

plain concepts 

DÍA 2: AGENDA

- Ingestión de datos en Hadoop
- SQL en Hadoop: Hive
- Lenguajes de flujo de datos: PIG
- **Ejercicio práctico del día**



HIVE

HIVE - ¿QUE ES?

- Subproyecto de Apache Hadoop para construir un Data Warehouse sobre el clúster
- Estructura los datos mediante conceptos clásicos de bases de datos
 - Tablas, particiones, filas, columnas
- Usando HiveQL, un lenguaje ANSI SQL
- Schema on Read
- Se encarga de ejecutar los jobs de MapReduce (o Tez) de forma automática
- Muchísimo mas sencillo que utilizar MapReduce (o Tez) directamente

HIVE - ¿QUE NO ES?

- Un RDBMS (Relational DataBase Management System)
 - Hay una base de datos donde almacena metadatos, pero los datos se almacenan en ficheros, como hasta ahora
- Diseñado para sistemas OLTP (Online Transaction Processing)
 - No hay consultas en tiempo real (esto es una verdad a medias)
 - No hay actualización de filas (esto también es una verdad a medias)

HIVE VS RDBMS

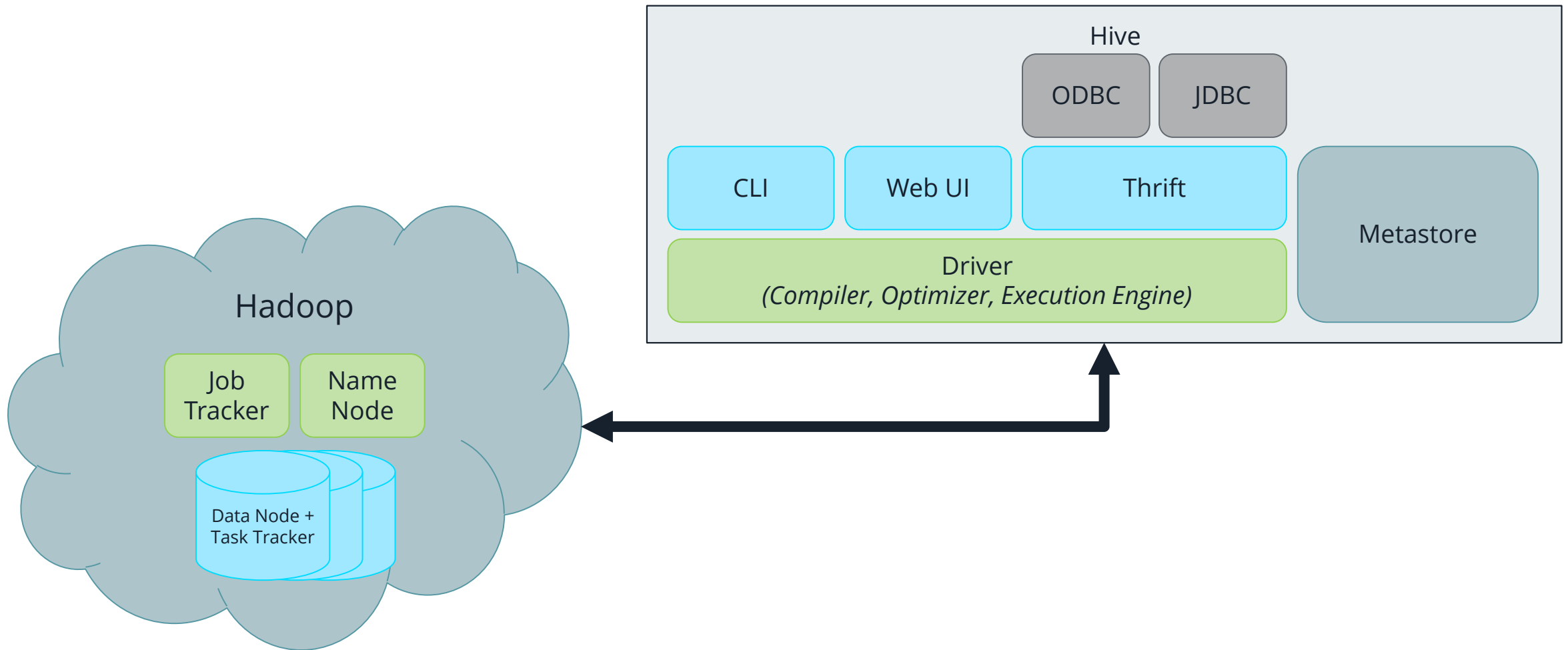
- RDBMS

- Schema On Write
- SQL
- Indexing
- Updates
- Integridad referencial
- Cost Based Query Optimization
- OLTP

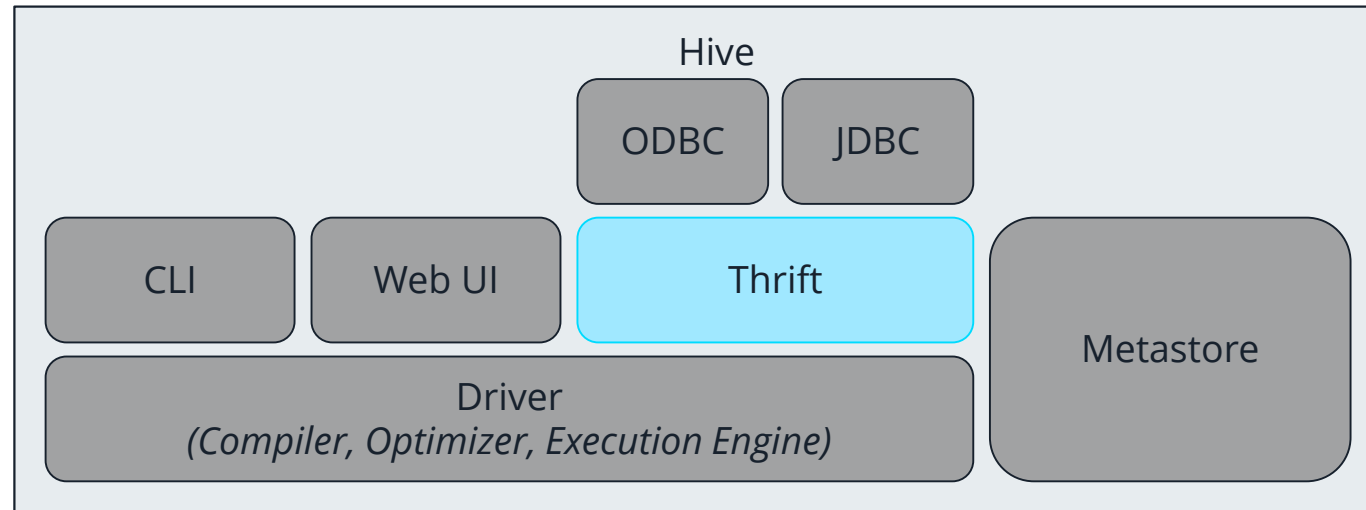
- Hive

- Schema On Read
- SQL
- Indexing
- Updates...
- Cost Based Query Optimization

ARQUITECTURA

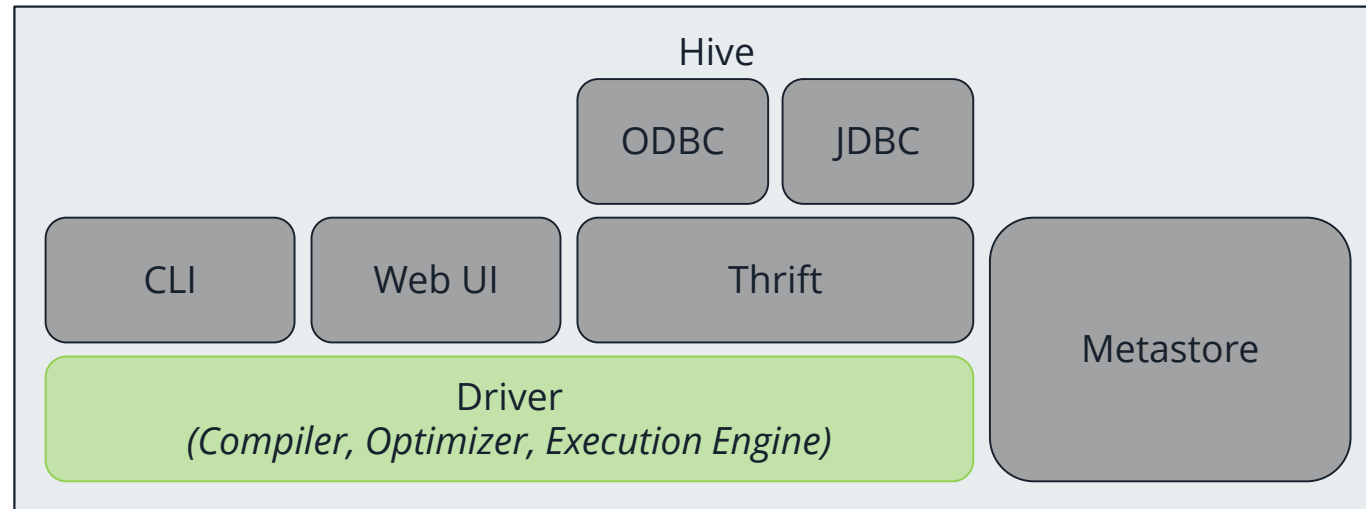


HIVE THRIFT SERVER



- Framework para servicios multilenguaje
- Servidor en Java
- Soporta clientes en distintos lenguajes

HIVE DRIVER



- **Driver** – Gestiona el ciclo de vida de una consulta HiveQL
- **Query Compiler** – Compila HiveQL en un DAG de tareas MR
- **Execution Engine** – Ejecuta el plan generado por el compilador e interactúa con Hadoop

COMPILADOR Y OPTIMIZADOR

- Convierte una consulta en HiveQL en un plan a ser ejecutado
- Analizador Semántico
 - Verifica la información del esquema, los tipos, la conversión de tipos implícita...
- Optimizador
 - Combina JOINS para reducir el numero de trabajos de MR, filtra columnas lo antes posible para minimizar transferencias de datos...
- Generador de plan
 - Crea un DAG de trabajos MR

CREACION DE UN ESQUEMA DE DATOS

- Montar un DW con Hive es sencillo
- Definimos
 - Campos
 - Formato de almacenamiento
 - Localización de los datos
 - Otros (particionado, compresión, etc)

ESQUEMA DE DATOS

```
CREATE TABLE Awards
(
    playerID string,
    awardID string,
    yearID int,
    lgID string,
    tie string,
    notes string
)
STORED AS TEXTFILE;
```

ESQUEMA DE DATOS

```
ALTER TABLE Awards
```

```
ADD COLUMNS (notes string);
```

```
DESCRIBE TABLE Awards;
```

```
SHOW CREATE TABLE Awards;
```

```
DROP TABLE Awards;
```

plain concepts

HIVE DATAWAREHOUSE



HIVE WAREHOUSE

- Los datos de las tablas Hive se almacenan en el warehouse
 - Por defecto en HDFS: /user/hive/warehouse
- Las tablas se almacenan como subdirectorios del warehouse
- Las particiones se almacenan como subdirectorios de las tablas
- Soporte a tablas externas – LOCATION (afecta LOAD y DROP)
- Los datos se almacenan en ficheros

plain concepts

HIVE WAREHOUSE



METASTORE

- Base de datos requerida por Hive
- Almacena
 - Definiciones de tabla (tipos de columna, localización en disco...)
 - Información de particionado
- Almacenada e un motor de BBDD relacional (SQL Server, MySQL...)

plain concepts

METASTORE



INGESTA DE DATOS

ESTRATEGIAS DE CARGA

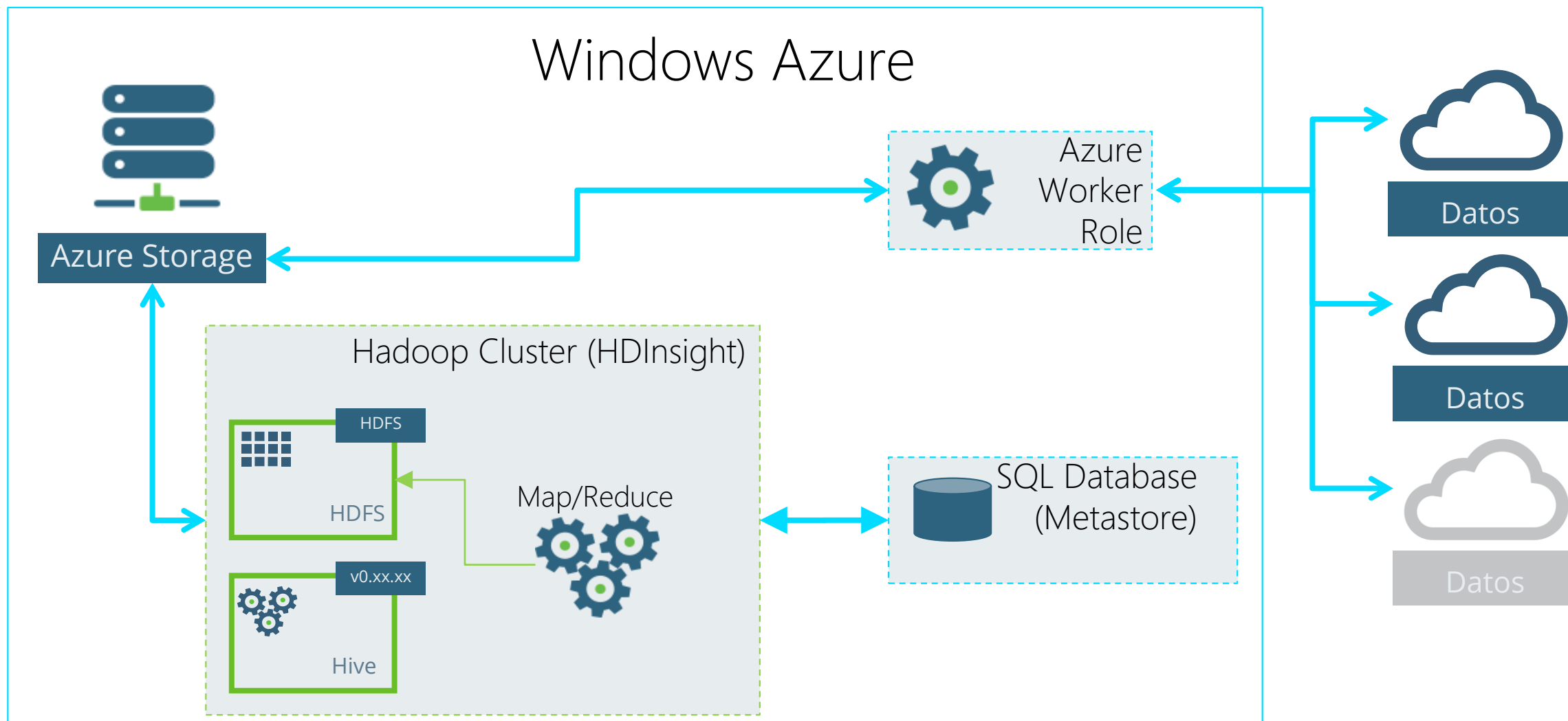
- Utilizando Ambari o Hue para gestionar los ficheros
 - Para pequeñas pruebas de concepto
- Creando un Worker Role para almacenar datos en Blob Storage (HDI)
- Utilizando la API de WebHDFS
- Utilizando Flume/Sqoop

plain concepts

CARGA CON AMBARI/HUE



INGESTA DESDE WORKER ROLE



plain concepts

WEB HDFS

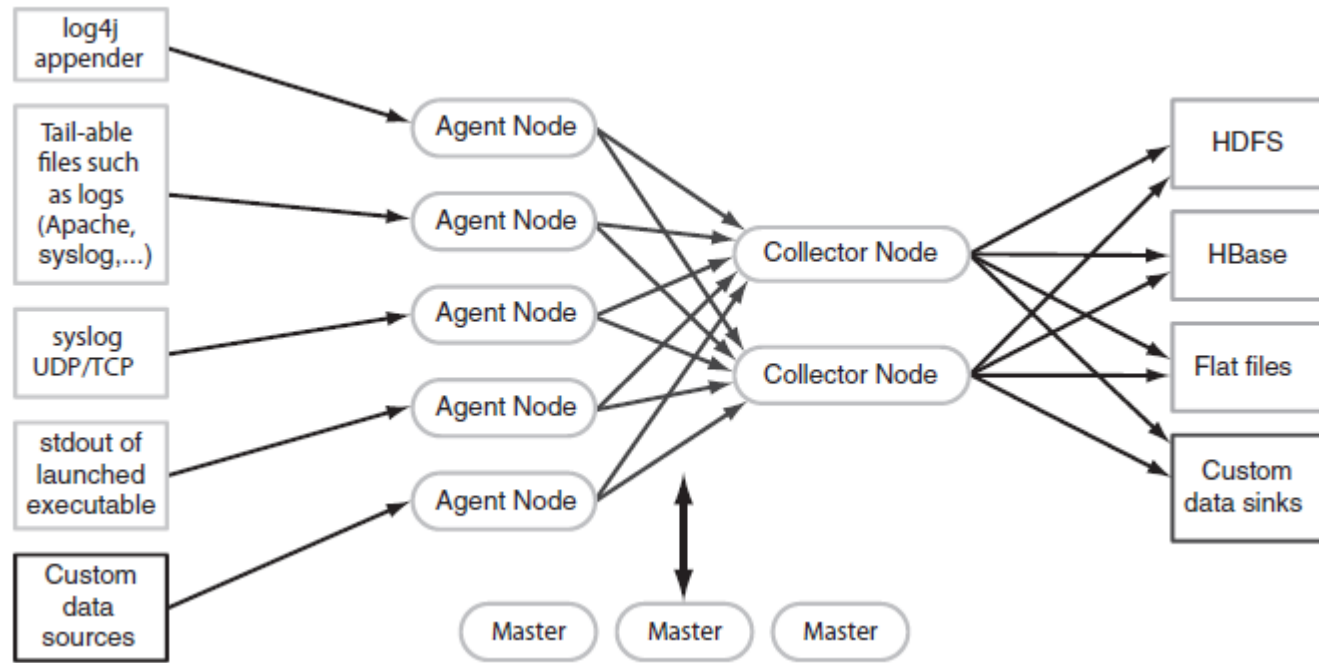


FLUME

- Apache Flume es un servicio distribuido para recolectar, agrupar y mover gran cantidad de datos streaming hacia HDFS
- Los nodos agente se instalan en las maquinas que generan los datos
- Se encargan de enviar la información a los collector nodes que la agregan y la envían al storage

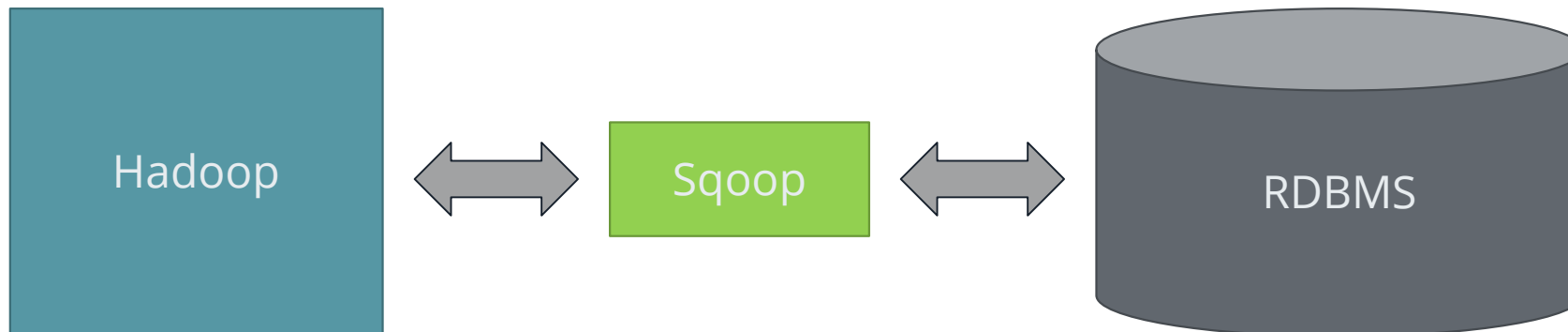


FLUME



SQOOP

- Utilidad de línea de comandos para transformar y transferir datos entre una base de datos relacional y Hadoop
- Soporta importaciones incrementales
- La función Import mueve datos de una base de datos relacional hacia Hadoop
- La función Export mueve datos desde Hadoop a una base de datos relacional



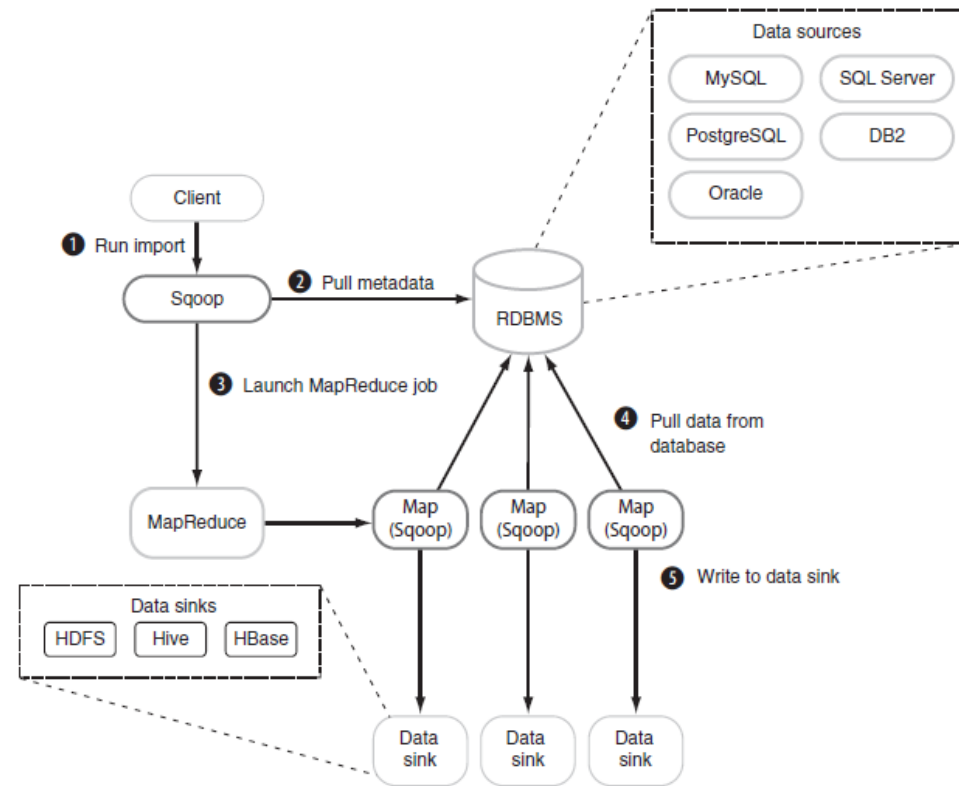
SQOOP

El dataset a transferir se divide en bloques pequeños

Sqoop lanza un job que contiene solamente mappers

Cada mapper es responsable de transferir un bloque del dataset

SQOOP



SQOOP

- http://www.cloudera.com/documentation/enterprise/latest/topics/cdh_ig_jdbc_driver_install.html

```
curl -L 'http://download.microsoft.com/download/0/2/A/02AAE597-3865-456C-AE7F-613F99F850A8/sqljdbc_4.0.2206.100_enu.tar.gz' | tar xz
```

```
sudo cp sqljdbc_4.0/enu/sqljdbc4.jar /var/lib/sqoop/
```

plain concepts

SQOOP



LOAD DATA

```
LOAD DATA INPATH '/tmp/admin/Awards.tsv' OVERWRITE INTO  
TABLE Baseball.Awards;
```

```
LOAD DATA INPATH '/tmp/admin/Players.csv' OVERWRITE INTO  
TABLE Baseball.Players;
```

TABLAS EXTERNAS

- Las tablas externas se almacenan fuera del directorio por defecto de Hive
- Requieren el modificador EXTERNAL, y especificar un directorio con LOCATION
- Al hacer DROP TABLE los datos permanecen

```
LOCATION 'hdfs://mstrainingmadrid-  
mn0.northeurope.cloudapp.azure.com:8020/user/clouderaadmin/players  
,
```


plain concepts

TABLAS EXTERNAS



SERDES Y FORMATOS

- SerDe es la abreviación de Serializador/Deserializador
- Son las interfaces que se encargan de interpretar el formato de los ficheros donde se almacenan los datos del clúster

Avro

ORC

RegEx

Thrift

Parquet

CSV

Json

Custom

SERDES Y FORMATOS

STORED AS TEXTFILE

STORED AS INPUTFORMAT

`'org.apache.hadoop.mapred.TextInputFormat'`

OUTPUTFORMAT

`'org.apache.hadoop.hive ql.io.IgnoreKeyTextOutputFormat'`

AVRO

- Framework de serialización
- Desarrollado como parte de Apache Hadoop
- Usa JSON para definir el esquema de datos
- Muy útil gracias al SDK de .NET

AVRO

```
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.avro.AvroSerDe'  
STORED AS  
INPUTFORMAT  
'org.apache.hadoop.hive.q1.io.avro.AvroContainerInputFormat'  
OUTPUTFORMAT  
'org.apache.hadoop.hive.q1.io.avro.AvroContainerOutputFormat'  
TBLPROPERTIES ('avro.schema.url'='hdfs://mstrainingmadrid-  
mn0.northeurope.cloudapp.azure.com:8020/user/clouderaadmin/avrosch  
emas/Awards.avsc');
```

plain concepts 

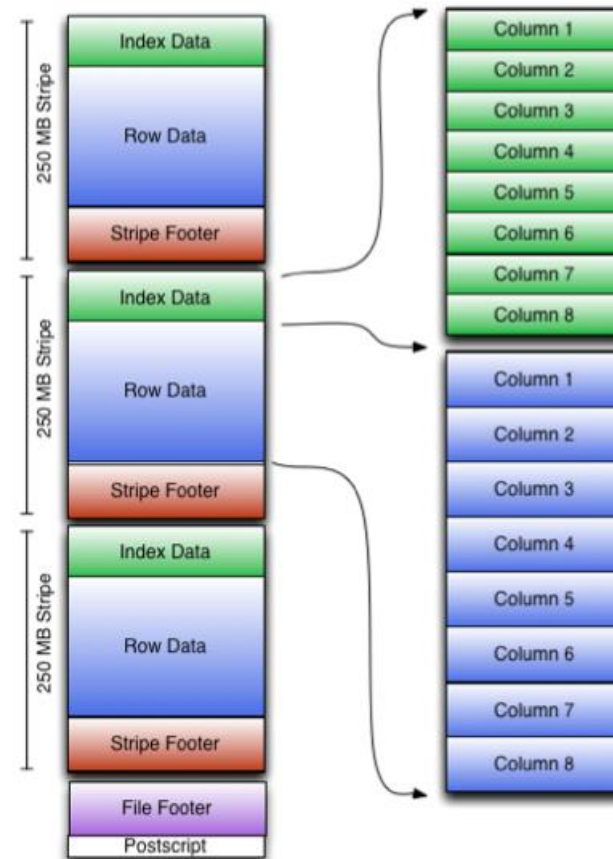
AVRO



ORC

- Optimized Row Columnar File Format
- Mejor rendimiento al leer, escribir y procesar
- Menor espacio de almacenamiento
- Almacena algunos valores utiles en el footer
 - COUNT, SUM, MAX, MIN
- Un único fichero de output
- Indexado intra-fichero

ORC



ORC

STORED AS ORC

ROW FORMAT SERDE 'org.apache.hadoop.hive.q1.io.orc.OrcSerde'

STORED AS INPUTFORMAT

'org.apache.hadoop.hive.q1.io.orc.OrcInputFormat'

OUTPUTFORMAT

'org.apache.hadoop.hive.q1.io.orc.OrcOutputFormat'

plain concepts

ORC



PARTICIONES EN HIVE

- PARTITIONED BY (BirthYear INT, BirthState STRING)
- Particiones físicas
- Optimización de consultas
- Optimización de escrituras
- Necesario especificar la partición en INSERT
- SHOW PARTITIONS

plain concepts

PARTICIONADO



LENGUAJES DE HADOOP

- Hadoop es perfecto para procesar grandes cantidades de datos pero...
 - Escribir programas en Java es lento y complejo
 - No todo el mundo sabe (o quiere) escribir código Java
- ¿La solución? Desarrollar lenguajes de proceso de datos de alto nivel
 - Hive usa HiveQL que es ANSI SQL
 - Pig usa Pig Latin que es similar a PEL

HIVEQL - ¿QUE ES?

- Hive Query Language
- Subconjunto de ANSI SQL
- Algunas limitaciones en comparaciones de igualdad y joins
- No permite updates (hasta la versión 0.14)
 - Pero si INSERT/INSERT OVERWRITE
- Nos permite escribir queries SQL y que Hive las traduzca a un job de MapReduce (o Tez)

plain concepts

HIVEQL



JOIN AVANZADO

- Si solo usamos una columna de cada tabla involucrada, solo habrá un job de MR
 - `SELECT * FROM a JOIN b ON a.key = b.key JOIN c ON b.key = c.key`
 - Un job de MR
 - `SELECT * FROM a JOIN b ON a.key = b.key JOIN c ON b.key2 = c.key`
 - Dos jobs de MR
- Si hacemos JOIN de múltiples tablas, pondremos la mayor al final
 - El reducer hará stream de la tabla mas grande, y mantendrá las otras en buffer

JOIN AVANZADO

- Evitar JOIN en WHERE
 - El optimizador no es capaz de trabajar
 - Aplicara el producto cartesiano complete antes de filtrar ☹

```
SELECT a.ymd, a.price_close, b.price_close
FROM stocks a
JOIN stocks b ON a.ymd = b.ymd
WHERE a.symbol = 'AAPL' AND
      b.symbol = 'IBM' AND
      a.ymd > '2010-01-01';
```

plain concepts

HIVEQL - JOIN



plain concepts

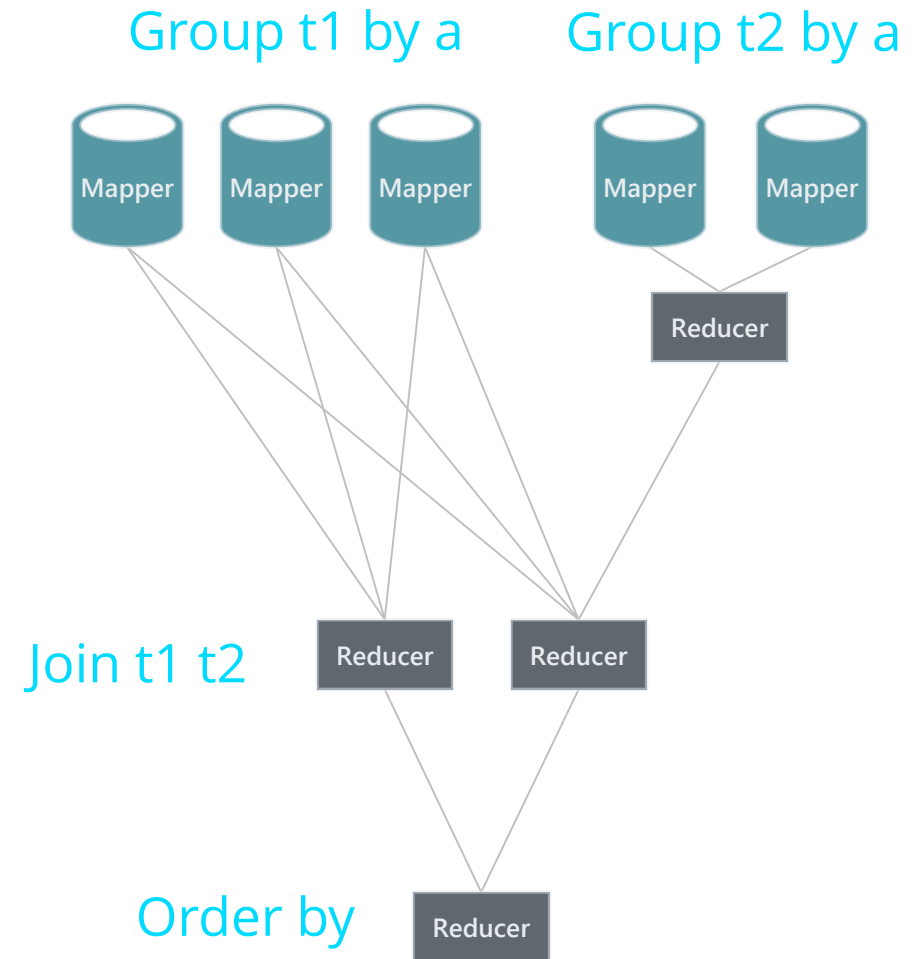
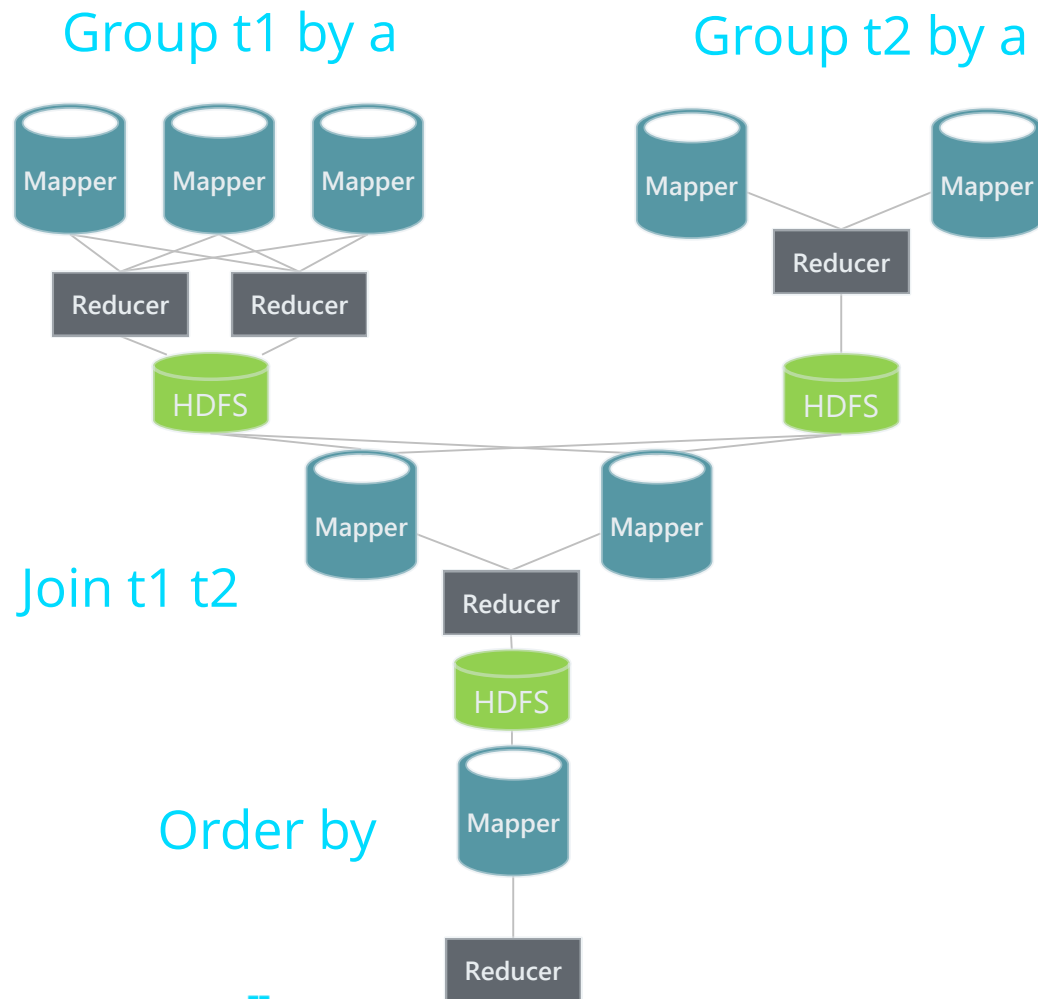
EXECUTION PLAN



TEZ

- Proyecto Apache para tener un sistema de computación distribuida de propósito general
- Generaliza el paradigma de MapReduce expresando el flujo como un DAG (Directed Acyclic Graph)
 - Elimina procesos de IO, replicaciones, lanzamiento de jobs innecesarios...
- Reemplaza la parte de procesamiento de datos de MapReduce
- Ampliamente personalizable
- Construido sobre YARN como sistema de gestión de recursos

TEZ VS MAPREDUCE



plain concepts

MR, TEZ E IMPALA



VECTORIZACION

- Require datos en format ORC
- `set hive.vectorized.execution.enabled = true;`
- Aumenta el rendimiento procesando bloques de 1024 filas
 - Dentro del bloque, cada columna es un vector (array)
 - Las operaciones simples pueden iterar sobre el vector con rapidez

CONFIGURACIONES INTERESANTES

- `set hive.execution.engine = mr; / set hive.execution.engine = tez;`
- `set tez.session.client.timeout.secs = -1;`
- `set mapreduce.input.fileinputformat.split.maxsize = 32000000;`
- `set hive.mapred.supports.subdirectories = true;`
- `set mapred.input.dir.recursive = true;`
- `set hive.stats.autogather = true;`
- `set hive.cli.print.header = true;`

Yellow Cards, Pig

PIG

PIG - ¿QUE ES?

- Apache Pig es una plataforma para analizar datos consistente en un lenguaje de alto nivel y un interprete de ese lenguaje
- Una de sus características principales es que la estructura de las consultas permite un alto nivel de paralelismo, mejorando el rendimiento

PIG LATIN

- Las consultas Pig se escriben en Pig Latin

Procedimental

Extensible

Sencillo de
programar

PIG VS SQL

Pig es procedimental

SQL es declarativo

El esquema es opcional

Requiere un esquema

Pensado para trabajos analiticos de tipo scan (sin lecturas o escrituras aleatorias)

Pensado para trabajos OLTP y OLAP

Optimización limitada

Queries faciles de optimizar

PIG VS SQL

```
Users = load 'users' as (name, age, ipaddr);
Clicks = load 'clicks' as (user, url, value);

ValuableClicks = filter Clicks by value > 0;
UserClicks = join Users by name, ValuableClicks by user;
Geoinfo = load 'geoinfo' as (ipaddr, dma);
UserGeo = join UserClicks by ipaddr, Geoinfo by ipaddr;
ByDMA = group UserGeo by dma;

ValuableClicksPerDMA = foreach ByDMA
    generate group, COUNT(UserGeo);
store ValuableClicksPerDMA into 'ValuableClicksPerDMA';
```

```
insert into ValuableClicksPerDMA
select dma, count(*) from geoinfo
join
( select name, ipaddr from users
join clicks on (users.name =
clicks.user) where value > 0; )
using ipaddr group by dma;
```

plain concepts

PIG



¿PREGUNTAS?

plain concepts

EJERCICIO PRÁCTICO





GRACIAS

Barcelona



Bilbao



Madrid



Sevilla



Dubai



London



Seattle