Tema 3: Análisis de datos masivos con Pig y Hive.

Máster Universitario en Ingeniería Informática

Extracción y Explotación de la Información

Índice

- Entornos de Hadoop: Pig
  - Programación en Pig
- Entornos de Hadoop: Hive

Entornos de Hadoop

- Hay una familia de proyectos que trabajan de forma conjunta con Hadoop
  - HBase: Almacenamiento de datos altamente escalable con estilo NoSQL.
  - Pig: Lenguaje de procesamiento de datos para tareas de Hadoop con estilo dataflow.
  - Hive: Lenguaje de procesamiento de datos con estilo SQL.
  - Mahout: Conjunto de herramientas de machine learning y data mining.

Pig y Hive

- La metodología MapReduce nos permite programar muchas aplicaciones en Hadoop.
- El problema es que, si estamos desarrollando una aplicación que incluye muchos pasos MapReduce, su programación puede ser complicada.
  - ¿Cuantas líneas de código hay para word count?
- Existen varia interfaces de programación de alto nivel que permiten programar aplicaciones paralelas sobre MapReduce, en particular Pig y Hive

Pig

- Tenemos un conjunto de datos de películas de cine. Para cada película tenemos el título, año de realización, puntuación de clasificación y duración del film

(ver http://www.rohitmenon.com/index.php/apache-pig-tutorial-part-1/)

Num	Título	Año	Rating	Tiempo (seg.)
133	Reservoir Dogs	1992	4	5963
134	Priest	1994	3.4	5862
135	Pocahontas	1995	3.8	4864
136	The Battleship Potemkin	1925	3.6	4153
137	Platoon	1986	4	7187
138	Re-Animator	1985	3.4	5146
139	Pulp Fiction	1994	4.1	9265

- Por ejemplo, supongamos que se necesita listar las películas con un rating mayor que 4.

Pig

- En la función mapper (en Python) se realiza el filtrado de las películas con rating mayor que 4

```
def mapper(record):
   if float(record[3])>4:
            mr.emit_intermediate(record[0], record[1])
```

- En la función reduce simplemente se hace el output del resultado

```
def reduce(key, list_of_values):
  mr.emit((key, list_of_values))
```

- Sería necesario programar estas funciones MapReduce para cada grupo de resultados que se desee obtener.

Pig

- Con Pig es posible facilitar el proceso de programar en MapReduce
- Pig es un lenguaje de flujo de datos
  - Más sencillo que Java
  - Simplifica el procesamiento de datos
  - Es un lenguaje de alto nivel sobre MapReduce
- Mediante Pig podemos realizar transformaciones a conjuntos de datos.
- Con respecto al ejemplo anterior:

```
movies = load ‘movies.csv’  using PigStorage(‘,’) as (id, title, year,rating, duration);
movsup4 = filter movies by rating>4;
store movsup4 into ‘movssup4’;
```

Pig

- Para ejecutar dicho script en hadoop, se carga la herramienta grunt

```
bigdata@ubuntu:~$ pig
……
grunt>
```

- También se puede lanzar el script en **modo map-reduce** mediante ‘pig script.pig’

- Si lanzamos ‘pig –x local script.pig’ el script se ejecuta en **modo local** (sin hadoop).

Pig

- Una operación **GROUP ... BY ...** se realiza de la forma siguiente:

```
movies = LOAD 'movies.csv'  using PigStorage(',') as (id, title, year,rating, duration);
movgrp =  GROUP movies by year;
movcnt = FOREACH movgrp GENERATE group, COUNT(movies) as cuenta;
STORE movcnt INTO 'group_mov';
```

- Una operación **FILTER...** se realiza de la forma siguiente:

```
movies = LOAD 'movies.csv'  using PigStorage(',') as (id, title, year,rating, duration);
movgrp =  GROUP movies by year;
movcnt = FOREACH movgrp GENERATE group, COUNT(movies) as cuenta;
movfilt = FILTER movcnt BY cuenta > 2;
STORE movfilt INTO 'filter_mov';
```

Pig

- Una operación **ORDER ... BY ...** se realiza de la forma siguiente:

```
movies = LOAD 'movies.csv'  using PigStorage(',') as (id, title, year,rating, duration);
movgrp =  GROUP movies by year;
movcnt = FOREACH movgrp GENERATE group, COUNT(movies) as cuenta;
movfilt = FILTER movcnt BY cuenta > 2;
movsrt = ORDER movfilt BY cuenta DESC;
STORE movsrt INTO 'sort_mov';
```

Pig

- La operación **JOIN ... BY ...** la podemos ver obteniendo, para cada año, la película con el menor rating:

```
movies = LOAD 'movies.csv'  using PigStorage(',') as (id, title, year,rating, duration);
movgrp =  GROUP movies by year;
movstat = FOREACH movgrp GENERATE
            group as year,
            MIN(movies.rating) as min_rating;

movmin = JOIN movies BY (year,rating), movstat BY (year,min_rating);
movmin = FOREACH movmin GENERATE movies::id as id, movies::title as title,
movies::year as year, movies::rating as rating, movies::duration as duration;

movminsort = ORDER movmin BY year;
movminfilt = FILTER movminsort BY year >= 2000;

STORE movminfilt INTO 'min_mov' using PigStorage(',');
```

Índice

- ✓ Entornos de Hadoop: Pig
- ➢ Entornos de Hadoop: Hive
  - ▫ Programación en Hive

Hive

- Hive es la infraestructura de almacenamiento de datos (warehouse) que facilita la búsqueda y gestión de grandes conjuntos de datos en un sistema de ficheros distribuido (HDFS)
  - Utiliza un lenguaje SQL-like (HiveQL – Hive query language)
  - Se convierten las sentencias HiveQL en MapReduce jobs.

- Hay algunas diferencias de Hive con las BBDD tradicionales basadas en SQL que deben ser tenidas en cuenta:
  - Hive no es adecuado para aplicaciones que necesitan tiempos de respuesta muy rápidos como se espera en las BBDD tradicionales (Hive tiene alta latencia).
  - Hive está basado en la lectura de datos y no resulta apropiado para procesamiento de transacciones que supongan un gran número de operaciones de escritura.

Hive

- Hive proporciona los siguientes aspectos:
  - Tools to enable easy data extract/transform/load (ETL)
  - A mechanism to impose structure on a variety of data formats
  - Access to files stored either directly in Apache HDFSTM or in other data storage systems such as Apache HBaseTM
  - Query execution via MapReduce

Hive

- Para ejecutar dicho script en hadoop, se carga la herramienta hive (terminal)

```
bigdata@ubuntu:~$ hive
……
hive>
```

- También se puede lanzar el script en **modo script** mediante ‘hive –f script.sql’

- Una consulta se puede lanzar como hive -e 'select * from table'

Hive

- Para crear una tabla en HIVE:

```
CREATE TABLE movstats (
min_year INT,
min_id INT,
min_title STRING,
min_rating FLOAT,
max_id INT,
max_title STRING,
max_rating FLOAT)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' STORED AS
TEXTFILE;
```

- Podemos describir la tabla creada:

```
DESCRIBE movstats;
```

Hive

- La carga de datos se realiza mediante **LOAD DATA**:

```
LOAD DATA LOCAL inpath 'movstat.txt' into table movstats;
```

- El archivo ‘movstat.txt’ se crea a partir de la salida generada anteriormente mediante pig.

- Se pueden hacer consultas mediante **SELECT**:

```
SELECT * FROM movstats LIMIT 6;
```

Hive

- Otras funciones que se pueden utilizar en Hive:

Function	Hive
Selecting a database	USE database;
Listing databases	SHOW DATABASES;
Listing tables in a database	SHOW TABLES;
Describing the format of a table	DESCRIBE table;
Creating a database	CREATE DATABASE db_name;
Dropping a database	DROP DATABASE db_name (CASCADE);

Hive

- Otras funciones que se pueden utilizar en Hive:

Function	Hive
Retrieving With Multiple Criteria	SELECT * FROM TABLE WHERE rec1 = "value1" AND rec2 = "value2";
Retrieving Unique Output	SELECT DISTINCT column_name FROM table;
Sorting	SELECT col1, col2 FROM table ORDER BY col2;
Sorting Reverse	SELECT col1, col2 FROM table ORDER BY col2 DESC;
Grouping With Counting	SELECT owner, COUNT(*) FROM table GROUP BY owner;
Maximum Value	SELECT MAX(col_name) AS label FROM table;
Selecting from multiple tables (Join same table using alias w/"AS")	SELECT pet.name, comment FROM pet JOIN event ON (pet.name = event.name)