UNIVERSIDAD COMPLUTENSE DE MADRID

ntic master School

Apache Hive

Dr. Pablo J. Villacorta

Septiembre de 2020

Apache Hive

Apache Hive

Sistema de DataWarehouse para manejar (leer, escribir, ETL, reporting, análisis) datos almacenados de manera distribuida utilizando lenguaje SQL

- Desarrollada inicialmente por Facebook en 2007, proyecto Apache desde 2010.
- Hive propociona un **armazón SQL** para manejar datos existentes (**estructurados**)
  - Herramientas para acceder fácilmente a los datos usando SQL, para uso por analistas BI que no dominan la programación en Java (MapReduce o Spark) y para tareas como ETL, reporting y análisis de datos.
  - Mecanismo para imponer estructura a datos almacenados tanto en HDFS como en Apache HBase (base de datos no relacional distribuida)
  - Lenguaje de consulta: HiveQL, con soporte para funciones a nivel de fila definidas por el usuario (UDF), y soporte para tipos complejos
  - Utilización: OLAP (Online Analytical Processing), es decir, análisis en batch del informacional (histórico) de la empresa. **No es adecuado para OLTP**
- Motor de ejecución **intercambiable**: MapReduce, Apache Spark o Apache Tez
  - Traduce consultas SQL a jobs de MapReduce, de Spark o de Tez
- Formatos de archivo soportados: RCFile, Parquet, ORC, Avro, archivos comprimidos
- No comprueba el esquema (schema-on-read), como sí harían las BBDD relacionales (schema-on-write, comprobación en el momento de inserción de datos)

Apache Hive vs BBDD relacionales

- Con el paso del tiempo cada vez se asemejan más ya que se están eliminando las restricciones y añadiendo a Hive características propias de las BBDD tradicionales

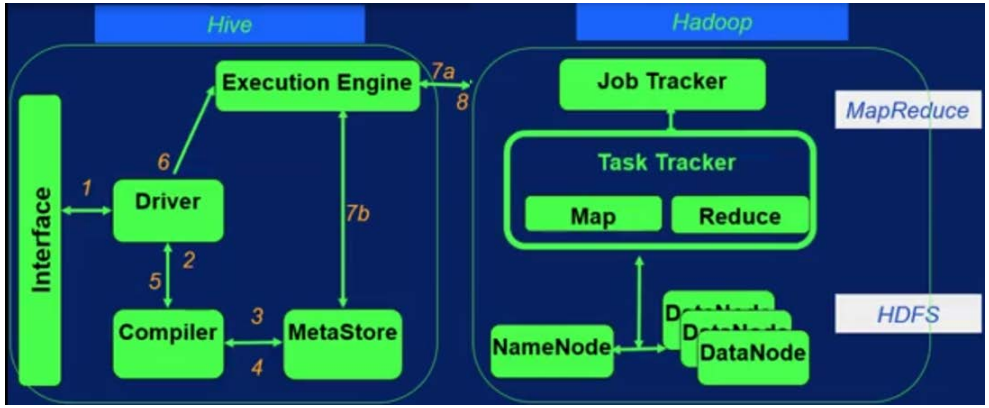
- No comprueba el esquema (solo en la lectura: schema-on-read), a diferencia de las BBDD relacionales (schema-on-write, comprobación en el momento de inserción de datos)

- Limitaciones propias de HDFS: no permite modificaciones. Archivos delta, mergeados periódicamente por procesos MapReduce ejecutados en segundo plano por el metastore (requiere transaccionalidad activada para la tabla que se está actualizando)

- Transaccionalidad añadida posteriormente, al igual que los índices.
  - Versiones recientes: transacciones con semántica ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad).
  - Sólo soportadas para ficheros ORC

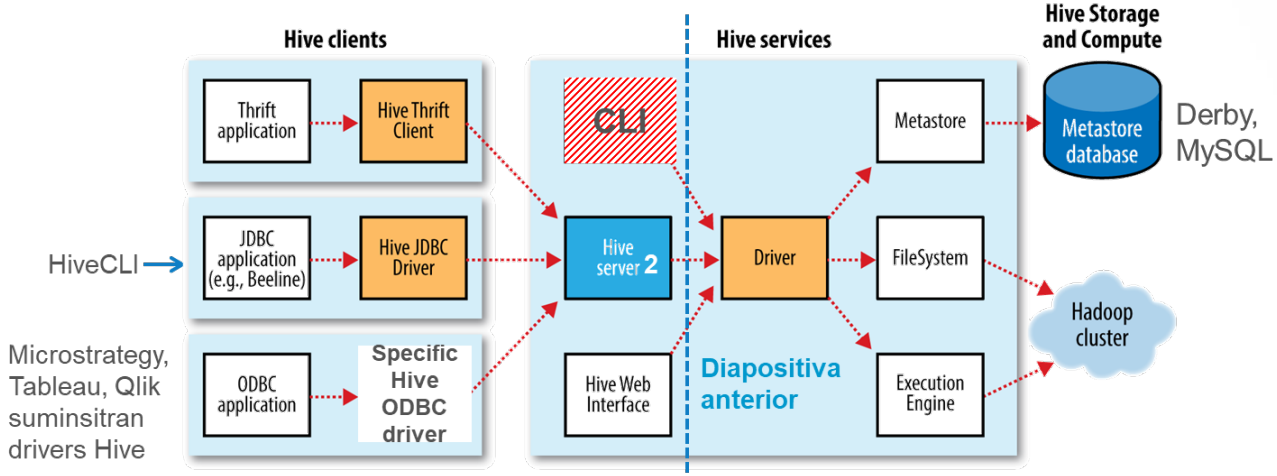
- Bloqueo a nivel de tabla y de partición gestionado por ZooKeeper.

Funcionamiento de Hive sobre MapReduce



- Driver: controlador que recibe sentencias HiveQL y coordina todo el ciclo de ejecución, recoge resultados de la fase Reduce y los envía de vuelta a la interfaz desde la que se está usando Hive
- Compilador: convierte la consulta a un Abstract Syntax Tree (AST), comprueba su validez con los datos del metastore, y pasa el AST a un DAG optimizado de MapReduce (stages y tasks)
- Metastore: servicio de Hive que recupera los metadatos físicamente almacenados en una BBDD relacional (por defecto Apache Derby, pero puede ser también MySQL)
- Motor de ejecución: envía el plan de trabajos al cluster de Hadoop, y también efectúa las operaciones necesarias con el metastore para actualizarlo si es necesario según la consulta SQL
- Interface: puede ser desde una CLI (Command Line Interface) hasta una aplicación remota que se conecta a Hive y envía sentencias SQL que el usuario ha escrito en esa aplicación

Funcionamiento de Hive




```
% bin/beeline
Hive version 0.11.0-SNAPSHOT by Apache
beeline> !connect jdbc:hive2://localhost:10000 scott tiger
!connect jdbc:hive2://localhost:10000 scott tiger
Connecting to jdbc:hive2://localhost:10000
Connected to: Hive (version 0.10.0)
Driver: Hive (version 0.10.0-SNAPSHOT)
Transaction isolation: TRANSACTION_REPEATABLE_READ

0: jdbc:hive2://localhost:10000> show tables;
...
```

```
...
show tables;
+-------------------+
|    tab_name       |
+-------------------+
| primitives        |
| src               |
| src1              |
| src_json          |
| src_sequencefile  |
| src_thrift        |
| srcbucket         |
+-------------------+
7 rows selected (1.079 seconds)
```

Unidades de información en Hive

- Tabla: físicamente es un directorio de HDFS que almacena los datos de una tabla
  - **Tabla manejada (managed)**: Hive es dueño de ella. El borrado de la tabla implica borrado físico de los datos
  - **Tabla externa (external):** Hive sólo genera metadatos para ficheros de datos ya existentes. El borrado de la tabla solo borra metadatos de Hive pero no los datos físicos
- Partición: un subdirectorio de HDFS dedicado al subconjunto de filas que tienen un cierto valor en una columna (la cual en realidad no forma parte de los metadatos)
- Bucket (compartimentos): grupos de filas de una tabla que no necesariamente comparten el valor de una columna, pero los valores de dicha columna han sido agrupados en unos pocos compartimentos (menos que los valores existentes).
  - Hace más eficientes las consultas, incluyendo los JOIN si los atributos de JOIN están buckerizados en ambas tablas.