Tutorial Big Data
CLEI 2019
Ciudad de Panamá, Panamá

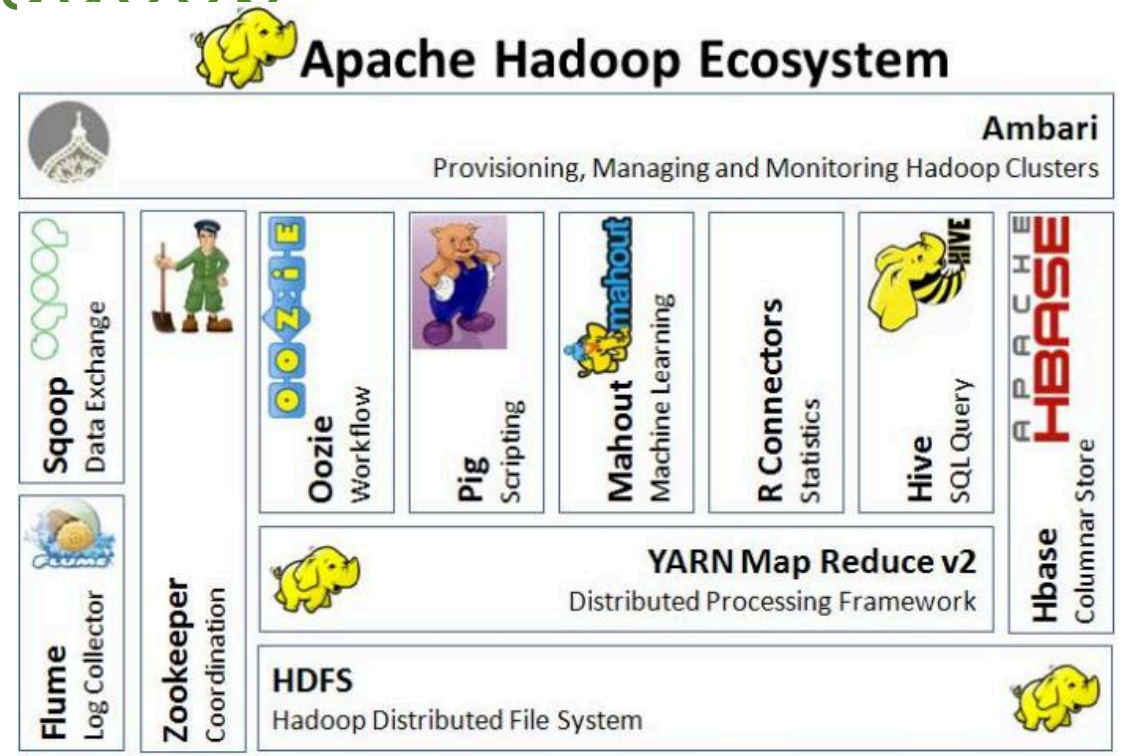
Por: Edwin Montoya
emontoya@eafit.edu.co

Hive y Sqoop

Edwin Montoya – emontoya@eafit.edu.co

2019

Hive en el Ecosistema de Hadoop



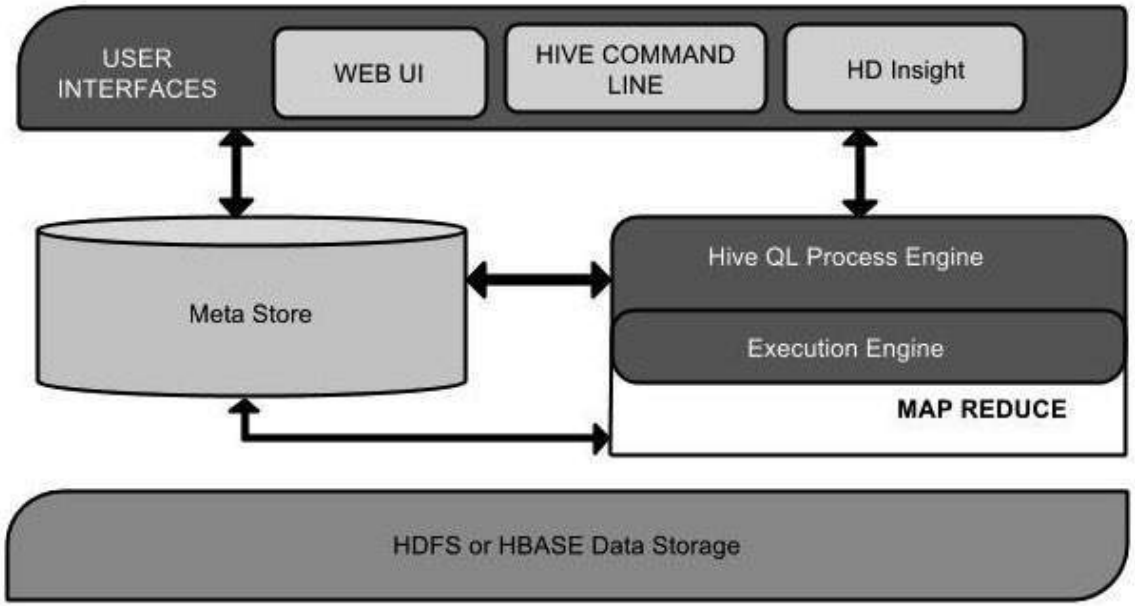
Hive

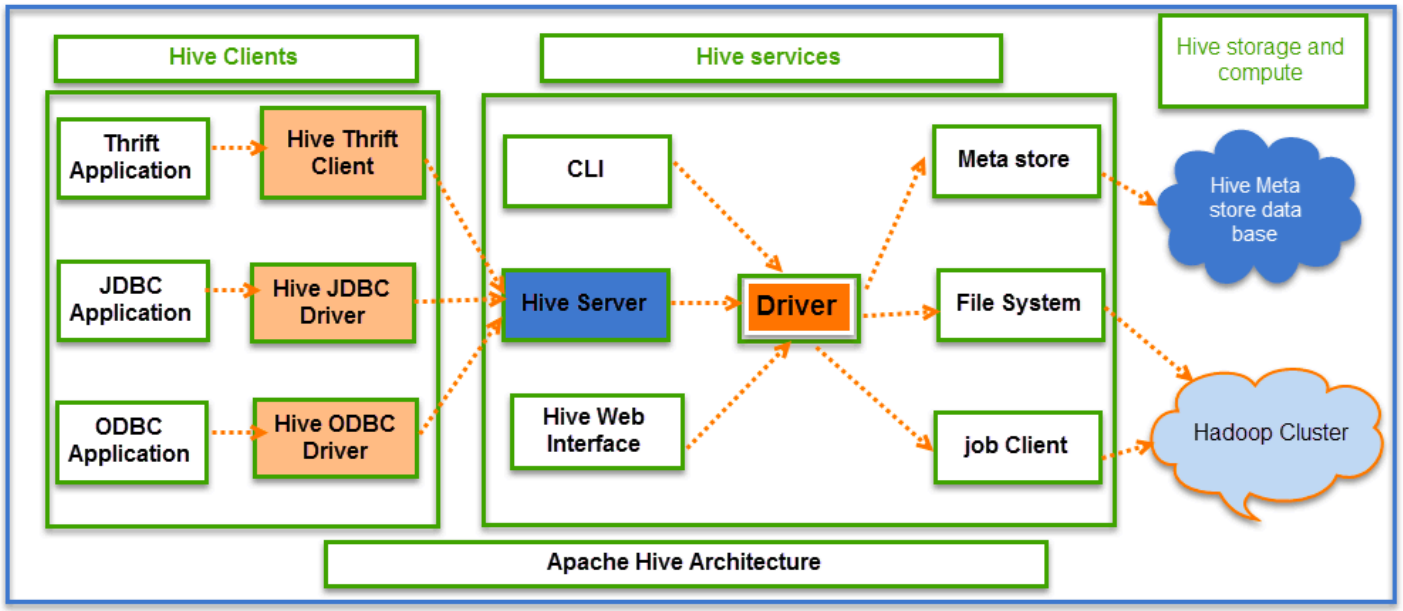
- **Que es HIVE:**
  - Hive provee un dialecto SQL, llamado *Hive Query Language* (abreviado *HiveQL* o solo *HQL*) para consultas en datos almacenados en un cluster Hadoop.
- **Hive es adecuado para aplicaciones de Data Warehouse.**
- **Hive NO DISEÑADO PARA:**
  - Aplicaciones transaccionales.
  - No provee actualización, inserción ni borrado a nivel de registro.

Hive

- **Hive Diseñado para carga masiva de datos, o mapeo de datos HDFS con vista SQL, y provisión de Consultas y Procesamiento.**

ARQUITECTURA HIVE




Hive Clients
Hive services
Hive storage and compute
Thrift Application
Hive Thrift Client
JDBC Application
Hive JDBC Driver
ODBC Application
Hive ODBC Driver
CLI
Hive Server
Hive Web Interface
Driver
Meta store
File System
job Client
Hive Meta store data base
Hadoop Cluster
Apache Hive Architecture

Modelo de Datos

- Tables
  - Typed columns (int, float, string, boolean)
  - Also, list: map (for JSON-like data)
- Partitions
  - For example, range-partition tables by date
- Buckets
  - Hash partitions within ranges (useful for sampling, join optimization)

Source: cc-licensed slide by Cloudera

Metastore

- Database: namespace conteniendo un conjunto de tablas
- Mantiene la definición de las tablas (column types, physical layout)
- Mantiene información de particiones
- Puede ser almacenado en Derby, MySQL, u otras bases de datos

Modelo fisico

- Directorio Warehouse en HDFS
  - E.g., /user/hive/warehouse
- Tablas almacenadas en subdirectorios del warehoue
  - Particiones forman subdirectorios de tablas
- Los datos reales son almacenados en archivos planos SIN ningun esquema
  - Control char-delimited text, or SequenceFiles
  - With custom SerDe, can use arbitrary format

Laboratorios

- https://github.com/edwinmontoya/tutorialbigdataclei2019.git
- **Se tendrá acceso a una serie de Recursos en Nube:**
  - **AWS EMR:**
    - http://emr1.emontoya.ml:8888 (hue)
    - http://emr2.emontoya.ml:8888 (si necesita)
    - http://emr3.emontoya.ml:8888 (si necesita)
      - User: admin Password: Clei2019*

    - http://emr1.emontoya.ml:8890 (zeppelin)
    - http://emr2.emontoya.ml:8890 (si necesita)
    - http://emr3.emontoya.ml:8890 (si necesita)
      - Sin autenticación

Crear la tabla HDI en Hive:

```
create database  cecdb;
use mydb;

CREATE TABLE HDI (id INT, country STRING, hdi FLOAT, lifeex INT,
mysch INT, eysch INT, gni INT) ROW FORMAT DELIMITED FIELDS
TERMINATED BY ',' STORED AS TEXTFILE LOCATION
‘/user/<username>/datasets/onu/ hdi-data.csv’;

show tables;
describe hdi;
```

Consultas en Hive

- hacer consultas y cálculos sobre la tabla HDI:

```
USE mydb;
SELECT * from hdi;
SELECT country, gni from hdi where gni > 2000;
```

WORDCOUNT EN HIVE:

```
// crear table
USE mydb;
CREATE EXTERNAL TABLE docs (line STRING)
        STORED AS TEXTFILE
        LOCATION ‘s3://emontoyapublic/datasets/gutenberg-small/’;
```

Opción 2:

```
CREATE EXTERNAL TABLE docs (line STRING)
        STORED AS TEXTFILE
        LOCATION ‘/user/<username>/datasets/gutenberg-small/';
```

WORDCOUNT EN HIVE:

```
// ordenado por palabra
SELECT word, count(1) AS count FROM (SELECT explode(split(line,' ')) AS word FROM docs) w GROUP BY word ORDER BY word DESC LIMIT 10;
// ordenado por frecuencia de menor a mayor
SELECT word, count(1) AS count FROM (SELECT explode(split(line,' ')) AS word FROM docs) w GROUP BY word ORDER BY count DESC LIMIT 10;
```

RETO:

¿Cómo llenar una tabla con los resultados de un Query? por ejemplo, como almacenar en una tabla el diccionario de frecuencia de palabras en el wordcount?

```
use mydb;
create table wordcount (word STRING, cont INT);

INSERT INTO wordcount
        (SELECT word, count(1) AS count
        FROM (SELECT explode(split(line,' ')) AS word
                FROM docs) w
        GROUP BY word
        );

select * from wordcount order by cont desc limit 10;
```

Apache Sqoop

Gestión en Mysql

Datos en Mysql

En database-1.cj1yhistqein.us-east-2.rds.amazonaws.com**, se tiene Mysql con:**

Base de datos: "cursodb"

Tabla: "employee"

User: curso / curso

$ mysql –u curso –h database-1.cj1yhistqein.us-east-2.rds.amazonaws.com **–p**

Enter password: ****

mysql> use cursodb;

Creando la Base de datos y Tabla

$ mysql -u root –h database-1.cj1yhistqein.us-east-2.rds.amazonaws.com -p

Enter password: *****

mysql> create database cursodb;

mysql> use cursodb;

mysql> CREATE TABLE employee (emp_id INT NOT NULL, name VARCHAR(45), salary INT, PRIMARY KEY (emp_id));

```
mysql> describe employee;
+--------+-------------+------+-----+---------+-------+
| Field  | Type        | Null | Key | Default | Extra |
+--------+-------------+------+-----+---------+-------+
| emp_id | int(11)     | NO   | PRI | NULL    |       |
| name   | varchar(45) | YES  |     | NULL    |       |
| salary | int(11)     | YES  |     | NULL    |       |
+--------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)

mysql>
```

Creando usuario

```
$ mysql -u root –h database-1.cj1yhistqein.us-east-2.rds.amazonaws.com -p
Enter password: *****
mysql> CREATE USER 'curso'@'%' IDENTIFIED BY 'curso';
mysql> GRANT ALL PRIVILEGES ON cursodb.* TO 'curso'@'%';
```

Datos de prueba

```
$ mysql -u curso –h database-1.cj1yhistqein.us-east-2.rds.amazonaws.com -p
Enter password: curso
mysql> use cursodb;
mysql> insert into employee values (101, 'name1', 1800);
mysql> insert into employee values (102, 'name2', 1500);
mysql> insert into employee values (103, 'name3', 1000);
mysql> insert into employee values (104, 'name4', 2000);
mysql> insert into employee values (105, 'name5', 1600);
Query OK, 1 row affected (0.00 sec)
mysql>
```

Consulta

```
mysql> select * from employee;
+--------+-------+--------+
| emp_id | name  | salary |
+--------+-------+--------+
|    101 | name1 |   1800 |
|    102 | name2 |   1500 |
|    103 | name3 |   1000 |
|    104 | name4 |   2000
|
|    105 | name5 |   1600 |
+--------+-------+--------+
5 rows in set (0.00 sec)
mysql>
```

Apache sqoop

- Todos los camandos sqoop

```
$ sqoop help
Available commands:
  codegen            Generate code to interact with database records
  create-hive-table  Import a table definition into Hive
  eval               Evaluate a SQL statement and display the results
  export             Export an HDFS directory to a database table
  help               List available commands
  import             Import a table from a database to HDFS
  import-all-tables  Import tables from a database to HDFS
  import-mainframe   Import datasets from a mainframe server to HDFS
  job                Work with saved jobs
  list-databases     List available databases on a server
  list-tables        List available tables in a database
  merge              Merge results of incremental imports
  metastore          Run a standalone Sqoop metastore
  version            Display version information
```

Apache sqoop

- **COMANDOS PRINCIPALES:**
  - **import** -> $ sqoop import …
  - **export** -> $ sqoop export …
  - **create-hive-table** -> $ sqoop create-hive-table …
  - **hive-import** -> $ sqoop import … --hive-import …
  - **eval** -> $ sqoop eval … --query "…"

Sqoop – import 2 hdfs

- Transferir datos de una base de datos (tipo mysql) hacia HDFS:

```
$ sqoop import --connect jdbc:mysql://database-1.cj1yhistqein.us-east-2.rds.amazonaws.com:3306/cursodb --username curso -P --table employee --target-dir /user/<username>/employee -m 1 --mysql-delimiters
```

sqoop import

```
$ hdfs dfs -ls /user/ <username>/employee
Found 2 items
-rw-r--r--   1 emontoya cluster-users          0 2017-04-16 23:38
/user/<username>/employee/_SUCCESS
-rw-r--r--   1 emontoya cluster-users         75 2017-04-16 23:38
/user/<username>/employee/part-m-00000
$
```

sqoop create-hive-table

```
$ sqoop create-hive-table --connect jdbc:mysql://database-1.cj1yhistqein.us-east-2.rds.amazonaws.com:3306/cursodb --username curso -P --table employee --hive-table employee –hive-database mydb -m 1 --mysql-delimiters
$
```

Sqoop – import 2 hive

- Transferir datos de una base de datos (tipo mysql) hacia HIVE vía HDFS:

```
$ sqoop import --connect jdbc:mysql://database-
1.cj1yhistqein.us-east-2.rds.amazonaws.com:3306/cursodb --
username curso -P --table employee --hive-database mydb    --
hive-import --hive-table employee -m 1 --mysql-delimiters
```

Sqoop – import all tables

- **Transferir TODAS LAS TABLAS desde una base de datos (tipo mysql) hacia HDFS:**

```
$ sqoop import-all-tables --connect jdbc:mysql://database-
1.cj1yhistqein.us-east-2.rds.amazonaws.com:3306/cursodb --
username=curso --password=curso --warehouse-dir
/user/<username>/cursodb -m 1 --mysql-delimiters
```

Transferir TODAS LAS TABLAS desde una base de datos (tipo mysql) hacia HIVE:

```
$ sqoop import-all-tables --connect jdbc:mysql://database-
1.cj1yhistqein.us-east-2.rds.amazonaws.com:3306/ cursodb --
username=curso --password=curso --warehouse-dir
/user/<username>/cursodb –hive-database <username> --hive-import
-m 1 --mysql-delimiters
```

Sqoop – import all tables

- **Transferir TODAS LAS TABLAS desde una base de datos (tipo mysql) hacia HIVE vía HDFS:**

```
$ sudo –u hive sqoop import-all-tables -m 1 --connect
jdbc:mysql://database-1.cj1yhistqein.us-east-
2.rds.amazonaws.com:3306/retail_db --username=retail_dba --
password=caoba --mysql-delimiters --hive-database <mydb> --
create-hive-table --warehouse-dir=/tmp/ <mydb>/ --hive-import
```

```
$ sudo –u hive sqoop import-all-tables --connect
jdbc:mysql://database-1.cj1yhistqein.us-east-
2.rds.amazonaws.com:3306/retail_db --username=retail_dba --
password=caoba --hive-database <mydb> --hive-overwrite --
warehouse-dir=/tmp/stagemysql/ --hive-import -m 1 --mysql-
delimiters
```

Creación manual de tablas y carga (2)

```
describe employee;
OK
emp_id                  int

name                    string

salary                  int

Time taken: 0.219 seconds, Fetched: 3 row(s)
```

Sqoop export 2 mysql

- Crear una Tabla 'employee2' en Mysql con los mismos atributos de 'employee'

```
mysql> USE cursodb;

mysql> CREATE TABLE employee2 (  emp_id INT NOT NULL,  name VARCHAR(45),  salary INT,  PRIMARY KEY (emp_id));
```

Sqoop export 2 mysql

- Asumiendo datos separados por “,” en HDFS:

/user/<username>/mysql_in/*

```
$ sqoop export \
        --connect jdbc:mysql://127.0.0.1:3306/cursodb \
        --username curso -P \
        --table employee2 \
        --export-dir /user/<username>/mysqlOut
```

- En Mysql:

```
mysql> use cursodb;
mysql> select * from employee2;
…
mysql>
```