# Hadoop

Hadoop is an open-source framework to store and process Big Data in a distributed environment. It contains two modules, one is Map Reduce and another is Hadoop Distributed File System (HDFS).

**Map Reduce:** It is a parallel programming model for processing large amounts of structured, semi-structured, and unstructured data on large clusters of commodity hardware.

**HDFS:** Hadoop Distributed File System is a part of Hadoop framework, used to store and process the datasets. It provides a fault-tolerant file system to run on commodity hardware.

# Executing Maping Reduce

**There are various ways to execute Map Reduce operations:**

- The traditional approach using Java Map Reduce program for structured, semi-structured, and unstructured data.
- The scripting approach for Map Reduce to process structured and semi structured data using Pig.
- The Hive Query Language (HiveQL or HQL) for Map Reduce to process structured data using Hive.

# Cont…

The Hadoop ecosystem contains different sub-projects (tools) such as Sqoop, Pig, and Hive that are used to help Hadoop modules.

**Sqoop**: It is used to import and export data to and fro between HDFS and RDBMS.

**Pig**: It is a procedural language platform used to develop a script for Map Reduce operations.

**Hive**: It is a platform used to develop SQL type scripts to do Map Reduce operations.

# What is Hive?

- Hive is a data warehouse infrastructure tool to process structured data in Hadoop.
- It resides on top of Hadoop to summarize Big Data, and makes querying and analyzing easy.
- Initially Hive was developed by Facebook, later the Apache Software Foundation took it up and developed it further as an open source under the name Apache Hive.
- It is used by different companies. For example, Amazon uses it in Amazon Elastic MapReduce.

# Cont…

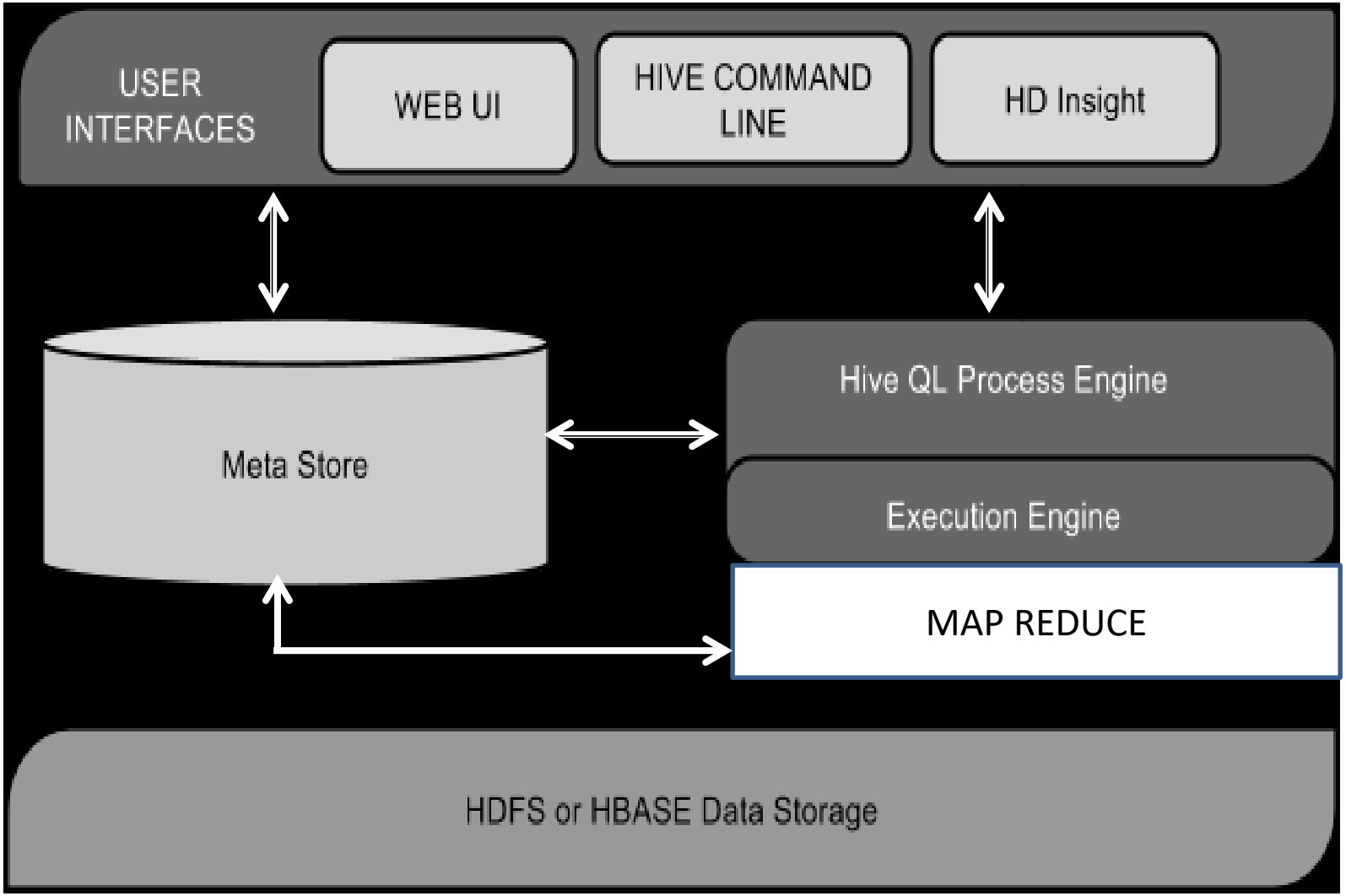
- **Hive is not**
  - A relational database
  - A design for OnLine Transaction Processing (OLTP)
  - A language for real-time queries and row-level updates

# Features of Hive

Here are the features of Hive:

- It stores schema in a database and processed data into HDFS.
- It is designed for OLAP.
- It provides SQL type language for querying called HiveQL or HQL.
- It is familiar, fast, scalable, and extensible.

# Architecture



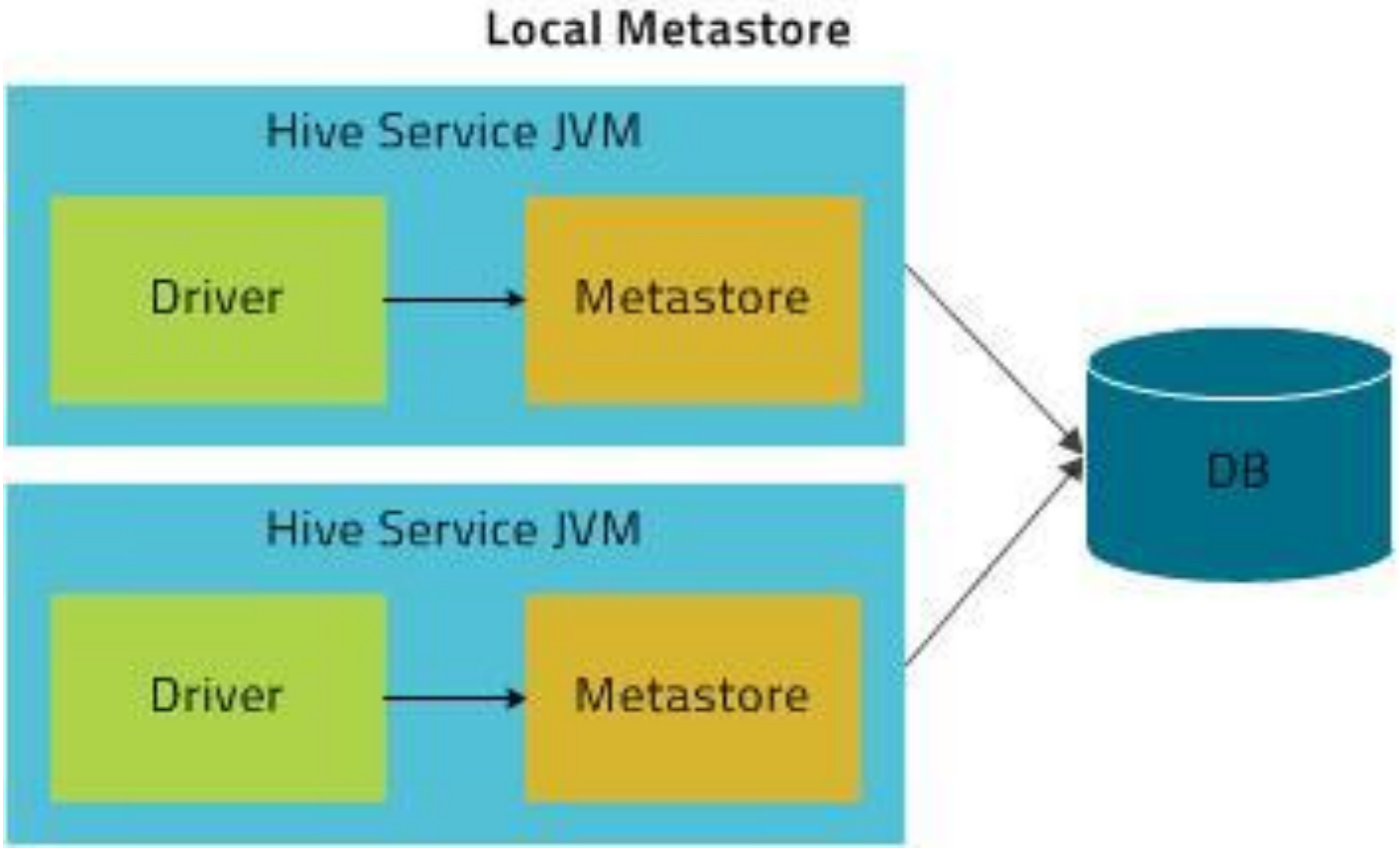
# Explanation

**User Interface**

Hive is a data warehouse infrastructure software that can create interaction between user and HDFS. The user interfaces that Hive supports are Hive Web UI, Hive command line, and Hive HD Insight (In Windows server).

**Meta Store**

Hive chooses respective database servers to store the schema or Metadata of tables, databases, columns in a table, their data types, and HDFS mapping.




In Local mode, the Hive metastore service runs in the same process as the main HiveServer process, but the metastore database runs in a separate process, and can be on a separate host. The embedded metastore service communicates with the metastore database over JDBC.

# Cont...

**HiveQL Process Engine**

HiveQL is similar to SQL for querying on schema info on the Metastore. It is one of the replacements of traditional approach for Map Reduce program. Instead of writing MapReduce program in Java, we can write a query for MapReduce job and process it.

**HDFS or HBASE**

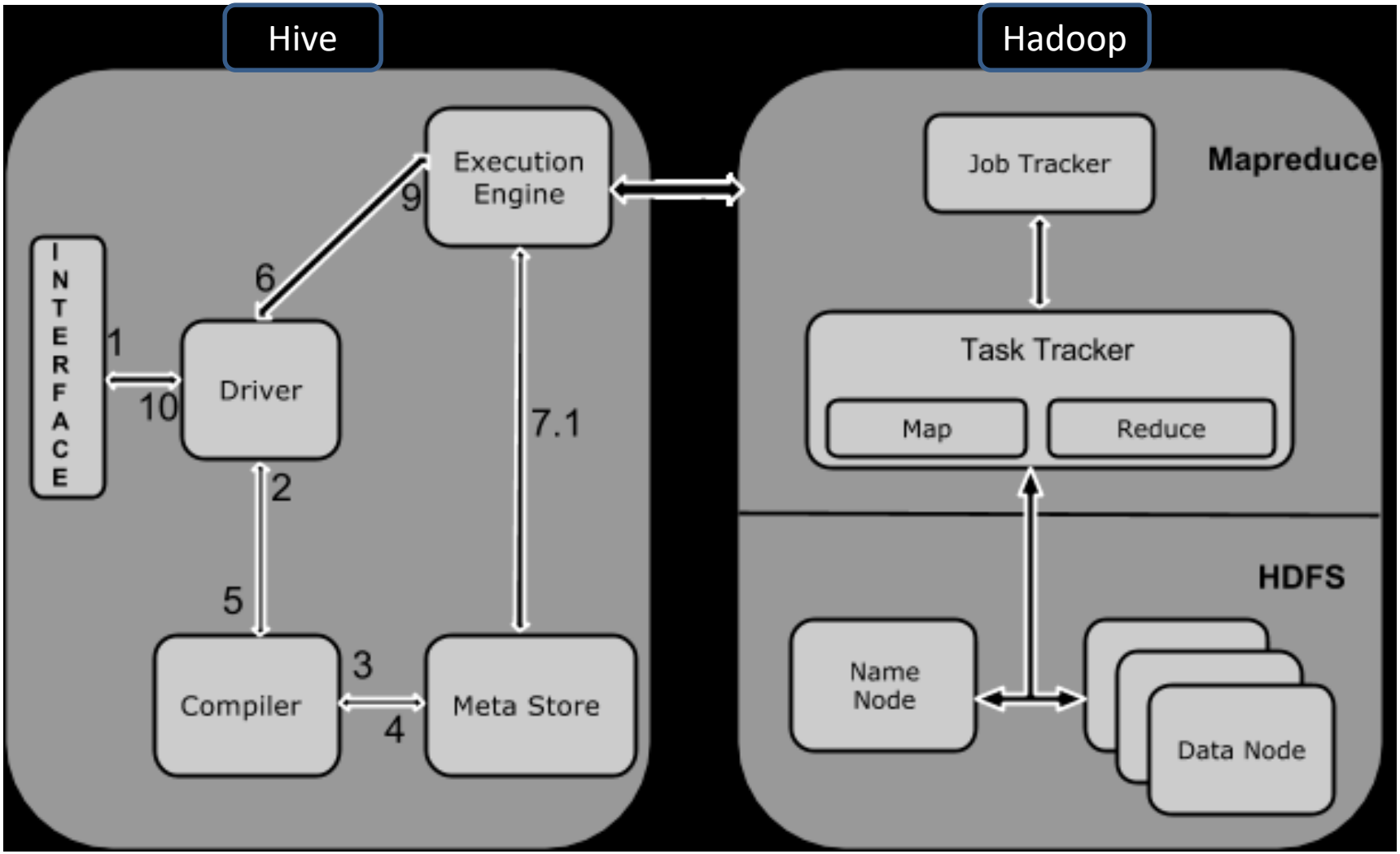
Hadoop distributed file system or HBASE are the data storage techniques to store data into file system.

# Cont...

**Execution Engine**

The conjunction part of HiveQL process Engine and MapReduce is Hive Execution Engine. Execution engine processes the query and generates results as same as MapReduce results. It uses the flavor of MapReduce.

# Working of Hive



# Interaction

**Execute Query**

- The Hive interface such as Command Line or Web UI sends query to Driver (any database driver such as JDBC, ODBC, etc.) to execute.

**Get Plan**

- The driver takes the help of query compiler that parses the query to check the syntax and query plan or the requirement of query.

**Get Metadata**

- The compiler sends metadata request to Metastore (any database).

# Cont…

**Send Metadata**

- Metastore sends metadata as a response to the compiler.

**Send Plan**

- The compiler checks the requirement and resends the plan to the driver. Up to here, the parsing and compiling of a query is complete.

**Execute Plan**

- The driver sends the execute plan to the execution engine.

**Execute Job**

- Internally, the process of execution job is a MapReduce job. The execution engine sends the job to JobTracker, which is in Name node and it assigns this job to TaskTracker, which is in Data node. Here, the query executes MapReduce job.

**Metadata Ops**

- Meanwhile in execution, the execution engine can execute metadata operations with Metastore.

**Fetch Result**

- The execution engine receives the results from Data nodes.

**Send Results**

- The execution engine sends those resultant values to the driver.

**Send Results**

- The driver sends the results to Hive Interfaces.

# HIVEQL DDL

HiveQL DDL statements are documented here, including:

- CREATE DATABASE/SCHEMA, TABLE, VIEW, FUNCTION, INDEX
- DROP DATABASE/SCHEMA, TABLE, VIEW, INDEX
- TRUNCATE TABLE
- ALTER DATABASE/SCHEMA, TABLE, VIEW
- SHOW DATABASES/SCHEMAS, TABLES, TBLPROPERTIES, VIEWS, PARTITIONS, FUNCTIONS, INDEX[ES], COLUMNS, CREATE TABLE
- DESCRIBE DATABASE/SCHEMA, table_name, view_name

# CREATING

**Create Database**

CREATE (DATABASE|SCHEMA) [IF NOT EXISTS] database_name [COMMENT database_comment] [LOCATION hdfs_path] [WITH DBPROPERTIES (property_name=property_value, ...)];

**Example:**

hive> CREATE SCHEMA userdb;

hive> CREATE TABLE IF NOT EXISTS employee ( eid int, name String, salary String, destination String)

COMMENT 'Employee details'

ROW FORMAT DELIMITED

FIELDS TERMINATED BY '\t'

LINES TERMINATED BY '\n'

STORED AS TEXTFILE;

# Cont…

**Use Database**

USE database_name;

**Drop Database**

DROP (DATABASE|SCHEMA) [IF EXISTS] database_name [RESTRICT|CASCADE];

**Example:**

hive> DROP DATABASE IF EXISTS userdb CASCADE;

# Cont...

**Alter Table**

ALTER TABLE name RENAME TO new_name

ALTER TABLE name ADD COLUMNS (col_spec[, col_spec ...])

ALTER TABLE name DROP [COLUMN] column_name

ALTER TABLE name CHANGE column_name new_name new_type

ALTER TABLE name REPLACE COLUMNS (col_spec[, col_spec ...])

**Rename To... Statement**

hive> ALTER TABLE employee RENAME TO emp;

**Change Statement**

hive> ALTER TABLE employee CHANGE name ename String; hive> ALTER TABLE employee CHANGE salary salary Double;