MBIT School
Madrid Business Intelligence Technology

Ecosistema Hadoop
Herramientas principales

Diego J. Bodas Sagi

¡Hola! Soy Diego J. Bodas Sagi

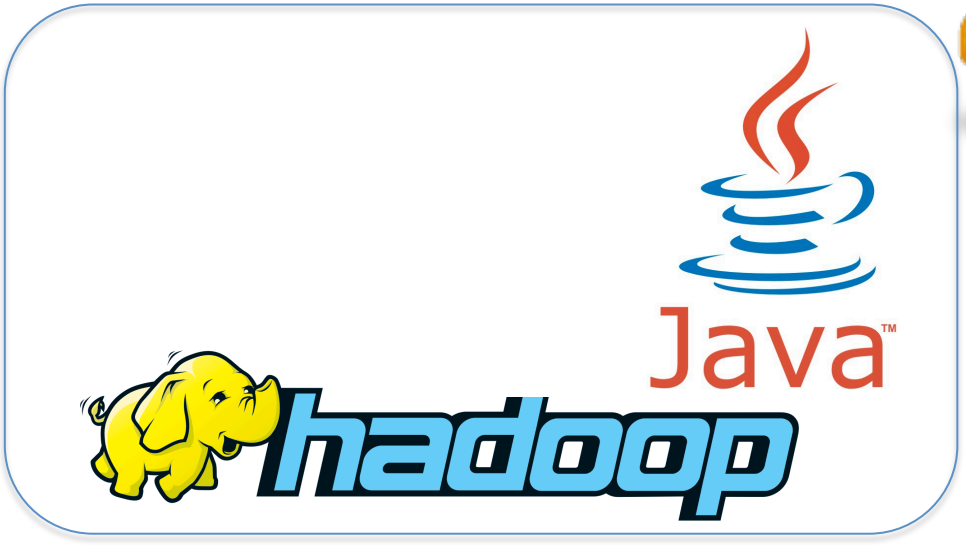
- BBVA Data & Analytics
- Experto en ingeniería de software, dirección de proyectos y análisis de datos
- Experiencia destacada en proyectos tanto de desarrollo como de investigación
- Más de 13 años de experiencia docente (UCM, UC3M, UFV)
- Doctor en Informática, licenciado CC. Matemáticas, Master in Executive Project Management (UV), M.B.A. (UNED), PMP.
- diegobodas@yahoo.es

Índice de la introducción

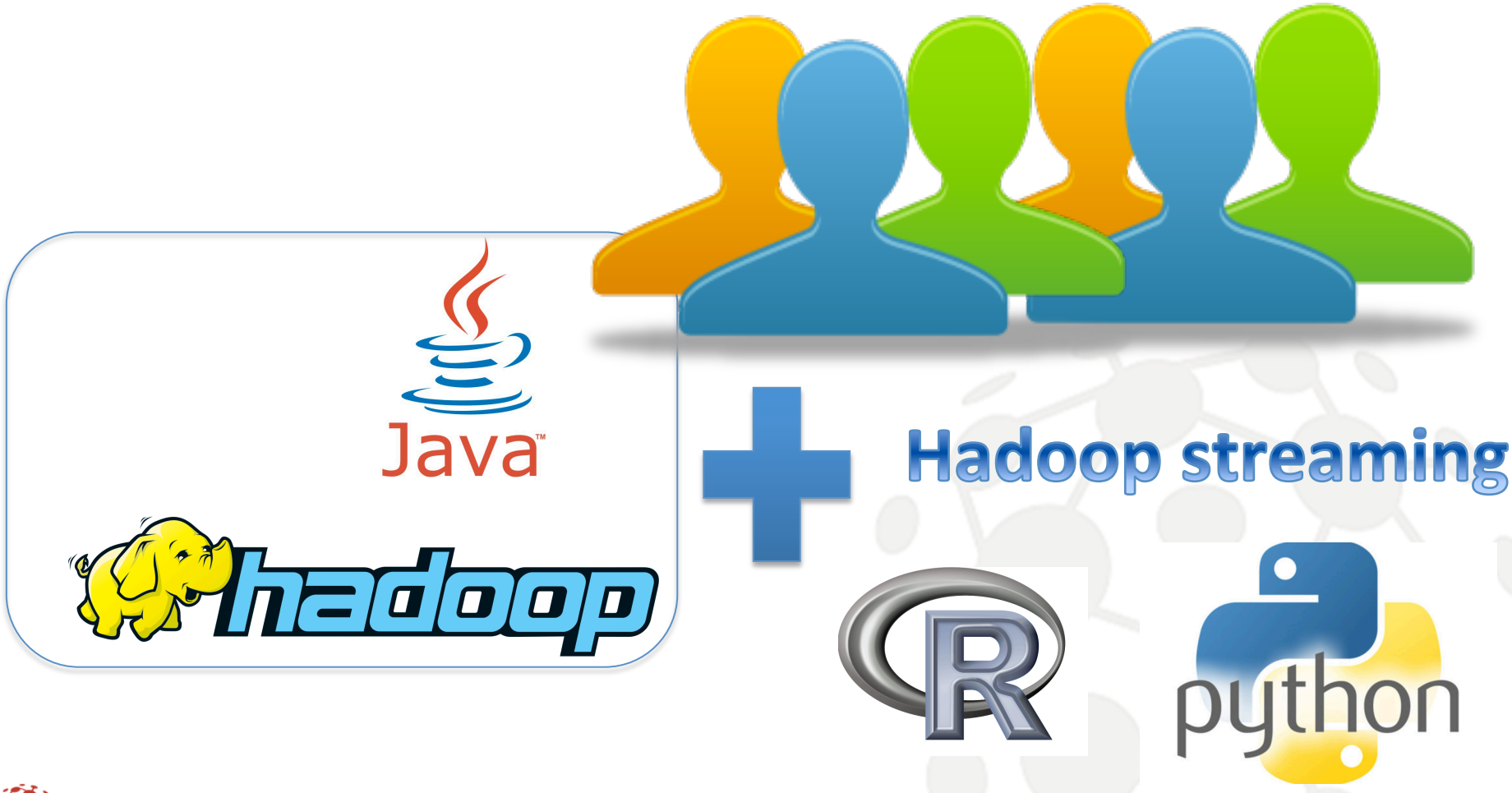
MBIT School
Madrid Business Intelligence Technology

- Contexto
- Sqoop
- Flume
- Hive
- Impala
- Pig
- Resumen

Situación inicial



Difusión

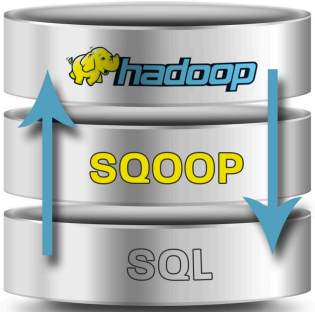


Sigue faltando gente del mundo BI

MBIT School
Madrid Business Intelligence Technology

- SQL????
- Simple reporting???
- User Interface???
- SQL Analytics???
- Getting relational data???

HUE



HIVE

cloudera
IMPALA

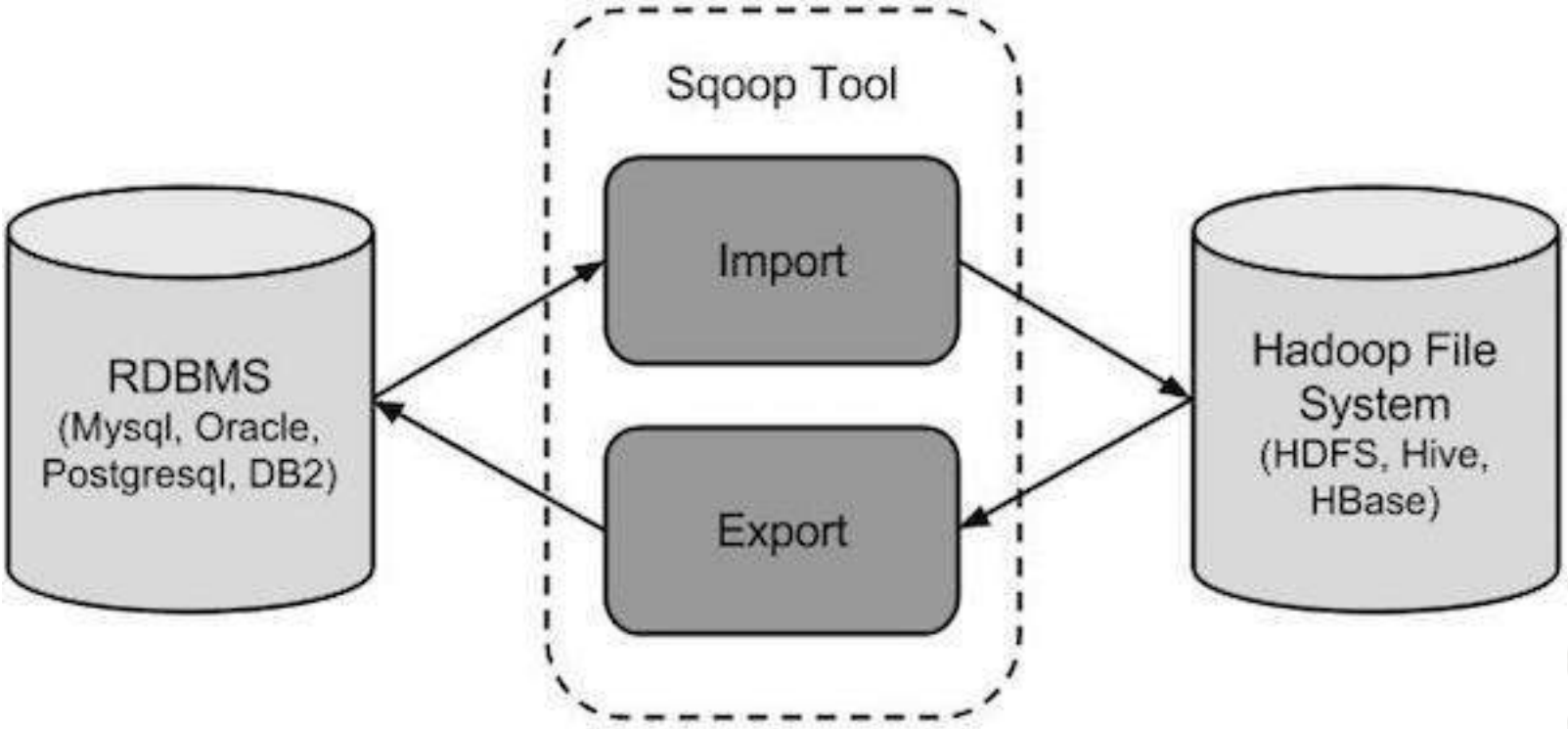
La película completa

Y un resumen para tener todo claro:

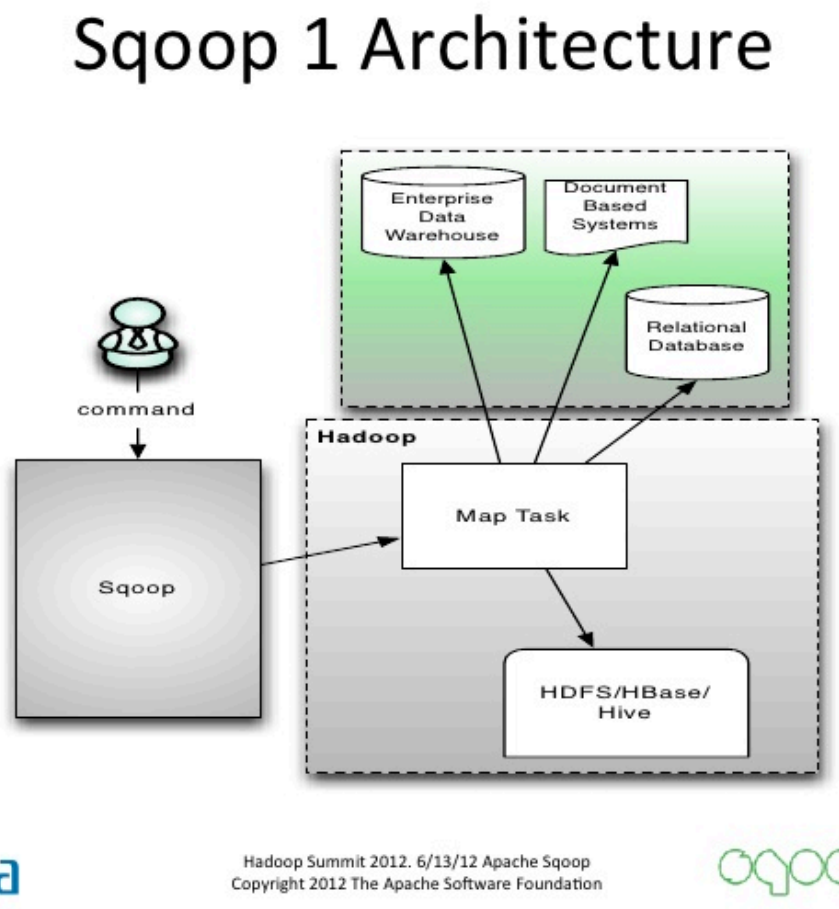
MBIT School
Madrid Business Intelligence Technology

- Si lo que quiero es…:
  1. Migrar datos de una BBDD relacional a un cluster de hadoop (o viceversa): Sqoop (procesamiento batch)
  2. Si lo que quiero es recolectar streaming data (ficheros de log i.e.): Flume
  3. Peticiones que pueden ser expresadas en forma de SQL: Hive
  4. Si queremos consultar fuentes de información HDFS pero con requisitos de latencia y rendimiento mayor: Impala (+rápido pero –simple que Hive)

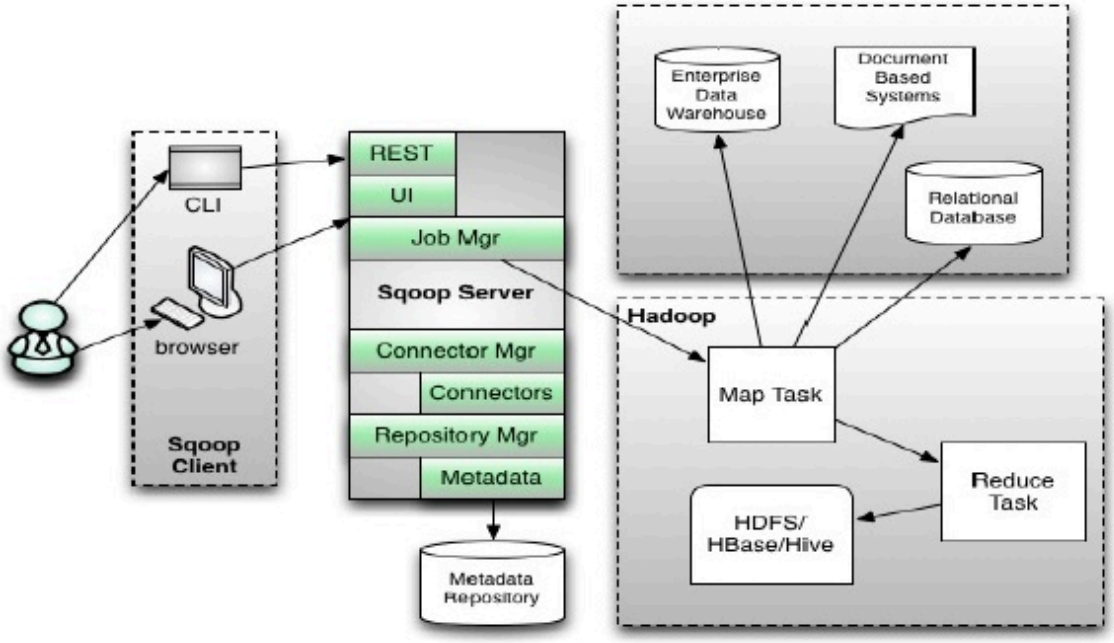
Let's start… Sqoop



Sqoop architectura



Sqoop 2 Architecture



Pasos detallados

- Examinar la tabla y los datos a llevar a HDFS: claves, número registros, número campos…
- Divide los registros a importar en varios mappers (4 por defecto) +mapper +paralelismo +velocidad
- Vuelca la información en formato csv, avro o SequenceFiles
- Además, la información se puede comprimir por el camino
- Posibilidad de añadir solo los registros nuevos (no importados previamente) (incremental append)
- Home HDFS y un directorio por tabla

Probemos Sqoop

- $ sqoop-version (warning Accumulo database conector, put the jar file in /var/lib/sqoop and set vars if needed).
- $ sqoop help
- $ sqoop help *command*
- Qué tablas tenemos
  - *mysql –u root –p* (password cloudera)
  - *show databases;*
  - *use retail_db;*
  - *show tables;*
- Import database
  - sqoop import --connect jdbc:mysql://localhost/retail_db –username root –password cloudera –table customers

¿Y …?

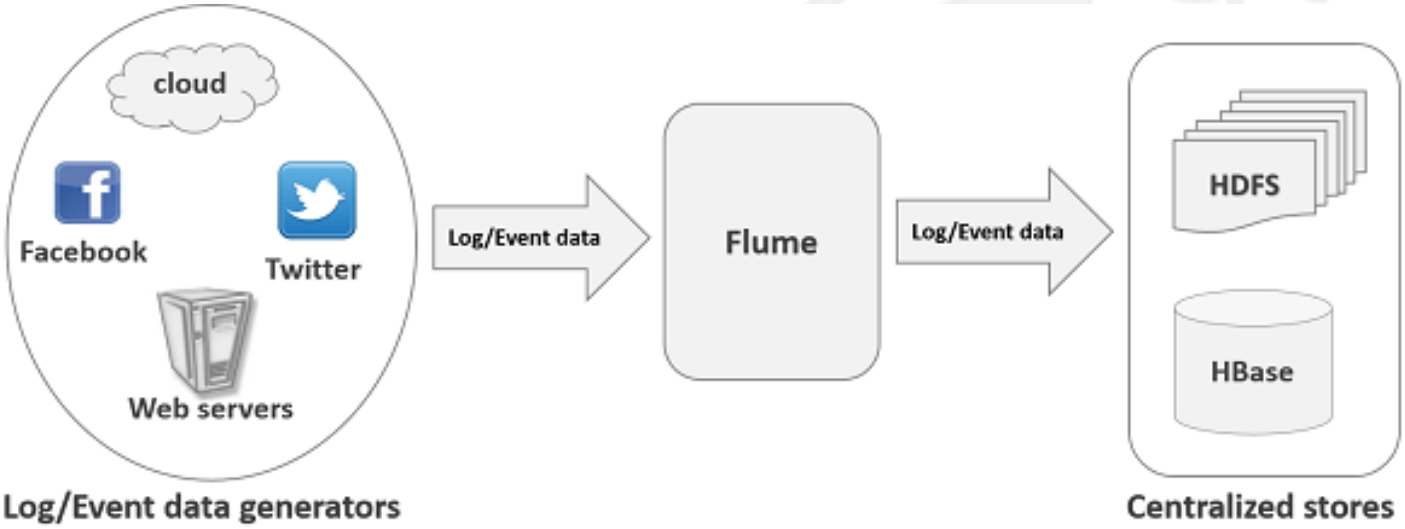
- ¿Funciona?
- ¿Está desactivado “Check HDFS permissions”?
- ¿Cómo están los servicios en el cloudera manager?
- Observar los que aparece en el home de cloudera (customers)

Ejercicio: export

- Objetivo: buscar documentación, probar, interactuar
- Pasos:
  1. Crea en mysql una tabla llamada custormers2 con la misma estructura que customers
  2. Exportar los datos en customers a la tabla customers2
  3. Comprobar que todo está en su sitio

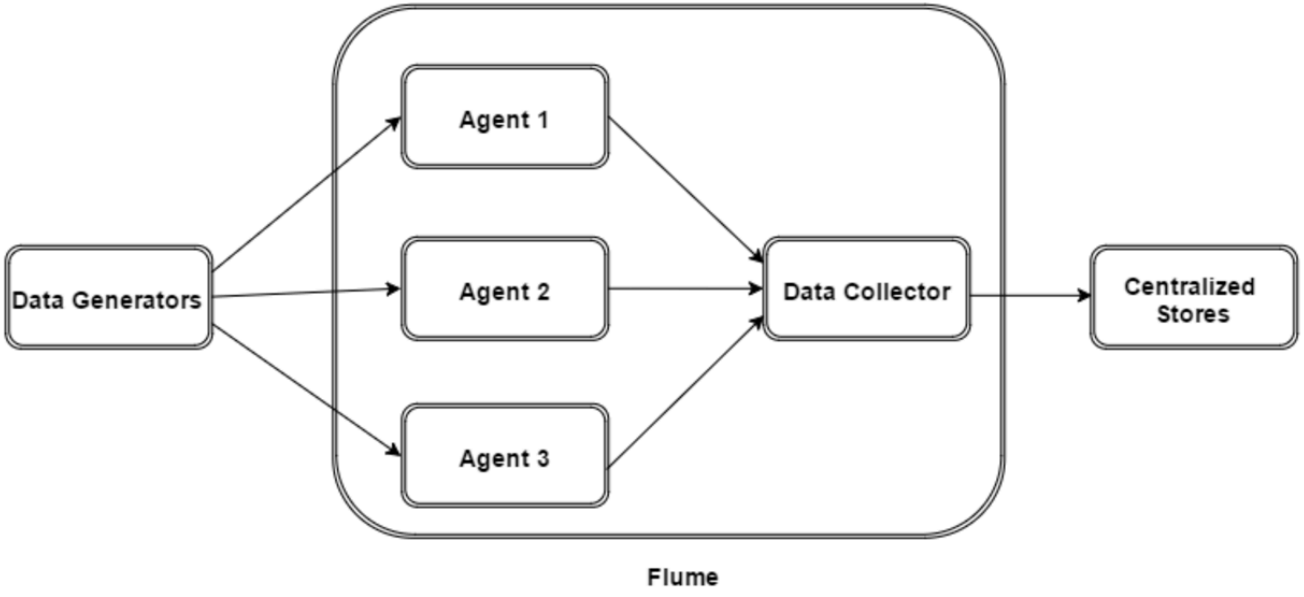
Flume

- Apache Flume is a tool/service/data ingestion mechanism for collecting aggregating and transporting large amounts of streaming data such as log files, events (etc...) from various sources to a centralized data store.
- Flume is a highly reliable, distributed, and configurable tool. It is principally designed to copy streaming data (log data) from various web servers to HDFS.



Flume: arquitectura

- Data generators generate data which gets collected by individual Flume agents running on them. A data collector (which is also an agent) collects the data from the agents which is aggregated and pushed into a centralized store such as HDFS or HBase

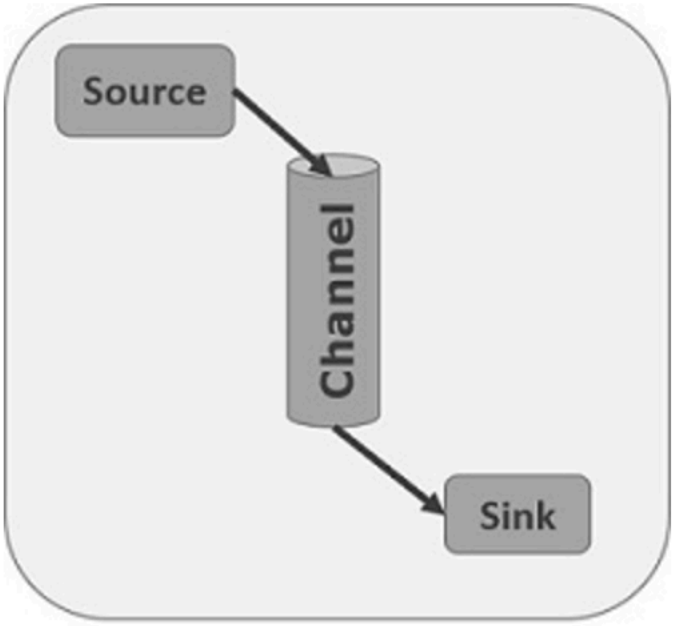


Flume: eventos

- An event is the basic unit of the data transported inside Flume. It contains a payload of byte array that is to be transported from the source to the destination accompanied by optional headers

Flume Agent

- An agent is an independent daemon process (JVM) in Flume. It receives the data (events) from clients or other agents and forwards it to its next destination (sink or agent). Flume may have more than one agent



Flume Agent

Flume: conceptos

MBIT School
Madrid Business Intelligence Technology

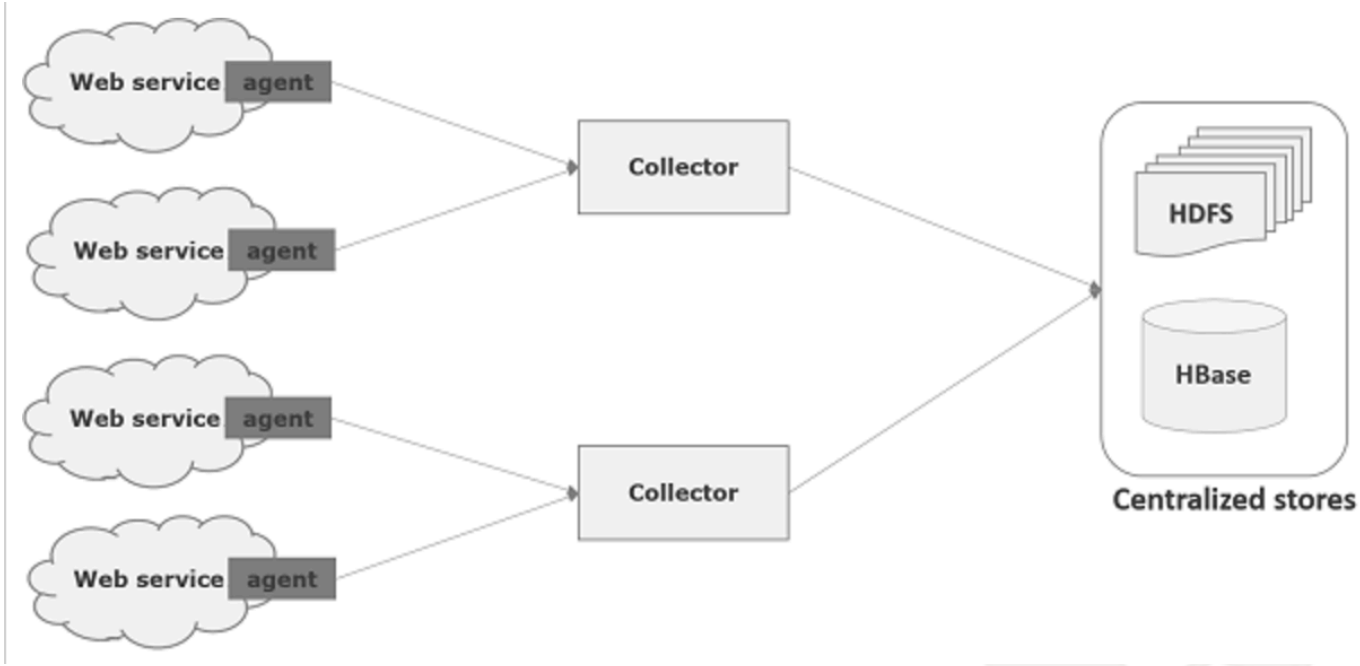
- A **source** is the component of an Agent which receives data from the data generators and transfers it to one or more channels in the form of Flume event
- A **channel** is a transient store which receives the events from the source and buffers them till they are consumed by sinks. It acts as a bridge between the sources and the sinks. These channels are fully transactional and they can work with any number of sources and sinks
- A **sink** stores the data into centralized stores like HBase and HDFS. It consumes the data (events) from the channels and delivers it to the destination. The destination of the sink might be another agent or the central stores

Otros conceptos

- **Interceptors** are used to alter/inspect flume events which are transferred between source and channel
- **Channel selectors** are used to determine which channel is to be opted to transfer the data in case of multiple channels
  - Default channel selectors or replicating channel selectors they replicates all the events in each channel
  - Multiplexing channel selectors: these decides the channel to send an event based on the address in the header of that event
- Sink processors are used to invoke a particular sink from the selected group of sinks. These are used to create failover paths for your sinks or load balance events across multiple sinks from a channel

Flume: Data Flow

- Multiple agents: multi-hop flow
- From one source to multiple channels: fan-out-flow
- From many sources to one channel: fan-in-flow



Flume: failure hadnly

- For each event, two transactions take place: one at the sender and one at the receiver
  - The sender sends events to the receiver. Soon after receiving the data, the receiver commits its own transaction and sends a “received” signal to the sender.
  - After receiving the signal, the sender commits its transaction. (Sender will not commit its transaction till it receives a signal from the receiver)

Flume: practice

- Install telnet: $sudo yum install telnet
- Abrir Cloudera Manager
  - Add a service: Flume
  - En el localhost
  - Start Flume
- Launch telnet para probar la implementación por defecto de Flume:
  - Lanzar telnet: telnet localhost 9999
- Teclear: ‘hola’ o cualquier otra cosa…
- Abrir: /var/log/flume-ng/flume-cmf-flume-AGENT-quickstart.cloudera.log

Flume: práctica

MBIT School
Madrid Business Intelligence Technology

- En Cloudera Manager arranca Hue y sigue las instrucciones siguientes:
  - Entra en Hue y accede a Hue Web UI
  - Desde ahí, en la parte superior derecha encontramos el File brower
  - En el home de cloudera, crear un directorio que se llame flume/events
  - Marca flume y en acciones, seleccionar permisos
  - Habilitar los permisos de escritura para todos

Flume: práctica

MBIT School
Madrid Business Intelligence Technology

- Ahora vamos a cambiar la configuración de flume
  - En el cloudera manager acceder y flume y dentro, a la configuración de flume
  - En 'Configuration file' ***añadir*** lo siguiente:

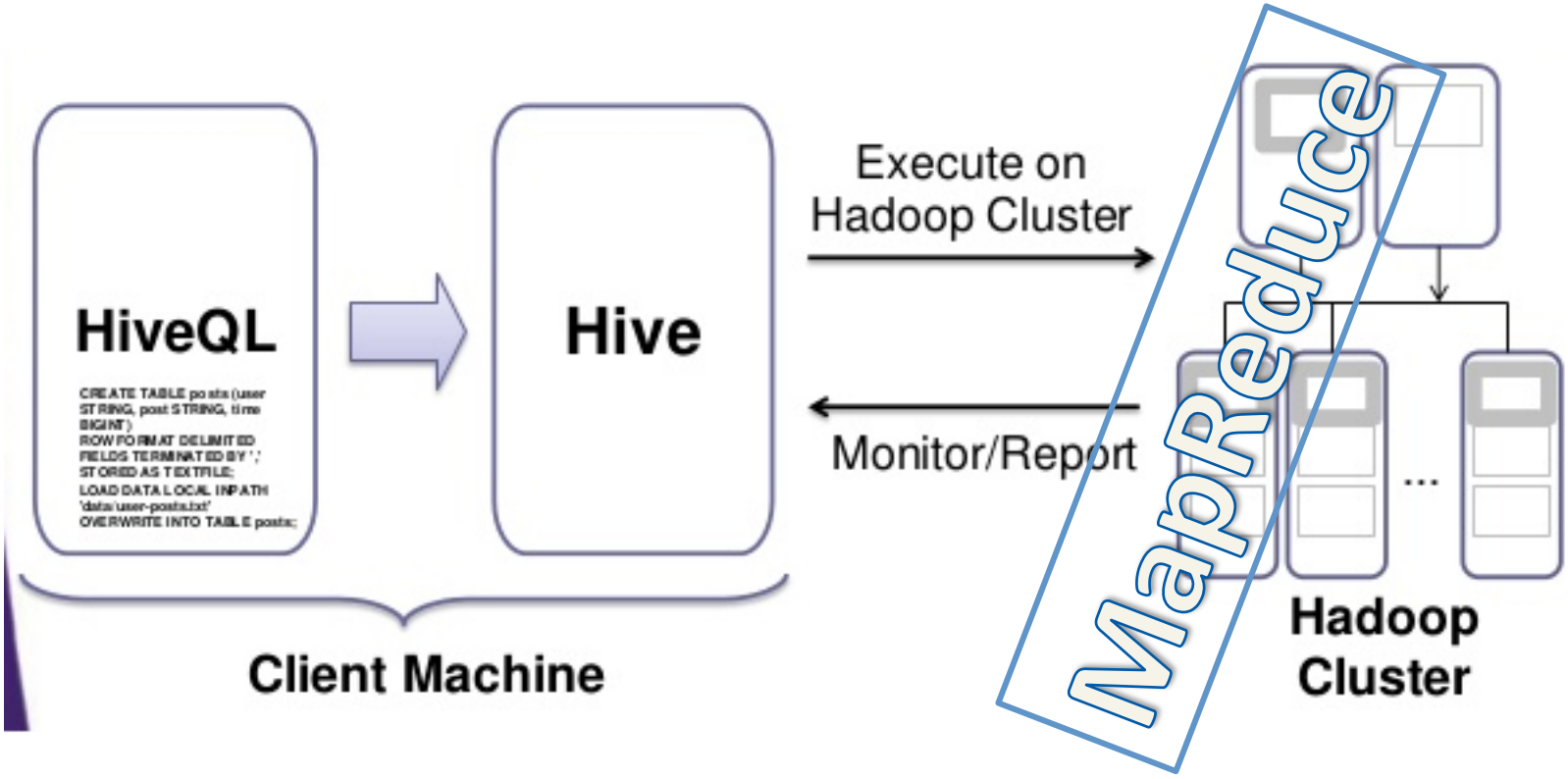
```
tier1.sinks.sink1.type = HDFS
tier1.sinks.sink1.fileType = DataStream
tier1.sinks.sink1.channel = channel1
tier1.sinks.sink1.hdfs.path = hdfs://localhost:8020/user/cloudera/flume/events
```

  - Salvar cambios y reiniciar el servicio
  - Iniciar el servicio Yarn
  - En la consola lanzar de nuevo `telnet localhost 9999`
  - Teclear algo y e ir a la ruta flume/events para ver si tenemos algo

Hive

- Data Warehouse para Hadoop
- Permite ETLs (Extract/Transform/Load)
- Permite asociar la extructura con variedad de formatos
  - De tabla lógica → a localización física
  - De tabla lógica → a Physical Data Format Handle (SerDe)
- Integrado con HDFS, HBase, MongoDB, etc.
- Las consultas se ejecutan con procedimiento MapReduce
- Creado por Facebook y luego liberado

La idea


HiveQL
CREATE TABLE posts (user STRING, post STRING, time BIGINT)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
STORED AS TEXTFILE;
LOAD DATA LOCAL INPATH 'data/user-posts.txt'
OVERWRITE INTO TABLE posts;
Hive
Client Machine
Execute on Hadoop Cluster
Monitor/Report
MapReduce
Hadoop Cluster

¿Por qué usar Hive?

MBIT School
Madrid Business Intelligence Technology

- Facilita el uso de un sistema MapReduce ya que permite la interacción mediante scripts tipo SQL (HiveQL)
- La herramienta lo compila como un proceso MapReduce
- De esta forma nos aprovechamos de los beneficios tanto de MapReduce como de HDFS
  - Tolerante a fallos
  - Robusto
  - Escalable
- Como exige compilación/traducción a MapReduce, muchas veces obtendremos peor rendimiento que programando el proceso en Java

¿Qué permite Hive?

MBIT School
Madrid Business Intelligence Technology

- DDL: create table, create view, create index…
- DML: select, where, joint, group by, order by…
- Pluggable User Defined Functions – UDFs (ej. from_unixtime)
- Pluggable User Defined Aggregate Functions – UDAFs (count, avg, …)
- Serialización, deserialización (SerDes)
- Pluggable custom input/output formats
- …

¿Qué no permite?

- Online Transaction Processing (OLTP) workload (baja latencia)
- Subconsultas correlacionadas (una consulta que usa datos de la consulta interna)
- Malo rendimiento con cantidades pequeñas de datos

Hive metastore

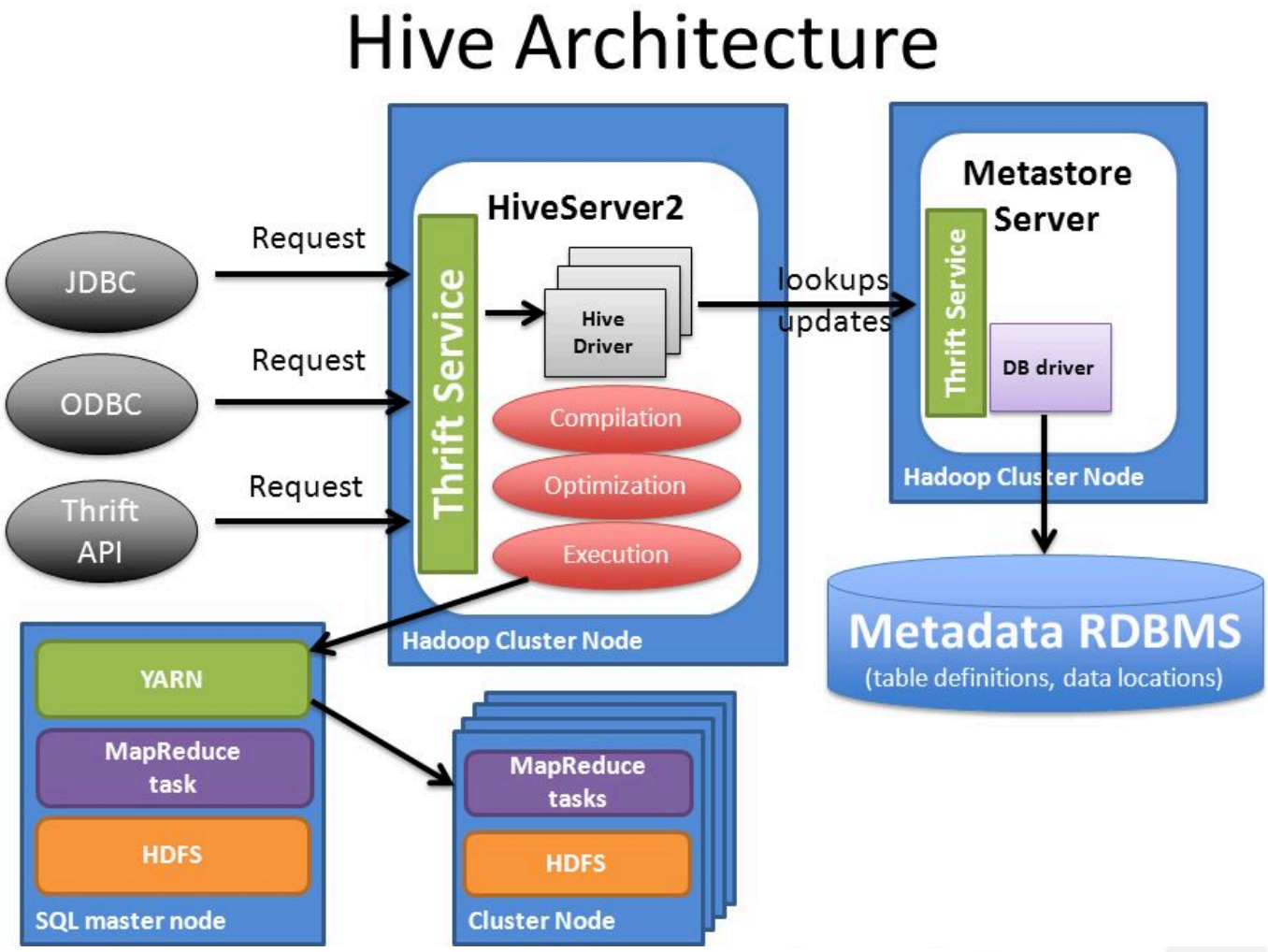
MBIT School
Madrid Business Intelligence Technology

- Una BBDD relacional que puede ser MySQL, ProgreSQL…
- Por defecto Derby
- Es el repositorio de metadatos
- Permite saber la localización y formato de los datos a la hora de interpretar las consultas
- Si se emplea HCatalog se hace accesible el repositorio a aplicaciones externas
- Modos:
  - Local
  - Remoto

Las tablas

- Se almacenan en HDFS
- Cada tabla corresponde a un único directorio
- En un único directorio puede haber muchos ficheros
- Los metadatos permiten interpretar correctamente los datos
- Cada tabla pertenece solo a una base de datos específica
- La ruta por defecto es `/user/hive/warehouse`

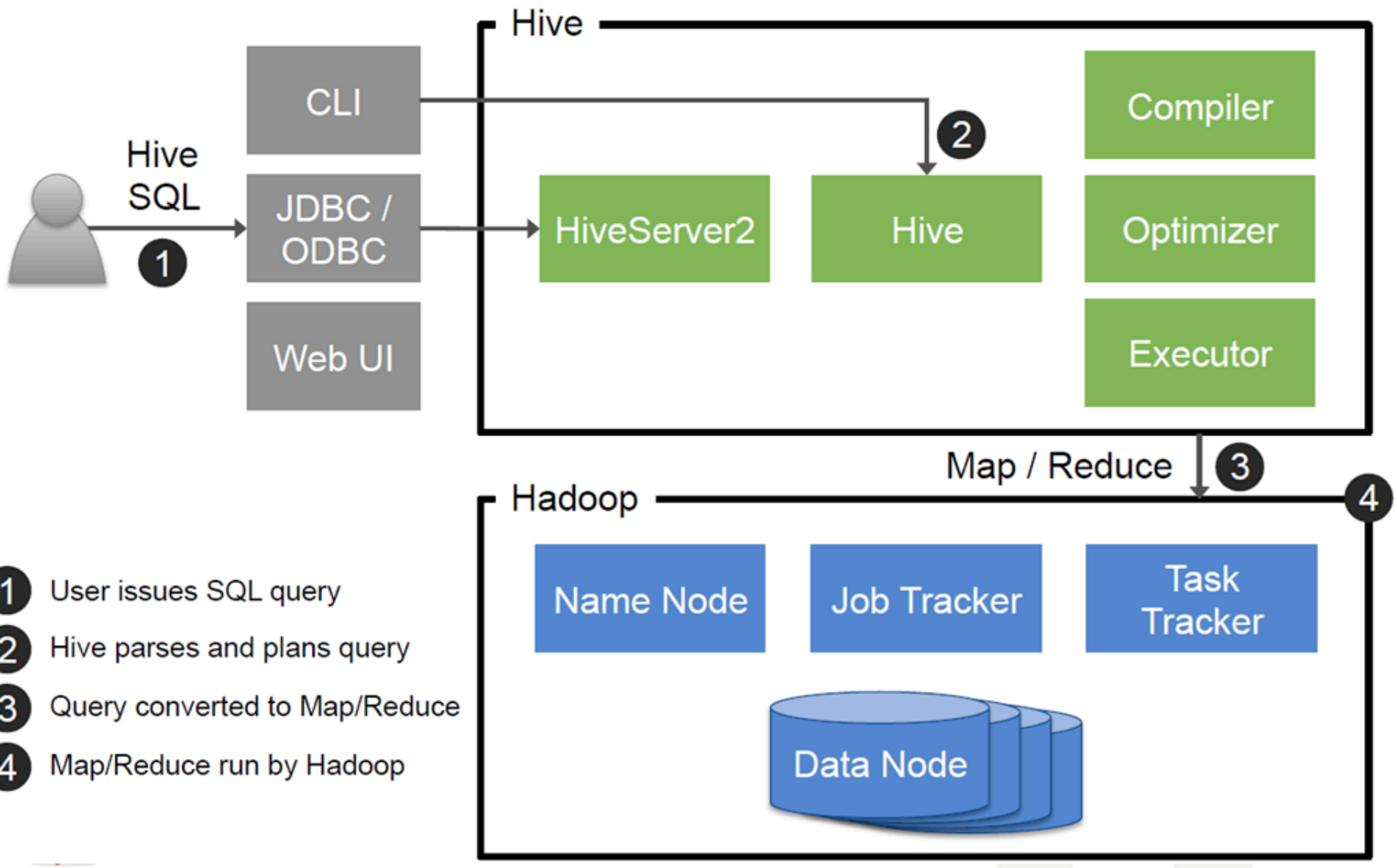
Esquema


Hive Architecture
JDBC
ODBC
Thrift API
Request
Request
Request
HiveServer2
Thrift Service
Hive Driver
Compilation
Optimization
Execution
Hadoop Cluster Node
lookups
updates
Metastore Server
Thrift Service
DB driver
Hadoop Cluster Node
Metadata RDBMS
(table definitions, data locations)
YARN
MapReduce task
HDFS
SQL master node
MapReduce tasks
HDFS
Cluster Node

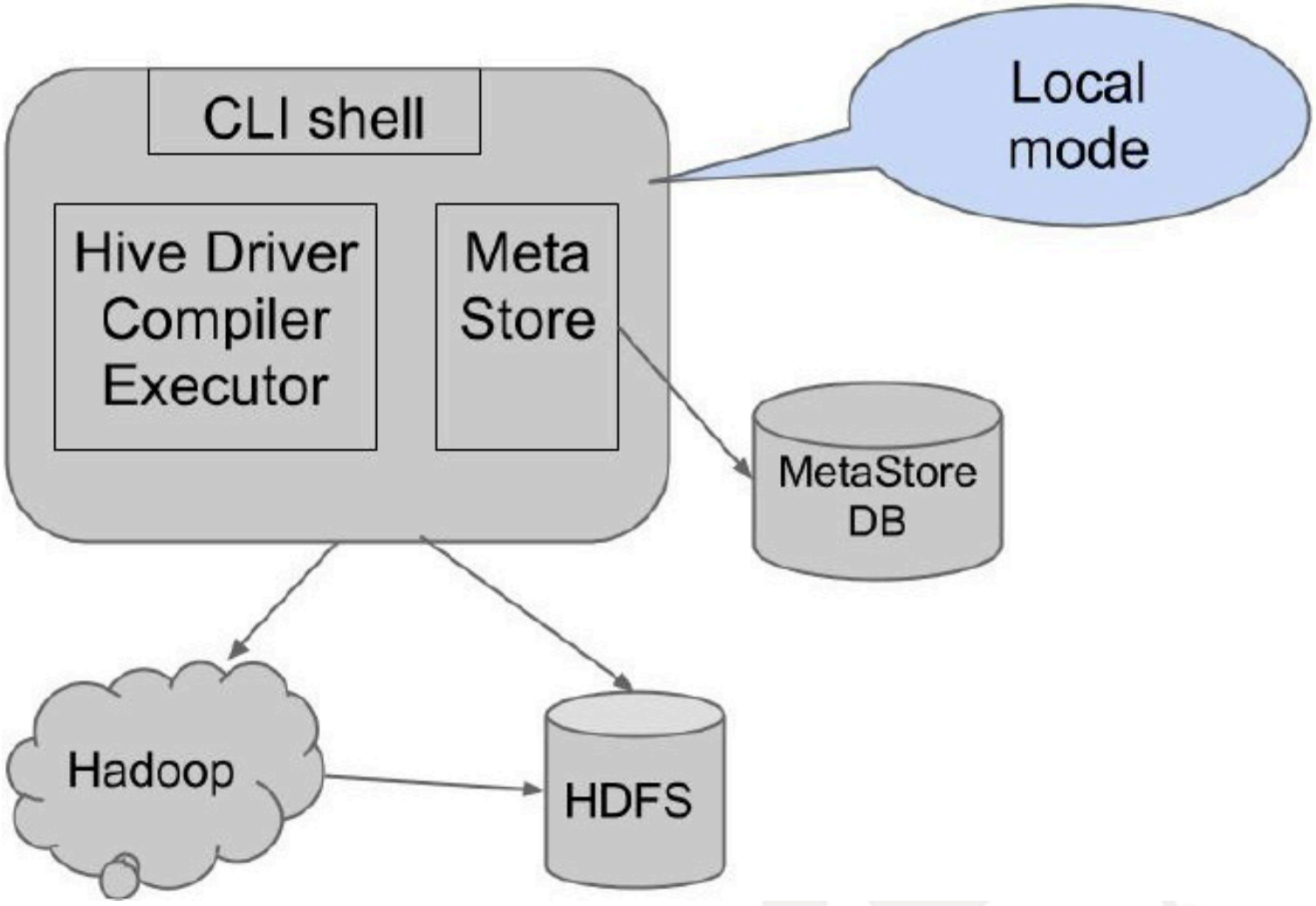
Particiones

- Por defecto los datos son almacenados en un único directorio “todos juntos”
- Pero imagenemos que guardamos históricos de cotizaciones bursátiles, pero nuestra estrategia solo emplea datos de los últimos 15 días, ¿es eficiente guardar todo junto?
- Quizá mejor particionar por año y mes, quizá así y cómo máximo, solo tendremos que acceder a dos particiones
- El tamaño del “fichero” a consultar influye en el rendimiento

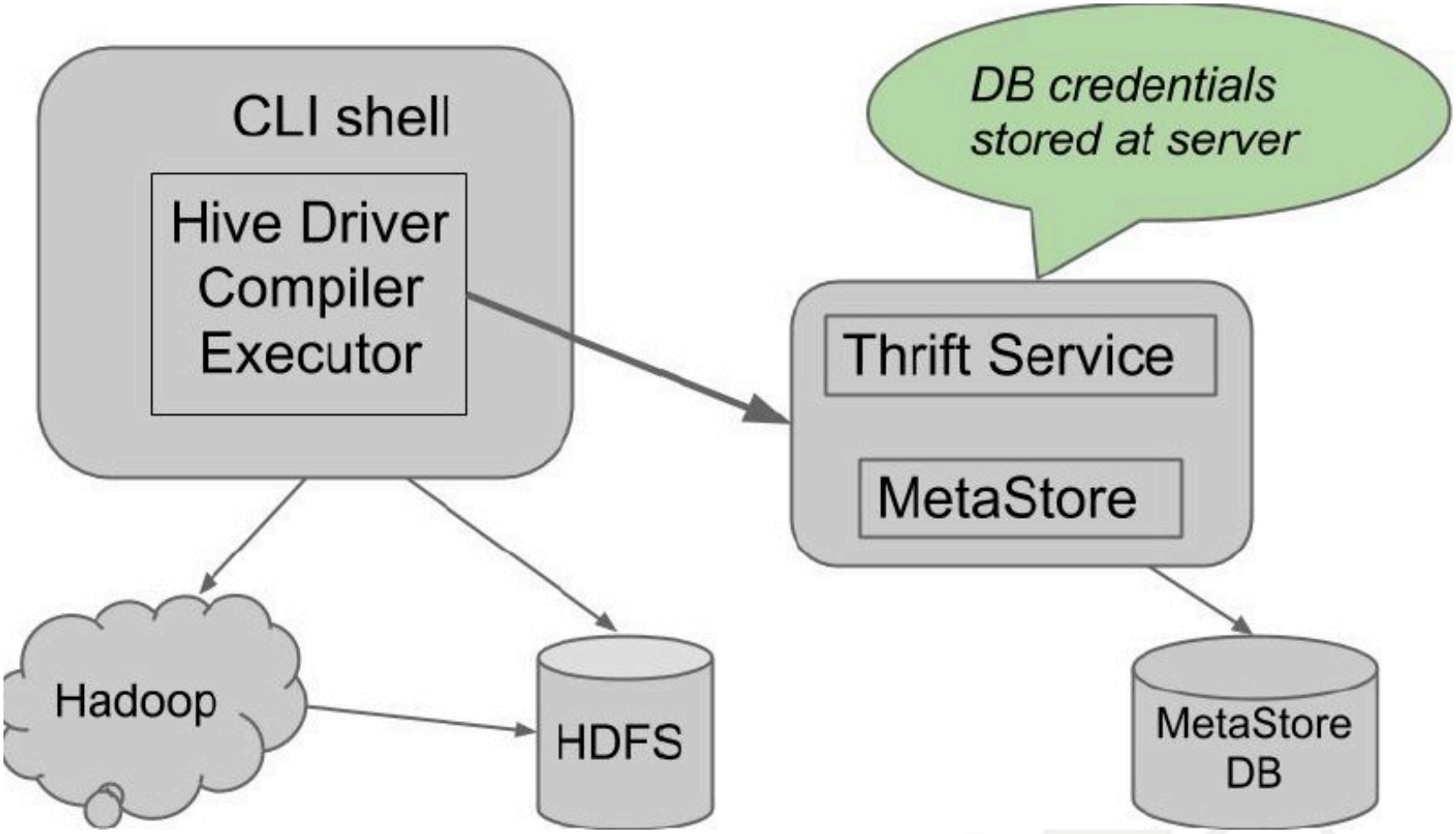
Hive: The SQL Interface to Hadoop



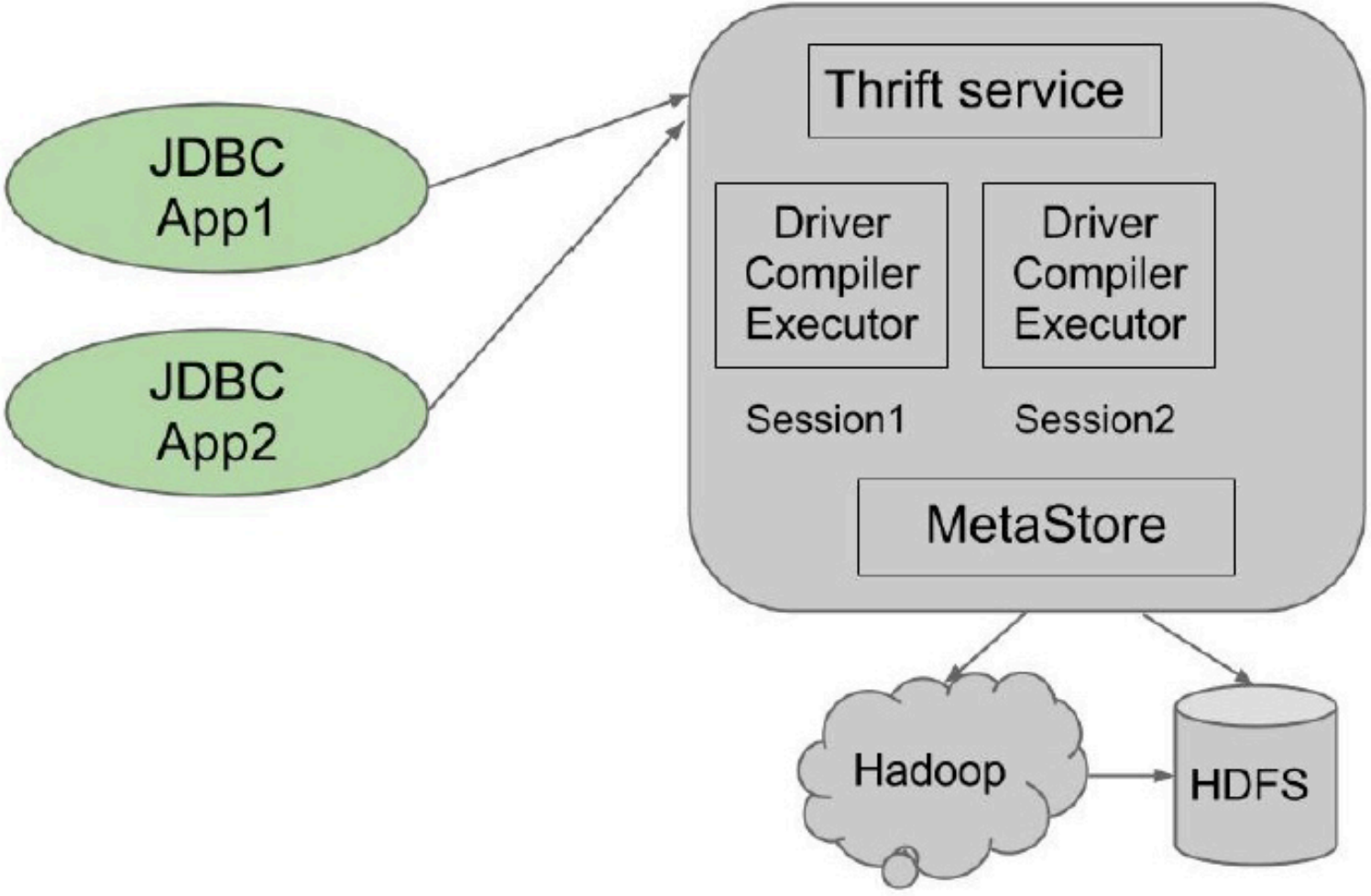
Arquitectura Hive (local)



Arquitectura Hive (remoto)



Hive Server


JDBC
App1
JDBC
App2
Thrift service
Driver
Compiler
Executor
Driver
Compiler
Executor
Session1
Session2
MetaStore
Hadoop
HDFS

Destacados problemas de Hive

MBIT School
Madrid Business Intelligence Technology

- Orientado al procesamiento batch
- No permite consultas interactivas
  - *Interactive means that you don't know your questions in advance. You ask ad-hoc questions. Then, based on the answer, you ask the next question. Here, speed of execution becomes a challenge as the questioner does not have many hours' worth of patience, which it would take for a MapReduce job to run*
- Los tiempos de respuestas a veces son elevados

Algunos comandos

- Para entrar en la consola hive: `$ hive`
- `$ create database mydb;`
- `$ create table mytable;`
- Crear una table copiando la estructura de otra:
  - `$ create table copia like mytable;`
- `$ show databases;`
- `$ use mytable;`
- `$ create table mytable (id INT, texto STRING, id2 DOUBLE) row format delimited fields terminated by ',' stored as textfiles;`
- `$ describe mytable;`

Data types

- TINYINT
- SMALLINT
- BIGINT
- INT
- STRING
- VARCHAR
- CHAR
- TIMESTAMP
- DATES
- DECIMAL
- DOUBLE
- BOOLEAN
- BINARY
- NULL

LEARNING BY EXAMPLE

MBIT School
Madrid Business Intelligence Technology

Observar qué aparece en /usr/hive/warehouse

```
create table table_name (
  id                  int,
  dtDontQuery         string,
  name                string
)
partitioned by (date string)
```

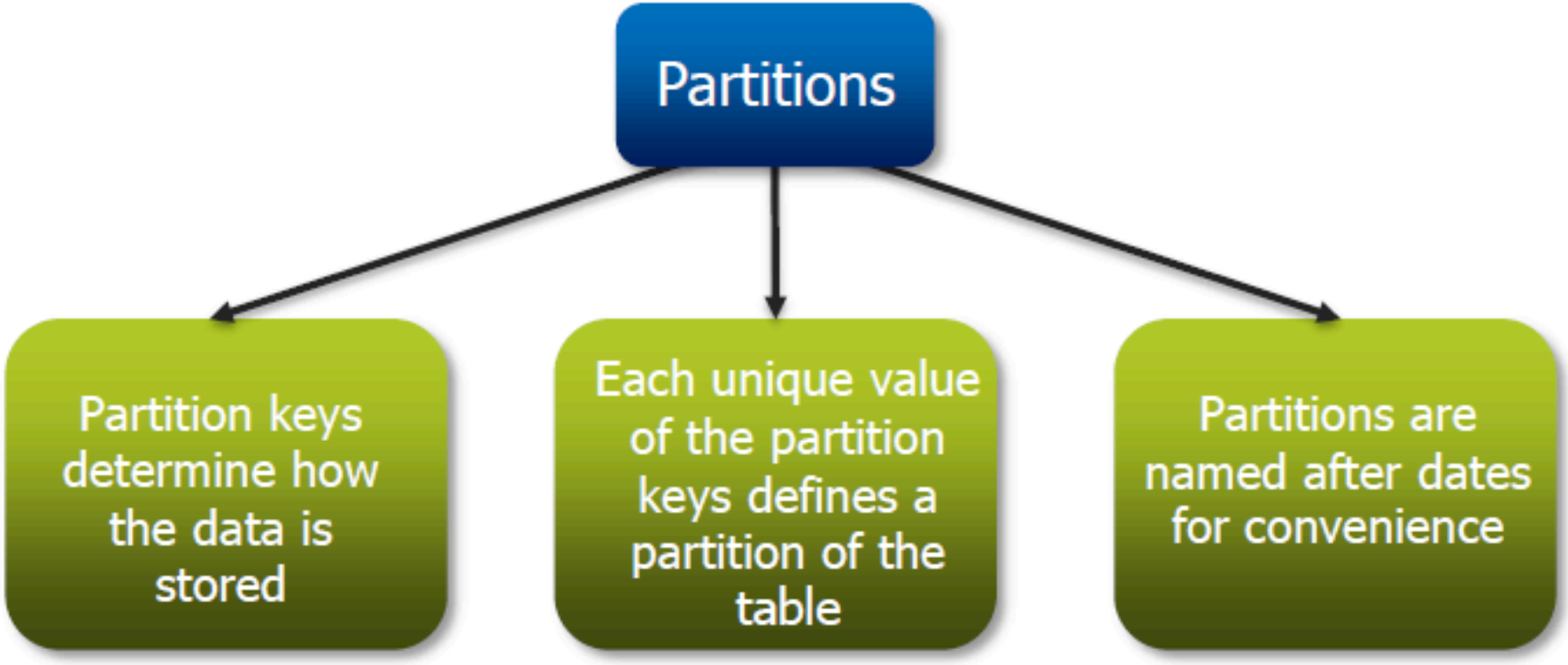
```
CREATE TABLE table2 LIKE table1
```

LEARNING BY EXAMPLE

```
CREATE TABLE page_view(viewTime INT, userid BIGINT,
     page_url STRING, referrer_url STRING,
     ip STRING COMMENT 'IP Address of the User')
 COMMENT 'This is the page view table'
 PARTITIONED BY(dt STRING, country STRING)
 ROW FORMAT DELIMITED
   FIELDS TERMINATED BY '\001'
STORED AS SEQUENCEFILE;
```

```
CREATE EXTERNAL TABLE page_view(viewTime INT, userid BIGINT,
     page_url STRING, referrer_url STRING,
     ip STRING COMMENT 'IP Address of the User',
     country STRING COMMENT 'country of origination')
 COMMENT 'This is the staging page view table'
 ROW FORMAT DELIMITED FIELDS TERMINATED BY '\054'
 STORED AS TEXTFILE
 LOCATION '<hdfs_location>';
```

LEARNING BY EXAMPLE



LEARNING BY EXAMPLE

- ‘Partitioned by‘ is used to divided the table into the Partition and can be divided in to buckets by using the ‘Clustered By‘ command.

```
CREATE TABLE user_info_bucketed(user_id BIGINT, firstname STRING, lastname STRING)
COMMENT 'A bucketed copy of user_info'
PARTITIONED BY(ds STRING)
CLUSTERED BY(user_id) INTO 256 BUCKETS;
```

Ojo con el orden cuando los datos proceden de una consulta, deben aparecer al final

The dynamic partition columns must be specified last among the columns in the SELECT statement and in the same order in which they appear in the PARTITION() clause.

LEARNING BY EXAMPLE

MBIT School
Madrid Business Intelligence Technology

To rename existing table to a new name. If a table with new name already exists then an error is returned:

```
ALTER TABLE old_table_name RENAME TO new_table_name;
```

To rename the columns of an existing table. Be sure to use the same column types, and to include an entry for each preexisting column:

```
ALTER TABLE old_table_name REPLACE COLUMNS (col1 TYPE, ...);
```

To add columns to an existing table:

```
ALTER TABLE tab1 ADD COLUMNS (c1 INT COMMENT 'a new int column', c2 STRING DEFAULT 'def val');
```

Data Manipulation Language (DML)

MBIT School
Madrid Business Intelligence Technology

- `$ load data local inpath 'home/loquesea' into table mytable;`
- `$ select count(*) from mytable;`
- INSERT OVERWRITE is used to overwrite the existing data in the table or partition
- INSERT INTO is used to append the data into existing data in a table

```
INSERT OVERWRITE TABLE user_active
SELECT user.*
FROM user
WHERE user.active = 1;
```

- `$ drop table mytable;`

Insertando datos

1. Mover o copiar el fichero con un comando hdfs al directorio correspondiente de hive/warehouse
2. LOAD DATA INPATH “fichero” INTO TABLE myTable;

Algunas referencias

- https://www.edureka.co/blog/hive-commands-with-examples

- https://cwiki.apache.org/confluence/display/Hive/LanguageManual+DML

Hue editor and queries

MBIT School
Madrid Business Intelligence Technology

HUE | Query Editors | Data Browsers | Workflows | Search

Hive Editor | Query Editor | My Queries | Saved Queries | History

Assist | Settings

default

Tables (0)

The database has no tables

```
use losvuelos;
show tables;
```

Execute | Save as... | Explain | Format | or create a | New query

Si quiero saber la estructura de una tabla

MBIT School
Madrid Business Intelligence Technology

- *`Describe schema.table`*

- Otros comandos interesantes son aquellos que manipulan fechas como:
  - `SELECT YEAR(fecha_compra) FROM…`
  - La fecha tiene que ser una fecha (timestamp por ejemplo)

Ejercicio básico

- Cogemos El Quijote
- Aplicamos el wordcount
- Creamos una tabla en Hive preparada para guardar los resultados del wordcount
- Comprueba que la estructura de esa tabla es la adecuada (comando *`describe`*)
- Cargamos los datos
- Buscamos las palabras con frecuencia mayor de X

Continuación…

- Hagamos lo mismo ahora para otro libro
- Creemos una tabla “mezcla” con las resultados de ambos libros
- Insertamos datos en la tabla “mezcla” haciendo un join por la palabra y que aparezca la frecuencia en ambos libros
- ¿Cuáles son las palabras que más veces aparecen considerando ambos libros?

Typical example dataset

MBIT School
Madrid Business Intelligence Technology

ASA Sections on:
Statistical Computing
Statistical Graphics

[Computing, Graphics]
[Awards, Data expo, Video library]
[Events, News, Newsletter]

Search

Data expo

Airline on-time performance

Have you ever been stuck in an airport because your flight was delayed or cancelled and wondered if you could have predicted it if you'd had more data? This is your chance to find out.

The results

We had a total of nine entries, and turn out at the poster session at the JSM was great, with plenty of people stopping by to find out why their flights were delayed.

The data

The data consists of flight arrival and departure details for all commercial flights within the USA, from October 1987 to April 2008. This is a large dataset: there are nearly 120 million records in total, and takes up 1.6 gigabytes of space compressed and 12 gigabytes when uncompressed. To make sure that you're not overwhelmed by the size of the data, we've provide two brief introductions to some useful tools: linux command line tools and sqlite, a simple sql database.

Data expo 09

- **Posters & results**
- Competition description
- Download the data
- Supplemental data sources
- Using a database
- Intro to command line tools

Source: http://stat-computing.org/dataexpo/2009/

Variable descriptions

	Name	Description
1	Year	1987-2008
2	Month	1-12
3	DayofMonth	1-31
4	DayOfWeek	1 (Monday) - 7 (Sunday)
5	DepTime	actual departure time (local, hhmm)
6	CRSDepTime	scheduled departure time (local, hhmm)
7	ArrTime	actual arrival time (local, hhmm)
8	CRSArrTime	scheduled arrival time (local, hhmm)
9	UniqueCarrier	unique carrier code
10	FlightNum	flight number
11	TailNum	plane tail number
12	ActualElapsedTime	in minutes
13	CRSElapsedTime	in minutes
14	AirTime	in minutes
15	ArrDelay	arrival delay, in minutes
16	DepDelay	departure delay, in minutes
17	Origin	origin IATA airport code
18	Dest	destination IATA airport code
19	Distance	in miles
20	TaxiIn	taxi in time, in minutes
21	TaxiOut	taxi out time in minutes
22	Cancelled	was the flight cancelled?
23	CancellationCode	reason for cancellation (A = carrier, B = weather, C = NAS, D = security)
24	Diverted	1 = yes, 0 = no
25	CarrierDelay	in minutes
26	WeatherDelay	in minutes
27	NASDelay	in minutes
28	SecurityDelay	in minutes
29	LateAircraftDelay	in minutes

http://stat-computing.org/dataexpo/2009/the-data.html

Guía de prácticas

MBIT School
Madrid Business Intelligence Technology

- Trabajaremos con los datos del 2008 por ejemplo.
- Debemos seguir los siguientes pasos (a modo de ejercicio):
  1. Crear en HDFS un directorio llamado ejercicioHive
  2. Copiar en ese directorio el archivo descargado con los datos del año 2008
  3. Hacer un ls sobre el directorio de hadoop y comprobar que todo está correcto (también se puede comprobar con el File Browser)

Comprueba que todo está correcto

MBIT School
Madrid Business Intelligence Technology

Browse Directory

/user/cloudera/ejercicioHive | Go!

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	cloudera	cloudera	657.48 MB	Sun Jan 08 19:35:16 +0100 2017	1	128 MB	2008.csv

Hadoop, 2016.

Aspectos técnicos

- Cuidado con la CDH quickstart 5.8
- En función de la configuración de tu equipo puede que tengas algún problema de rendimiento
- Prueba a apagar todos los servicios e iniciar en este order:
  - HDFS
  - YARN
  - Zookeeper
  - HIVE
  - HUE
- Por precaución, apaga todo los servicios antes de salir de la MV

Guía de prácticas

MBIT School
Madrid Business Intelligence Technology

- Crea la estructura de la tabla vuelos en hive. Se recomienda emplear los tipos de datos int, smallint, string…
- Esta tabla debe pertenecer al esquema ‘losvuelos’
- Para cargar los datos hay dos formas:
  - LOAD DATA INPATH ’ruta/2008.csv' OVERWRITE INTO TABLE losvuelos.vuelos PARTITION(YearPartition = 2008);
  - hdfs dfs –mv ‘ruta/2008.csv’ /user/hive/warehouse/vuelos/

losvuelos.db

Browse Directory

/user/hive/warehouse Go!

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxrwxrwx	cloudera	supergroup	0 B	Sun Jan 08 21:03:26 +0100 2017	0	0 B	losvuelos.db

Crear tabla usando interfaz

MBIT School
Madrid Business Intelligence Technology

Query Editors | Data Browsers | Workflows | Search

Metastore Manager

losvuelos

Tables (0)

The database has no tables

Step 1: Choose File | Step 2: Choose Delimiter | Step 3: Define Columns

Name Your Table and Choose A File

Table Name: vuelos2

Name of the new table. Table names must be globally unique. Table names tend to correspond to the directory where the data will be stored.

Description: Optional

Use a table comment to describe the table. For example, note the data's provenance and any caveats users need to know.

Input File: /user/cloudera/ejercicioHive/2008.csv

The HDFS path to the file on which to base this new table definition. It can be compressed (gzip) or not.

Import data from file: ☑

Check this box to import the data in this file after creating the table definition. Leave it unchecked to define an empty table.

Warning: The selected file is going to be moved during the import.

Guía de prácticas

- ¿Cuántos registros tenemos?
- ¿Cuántos aeropuestos distintos?
- ¿Cuántos vuelos cancelados?
- ¿Cuál es el aeropuerto con más vuelos cancelados?
- ¿Puedes analizar los destinos de los aviones cancelados (se puede tener en cuenta o no el aeropuerto de origen?
- Experimenta
- Presenta

IMPALA

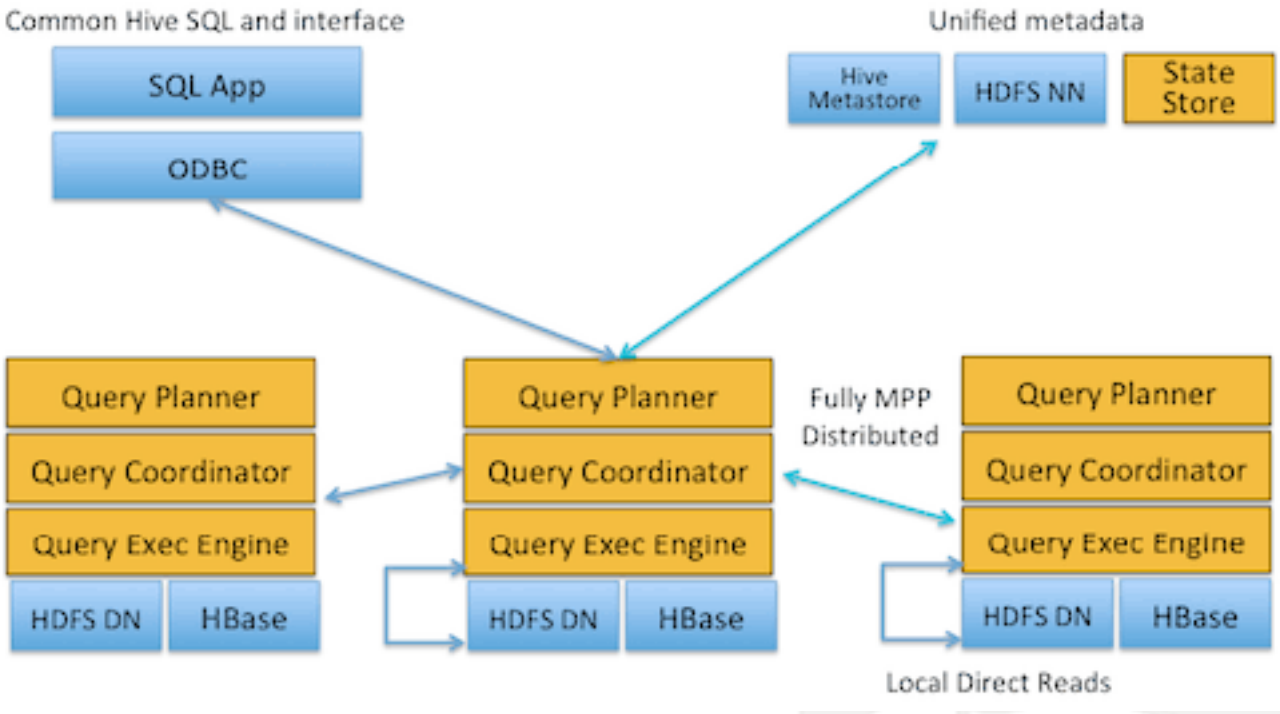
- Herramienta genérica de consulta SQL
- Muy rápida
- Integrada con Hadoop
- Pero NO MapReduce, emplea un motor propio
- Implementada en C++

Características de Impala

MBIT School
Madrid Business Intelligence Technology

- Se ejecuta como servicio distribuido. Existe un servicio Impala en cada DataNode
- Varios interfaces disponibles
- Utilize el Hive metastore
- Procedimiento de ejecución
  - Llega la consulta vía ODBC/JDBC
  - Se descompone la petición
  - El coordinador planifica y ejecuta

Arquitectura



Hive vs Impala

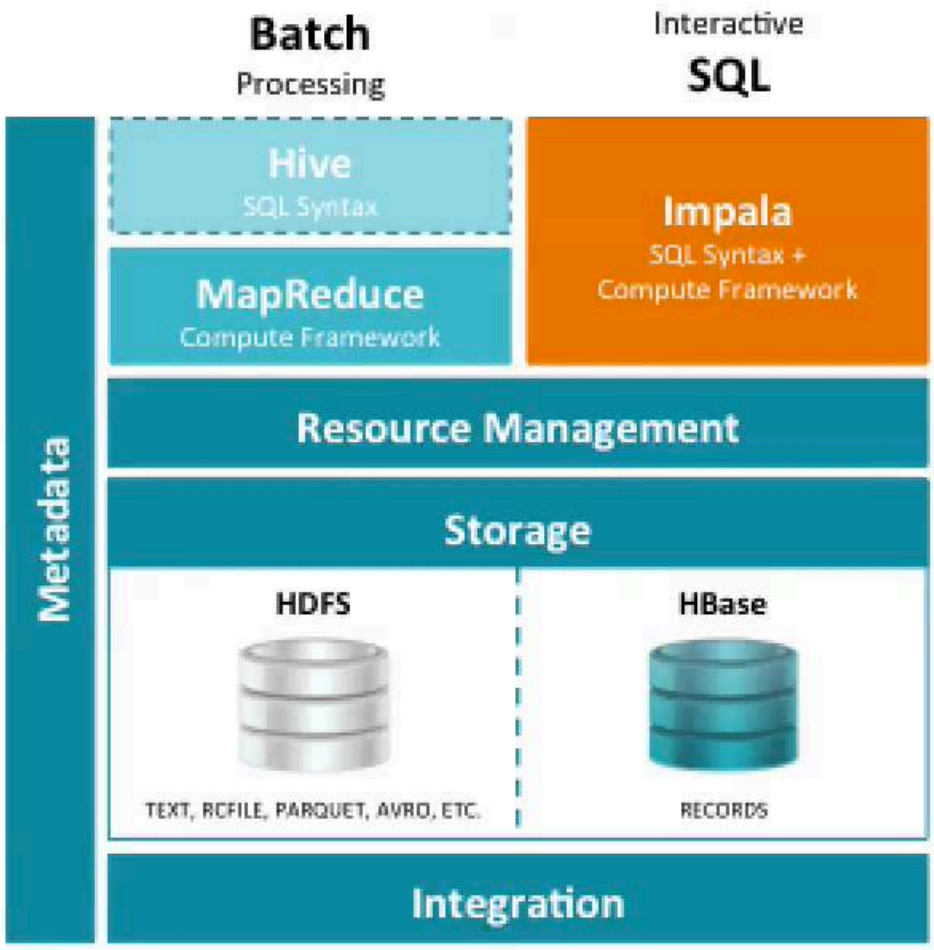
MBIT School
Madrid Business Intelligence Technology

Shares Everything Client-Facing

- Metadata (table definitions)
- ODBC/JDBC drivers
- SQL syntax (Hive SQL)
- Flexible file formats
- Machine pool
- Hue GUI

But Built for Different Purposes

- **Hive:** runs on MapReduce and ideal for batch processing
- **Impala:** native MPP query engine ideal for interactive SQL



Algunos comandos

```
$ hdfs dfs -put tabl.csv /user/cloudera/sample_data/tabl
$ hdfs dfs -ls /user/cloudera/sample_data/tabl
```

```
DROP TABLE IF EXISTS tabl;
-- The EXTERNAL clause means the data is located outside the central location
-- for Impala data files and is preserved when the associated Impala table is dropped.
-- We expect the data to already exist in the directory specified by the LOCATION clause.
CREATE EXTERNAL TABLE tabl
(
   id INT,
   col_1 BOOLEAN,
   col_2 DOUBLE,
   col_3 TIMESTAMP
)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
LOCATION '/user/cloudera/sample_data/tabl';
```

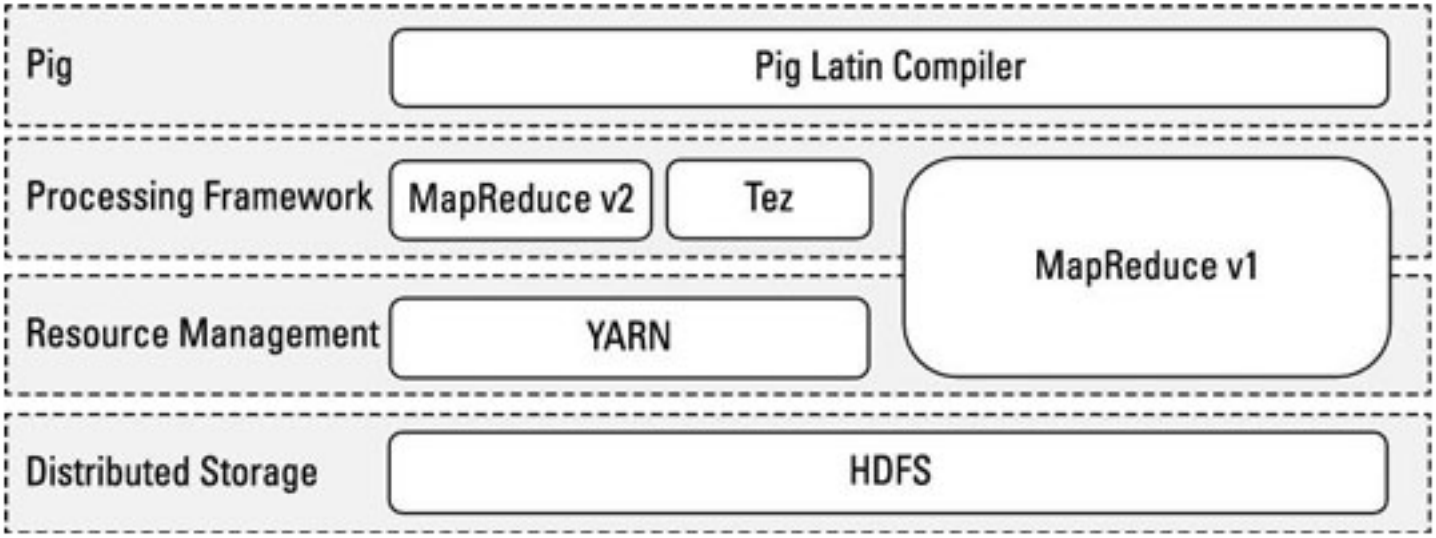
Source: https://www.cloudera.com/documentation/enterprise/5-8-x/topics/impala_tutorial.html

PIG

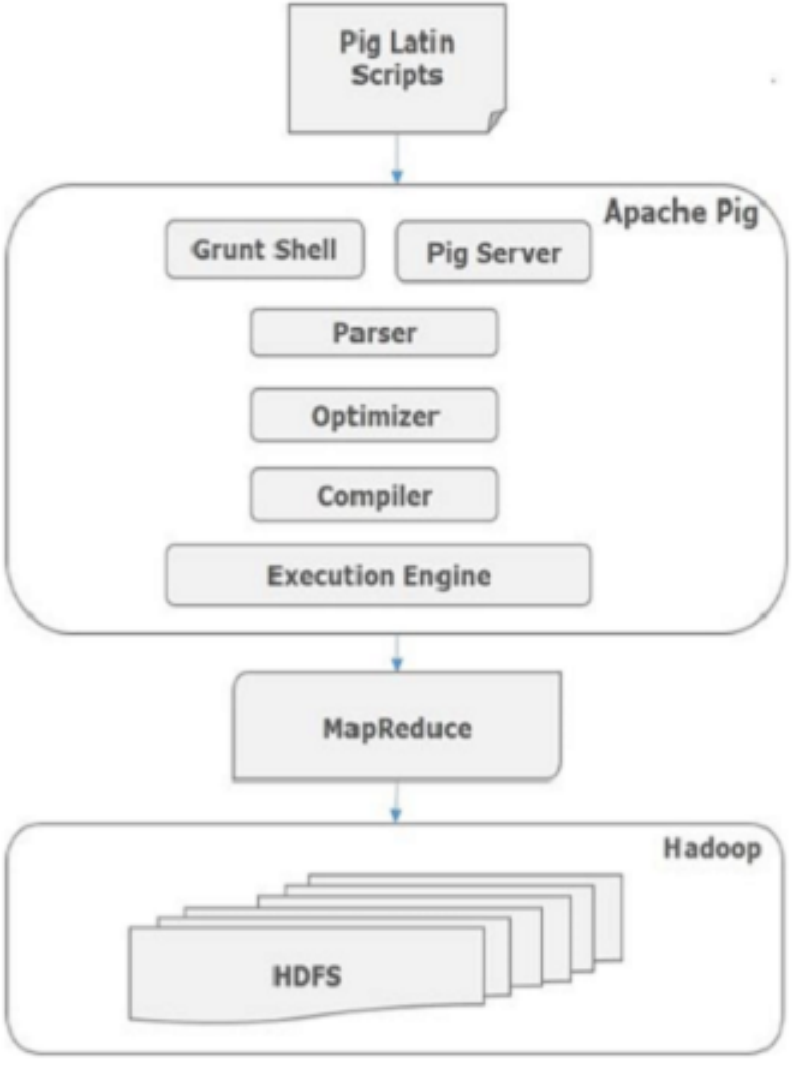
- Pig is a simple-to-understand data flow language used in the analysis of large data sets. Pig scripts are automatically converted into MapReduce jobs by the Pig interpreter, so you can analyze the data in a Hadoop cluster even if you aren't familiar with MapReduce

Pig Latin

- The Pig Latin compiler converts the Pig Latin code into executable code. The executable code is either in the form of MapReduce jobs or it can spawn a process where a virtual Hadoop instance is created to run the Pig code on a single node.



Pig Architecture



Learning by example

```
-- First, we load the raw data from a test dataset
RAW_DATA = LOAD '/home/michiard/work/input/AIRLINE/2008.csv' USING PigStorage(',') AS
    (year: int, month: int, day: int, dow: int,
    dtime: int, sdtime: int, arrtime: int, satime: int,
    carrier: chararray, fn: int, tn: chararray,
    etime: int, setime: int, airtime: int,
    adelay: int, ddelay: int,
    scode: chararray, dcode: chararray, dist: int,
    tintime: int, touttime: int,
    cancel: chararray, cancelcode: chararray, diverted: int,
    cdelay: int, wdelay: int, ndelay: int, sdelay: int, latedelay: int);

/*

-----------------------------------------
--  AGGREGATE OUTBOUND TRAFFIC, PER IATA AIRPORT CODE
-----------------------------------------
-- Group by the IATA code of the departure airport
SOURCE_IATA_GROUP = GROUP RAW_DATA BY scode;
-- Count the number of flights out-bound that particular airport
OUTBOUND_IATA_COUNT = FOREACH SOURCE_IATA_GROUP GENERATE group as IATA, COUNT(RAW_DATA) AS
num_out_flights;
```

Learning by example

```
-- Load input data from local input directory
A = LOAD './local-input/WORD_COUNT/sample.txt';

-- Parse and clean input data
B = FOREACH A GENERATE FLATTEN(TOKENIZE((chararray)$0)) AS word;
C = FILTER B BY word MATCHES '\\w+';

-- Explicit the GROUP-BY / SHUFFLE Phase
D = GROUP C BY word;

-- Generate output data in the form: <word, counts>
E = FOREACH D GENERATE group, COUNT(C);

-- Store output data in local output directory
store E into './local-output/WORD_COUNT/';
```

Resumen

- Hemos visto una serie de herramientas cada una para una cuestión concreta
- Quizá, las más importantes sean Hive e Impala
- Sqoop y Flume nos dan acceso a procedimientos alternativos para introducir datos en nuestros sistemas
- PIG nos proporciona un entorno de manipulación adicional

THANK YOU
merci
spasiba
kam ouen
gratzias
manana
mahalo
cheers
toda
grassie
thank you
danki
grazie
tak
kitos
hvala
gracias
welalin
mahalo
danki
thanks
takk
gracias
domo arrigato
danke
grattitude
merci
dankon
kitos
thanks
na gode
mesi
modupe
talofa
miigwetch
takk
dziekuje

KEEP
CALM
AND
ANALYZE
BIG DATA