Hive

* Why Hive ?

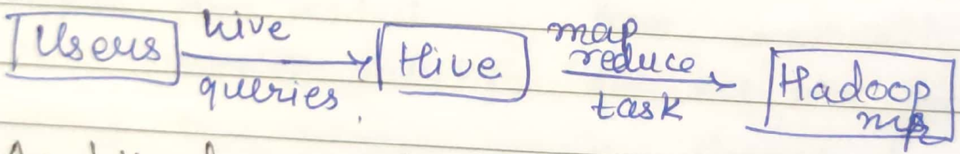
Problem → For processing & analyzing data users found it difficult to code as not all of them were well versed with coding lang.

Solution → users required a lang similar to SQL which was well known to all → Hive QL.

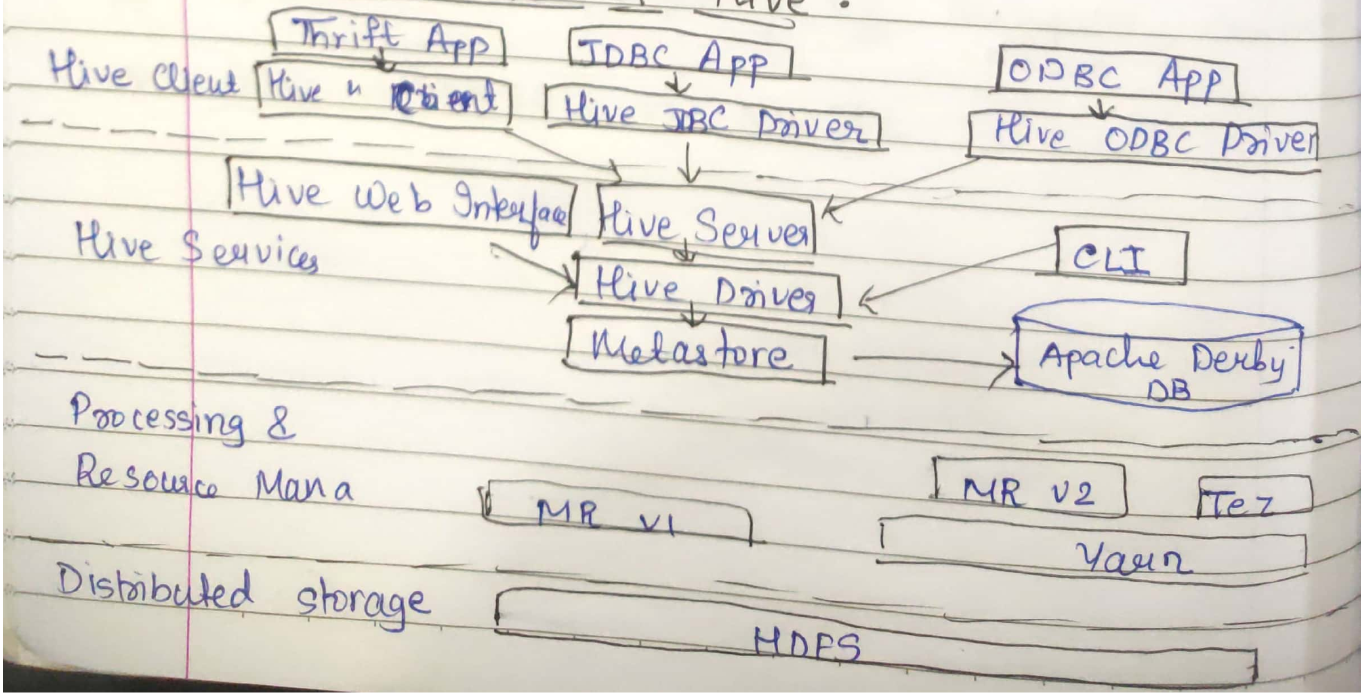
* What is Hive ?

→ Hive is a ~~dta~~ warehouse sys which is used for quering & analyzing large datasets stored in HDFS.

→ Hive uses a query language call HivQL similar to SQL.



* Architecture of Hive :



→ Hive Client supports different types of client app in diff lang for performing queries

→ Thrift is a s/f framework. Hive server is based on thrift, so it can serve the req from all the programming lang that supports thrift

→ JDBC Java DB Connectivity app is connected through JDBC driver

→ ODBC Open DB Connectivity app is connected through ODBC driver.

→ All the client requests are submitted to the Hive Server

→ GUI is provided to execute Hive queries

→ Commands are executed directly on CLI.

→ Hive driver is responsible for all the queries submitted

Steps:

1) Compiler: Hive driver passes query to compiler where it is checked & analyzed

2) Optimizer: Optimized logical plan in the form of a graph of MR & HDFS tasks is obtained

3) Executor: the tasks are executed.

→ Metastore is a repository for Hive metadata: Store metadata for hive tables.

→ Hive uses MR framework to process queries

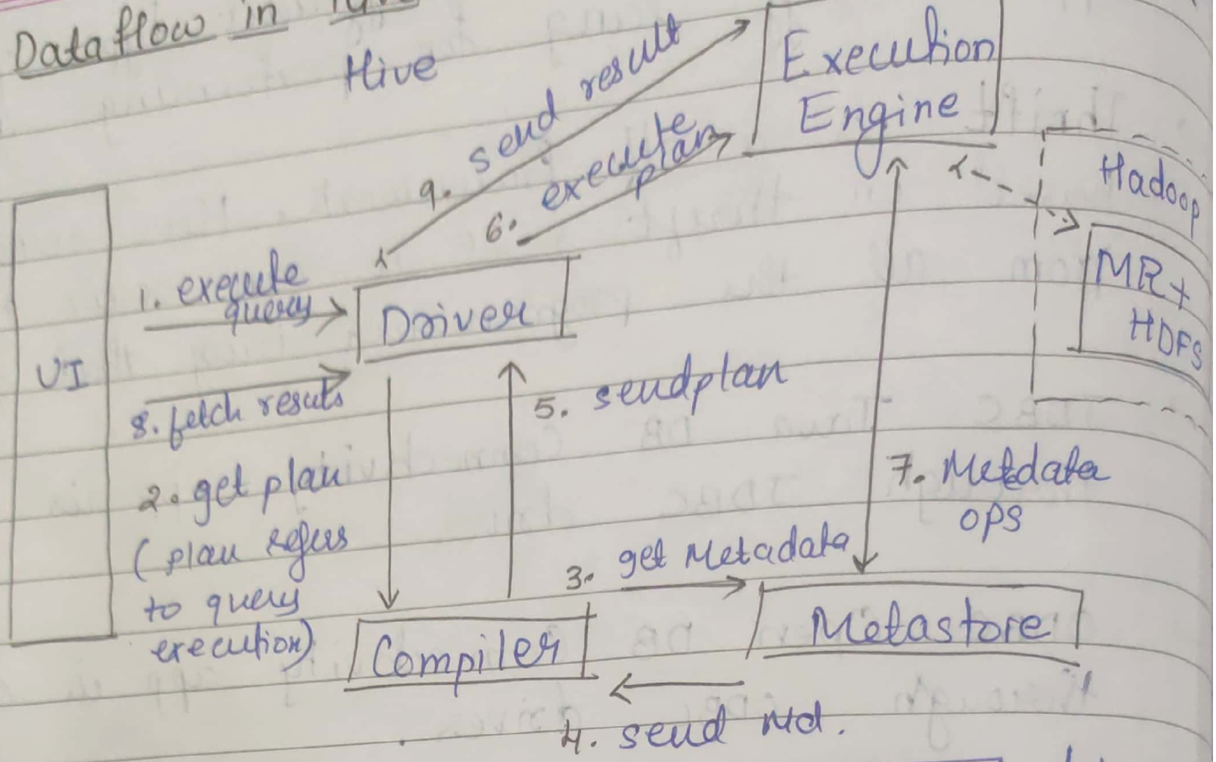
→ default loc for Hive metastore db

→ to keep metadata db files isolated for each instance of Hive

→ can be useful while running multiple instances of Hive on same mac, each working with its own tables & metadata

→ default db for metastore is Derby.

* Dataflow in Hive :



* Hive data Modeling :

↓

Tables

~~are~~ in hive are created the same way it is done in RDMS

→ Partitions

tables are organized into partitions for grouping same type of data based on partition key

→ Buckets

data present in partition can be further divided into buckets for efficient querying

* Modes of Hive :

↓

Local Mode

→ one data node in hadoop & data is small

→ processing is fast on smaller dataset which are present in local mac

→ MR Mode

→ multiple datanode in hadoop & data uis spread across various nodes

→ processing large datasets can be more efficient.

Hive	RDMS
→ schema on read	→ on write
→ data size in PB	→ TB
→ write once read many	→ write & read many times
→ data ware house	→ database based on relational model
→ easily scalable at low cost	→ not scalable at low cost.

* Features :

→ Tables are used similar to RDMS so easy to understand.

→ using hive QL, ~~query~~ multiple users can simultaneously query data

→ Hive QL easier than long codes

→ supports variety of data formats

* Appli :-

→ Batch processing rather than real-time proce

→ Data warehousing
organize & structure large volumes of data stored in HDFS

→ Large - scale Data Analysis
when data is distributed ~~over~~ across Hadoop cluster

→ Ad Hoc Queries :