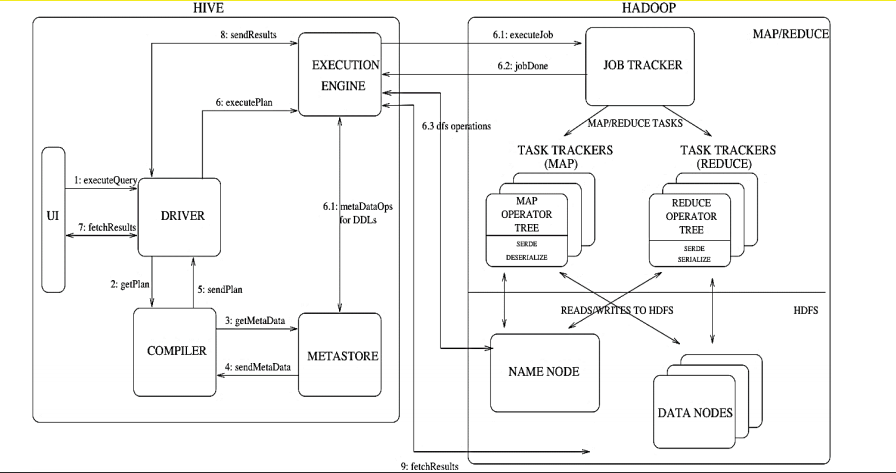
**Problem Statement**

1. **Explain Hive Architecture in Brief.**

**Hive Architecture-**

****

Hive Architecture is explained in following steps-

**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 typecheck 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. 7 Architecture (Contd.)

**Step 6-** The execution engine submits these stages to appropriate components. Once the output is generated, it is written to a temporary HDFS file though the serializer. 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.

1. **Explain Hive Components in Brief.**

Hive has following components-

1. **User Interface-**

* The user interface for users to submit queries and other operations to the system.
* For example, command line interface and a web based GUI.

1. **Driver**-

* The component which receives the queries.
* This component implements the notion of session handles and provides execute and fetch APIs modelled on JDBC/ODBC interfaces.

1. **Compiler**-

* The component that parses the query, does semantic analysis on the different query blocks and query expressions and eventually generates an execution plan with the help of the table and partition metadata looked up from the metastore.

1. **Metastore**-

* The component that 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.

1. **Execution Engine-**

* The component which executes the execution plan created by the compiler. The plan is a DAG of stages. The execution engine manages the dependencies between these different stages of the plan and executes these stages on the appropriate system components.

