associated with a table or intermediate output is used in order to read the rows from HDFS files. These are then passed through the associated operator tree.

Once the output gets generated, it is then written to the HDFS temporary file through the serializer. These temporary HDFS files are then used to provide data to the subsequent map/reduce stages of the plan.

For DML operations, the final temporary file is then moved to the table's location.

Step 8,9,10: sendResult: Now for queries, the execution engine reads the contents of the temporary files directly from HDFS as part of a fetch call from the driver. The driver then sends results to the Hive interface.