upGrad

Apache Hive and Querying

Mayukh Chakraborty

Agenda

- Key Features & Use Cases
- HIVE Architecture
- Data Model
- Queries
- Advanced Features
- Questions & Answers

Apache HIVE: Key Features

- In spite of having a tremendous volume and being unstructured in nature, where traditional RDBMSs fail, Hive is capable of processing unstructured (or rather, semi-structured) data.

- A SQL-like interface to write queries on large datasets residing a distributed cluster of machines.

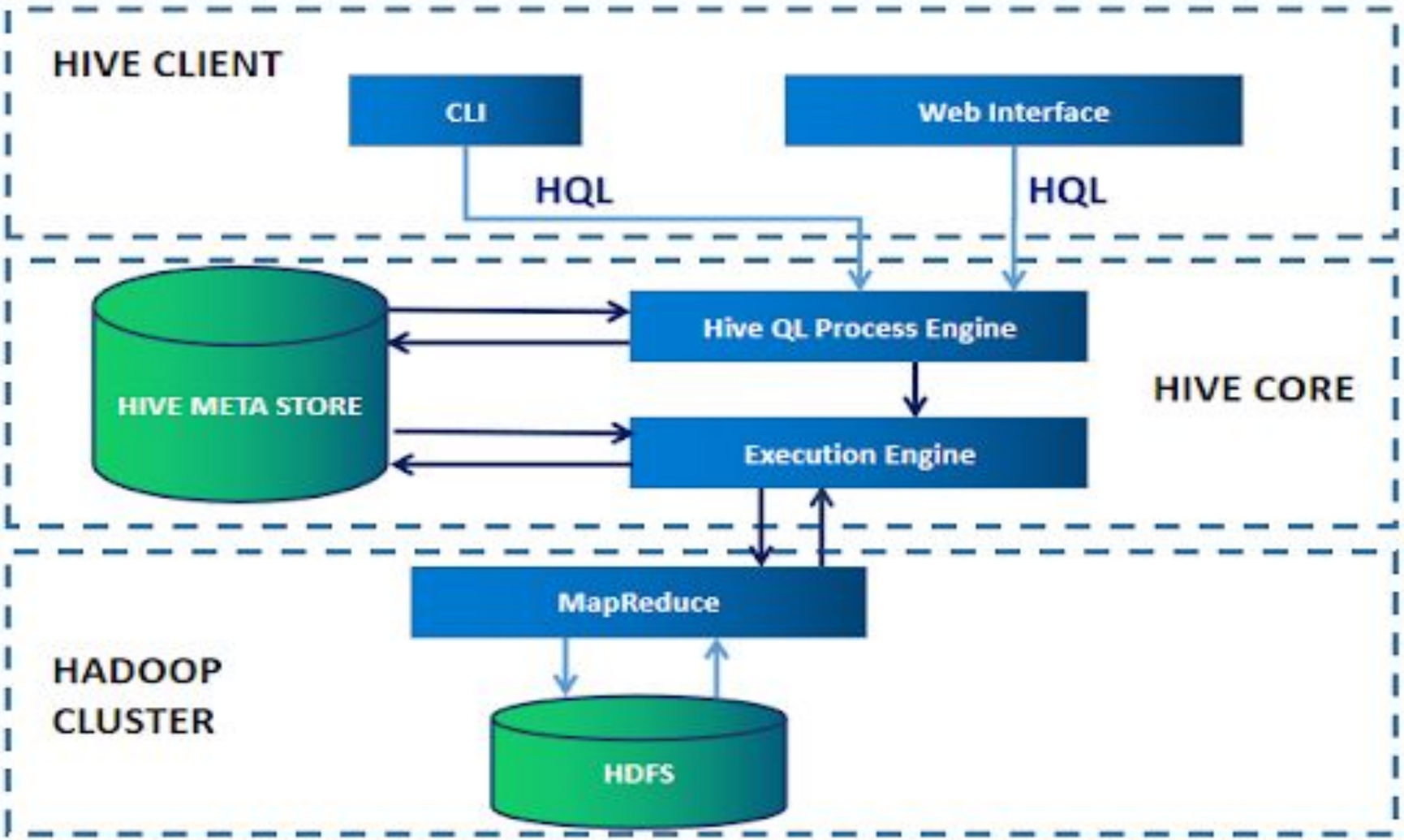
Key Features

- Hive can be used to process all variants of data i.e. Structured, Semi-structured and Unstructured.

- A variety of built-in functions for working with dates, strings along with support for custom user defined functions(UDF) etc.

- Easy Extraction, Transformation and Loading of data at large scale.

HIVE Architecture



HIVE Tables

- Internal Tables
  - Temporary Tables.
  - Life cycle Managed by HIVE.
- External Tables
  - Use the data outside of Hive as well.
  - Keep the data on HDFS even after deleting table in HIVE.

HIVE Table Creation: Internal Table

```
create table if not exists user_info (
        id int,
        age int,
        gender string,
        profession string
 )
row format delimited fields terminated by '|'
lines terminated by '\n'
stored as textfile;
```

HIVE Table Creation: External Table

```
create external table if not exists user_info (
        id int,
        age int,
        gender string,
        profession string
 )
row format delimited fields terminated by '|'
lines terminated by '\n'
stored as textfile;
```

Quiz

What is ACID property?

Does HIVE supports ACID property?

Commands to enable ACID property in HIVE

```
set hive.support.concurrency = true;

set hive.enforce.bucketing = true;

set hive.exec.dynamic.partition.mode = nonstrict;

set hive.compactor.initiator.on = true;

set hive.compactor.worker.threads = 1;

set hive.txn.manager =

org.apache.hadoop.hive.ql.lockmgr.DbTxnManager;
```

Commands to enable ACID property in HIVE

```
CREATE TABLE table_name(
col1 int,
col2 string,
col3 int)
CLUSTERED BY col1 INTO 4 BUCKETS
STORED AS orc
tblproperties('transactional' ='true');
```

Quiz

Is Update/Delete operation possible in HIVE?

Quiz

Why HIVE can't be used for OLTP?

HIVE Data Storage Formats

- ORC
  - Row columnar data format created by Hortonworks
  - Key statistics such as count, max, min, and sum of each column are cached
- Parquet
  - Row columnar data format created by Cloudera
  - Specialized in efficiently storing and processing nested data types
- Avro
  - Row-based data format
  - stored as JSON

Quiz

What is the use of UDF(User Defined Functions)?

UDF

1. Copy the jar file from local machine into EC2 machine:

```
$> scp -i RHEL.pem /.../JavaUDF/target/javaudf-1.0-SNAPSHOT.jar
ec2-user@<public-dns>:/home/ec2-user/
```

2. Place the jar file to HDFS

```
$> hdfs dfs -put /home/ec2-user/javaudf-1.0-SNAPSHOT.jar
/user/root/tmp/javaudf-1.0-SNAPSHOT.jar
```

3. Run “ADD jar” command

```
hive> ADD jar hdfs:///user/root/tmp/javaudf-1.0-SNAPSHOT.jar;
hive> list jars;
```

UDF

4. Create UDF

```
hive> CREATE FUNCTION get_honorifics AS
'org.example.javaudf.Honorifics';
```

5. Use UDF

```
hive> SELECT
 passenger_id,
 name,
 get_honorifics(gender),
 gender
FROM titanic
LIMIT 10;
```

Hands on Hive Query Exercise

Source: https://drive.google.com/file/d/1NABrPOZEDOwQdQ7vl7omoTcmDYAen0Xx/view?usp=sharing

Questions

1. Get the Average fare
2. Show every Survivor's passenger Id, name, gender, age, Ticket No
3. Gender wise Survivor count
4. Age group & Gender wise passenger count

   Age Group rule:

   Old: above 60

   Mid-Age: 35-60

   Young: 20-35

   Teenager: 13-19

   Child: below 13

Table Creation & Data Loading

```
create table if not exists titanic (
        passenger_id int, survived int, class int,
        name string, gender string, age decimal,
        ticket string, fare decimal
  )
 row format delimited fields terminated by '|'
 lines terminated by '\n' stored as textfile
 tblproperties("skip.header.line.count"="1");


load data local inpath './titanic.tsv' into table titanic;
```

Solutions

1. Get the Average fare

```
select avg(fare) from titanic;
```

2. Show every Survivor's passenger Id, name, gender, age, Ticket No

```
select passenger_id, name, gender, age,  ticket
from titanic where survived = 1;
```

3. Gender wise passenger count

```
select gender, count(*) from titanic group by
gender;
```

Solutions

4. Age group & Gender wise passenger count

```
select CASE
    WHEN age > 60 THEN 'Old'
    WHEN age > 35 AND age <= 60 THEN 'Mid-age'
    WHEN age > 20 AND age <= 35 THEN 'Young'
    WHEN age > 13 AND age <= 20 THEN 'Teenager'
    ELSE 'Child'
END as age_group, gender, count(*) from titanic
group by age_group, gender;
```

Hidden gem: HIVE MapJoin

```
select /* + MAPJOIN (department) */
      employee.emp_id,
      employee.emp_name,
      employee.designation,
      department.dep_id,
      department.dep_city
from employee
left join department on (employee.dept_id =
department.dep_id);
```

Partitions

```
create table if not exists part_user_info (
    id int,
    age int,
    gender string,
    ratings int
)
partitioned by (profession string)
row format delimited fields terminated by '|'
lines terminated by '\n';
```

Partitions

```
SET hive.exec.dynamic.partition=true;
set hive.exec.dynamic.partition.mode=nonstrict;

Insert into retail_clickstream_partition
partition(event_date)
select *, to_date(event_time) from retail_clickstream;
```

Bucketing

```
create table if not exists buck_user_info (
  id int,
  age int,
  profession string)
partitioned by (gender string)
clustered by (age)
into 7 buckets
row format delimited fields terminated by '|'
lines terminated by "\n"
stored as textfile;
```

Bucketing

```
set hive.enforce.bucketing=true;
SET hive.exec.dynamic.partition=true;
set hive.exec.dynamic.partition.mode=nonstrict;

Insert into retail_clickstream_bucket
select * from retail_clickstream;
```

Using SerDe library

```
create table IF NOT EXISTS retail_clickstream3
(event_time timestamp, event_type string, product_id string,
category_id string, category_code string, brand string,
price decimal(10,3), user_id bigint, user_session string)
ROW FORMAT SERDE
'org.apache.hadoop.hive.serde2.OpenCSVSerde' WITH
 SERDEPROPERTIES(
"separatorChar"=",",
"quoteChar"="\"",
"escapeChar"="\\")
STORED AS TEXTFILE
LOCATION '/user/root/tmp/ecom_data/';
```

Quiz

What is the use of rank & window functions in HIVE?

Questions & Answers