Práctica de HADOOP

Alumno: Inti Luna Avilés

Profesor: David Garcia Escobar

2024-05-08

Indice

Table of Contents

1. Ejercicio práctico

Debido a que la startup aún no ha podido desplegar el cluster, tu objetivo es, a través de los datos contenidos de películas contenidos en un dataset público (https://github.com/dgarciaesc/sample_dataset)

1.1 Ejercicio práctico

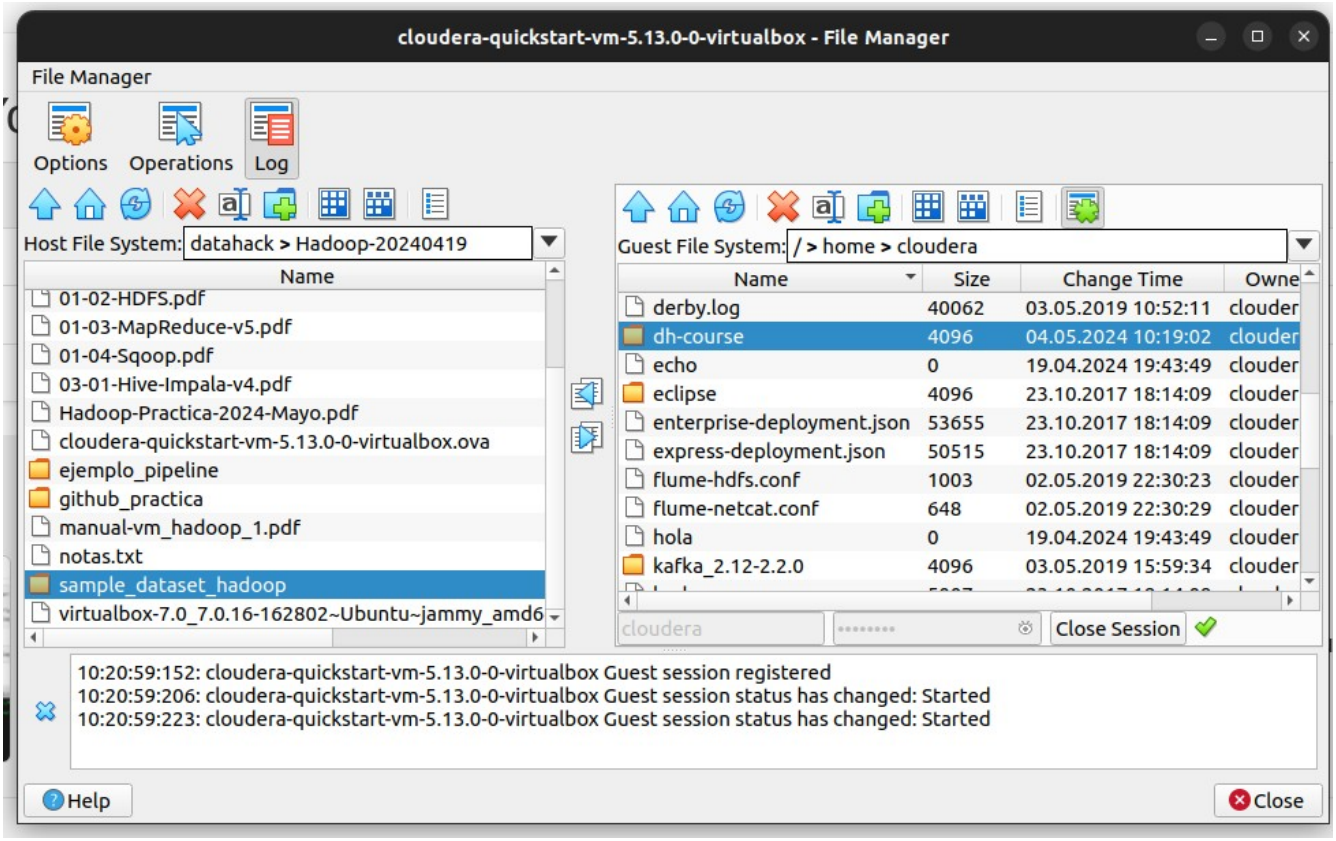
De cara a definir por qué genero apostar, identificar los influencers que pueden potenciar el marketing de MovieBuster y definir una estrategia de publicidad, el CEO te pide averiguar los siguientes datos del momento de mercado actúal:

1. Cuál es la película con más opiniones?
Qué 10 usuarios son los más activos a la hora de puntuar películas?
Cuáles son las tres mejores películas según los scores? Y las tres peores?
Hay alguna profesión en la que deberíamos enfocar nuestros esfuerzos en publicidad? Por qué?
Se te ocurre algún otro insight valioso que pudiéramos extraer de los datos procesados? Cómo?

Obtener datos y transferir a maquina virtual

Descarga en local de datos
>git clone git@github.com:dgarciaesc/sample_dataset.git
Transferencia a VM usando interface de virtual box

- Click en Machine/File Manager
- Se ingresa user y password
- Se escoge fichero entrada y salida



Se verifica en terminal dentro de VM:

```
[cloudera@quickstart ~]$ cd dh-course/
[cloudera@quickstart dh-course]$ ls
airfares.tsv        mapper.py
customers.csv       openflights_airports-parsed-sample.tsv
customers.java      param.txt
data1               python-utils.py
ex-param.pig        quijote_clean.txt
ex.pig              quijote.txt
flume-hdfs.conf     reducer.py
flume-netcat.conf   remove_punct.txt
hola_2.txt          sample_dataset_hadoop
hola.txt            test
log4j.properties    test-hive.hql
macro-ex.pig
[cloudera@quickstart dh-course]$
```

Revisamos ficheros en hdfs:
>hadoop fs -ls

```
[cloudera@quickstart dh-course]$ hadoop fs -ls
Found 8 items
drwxr-xr-x   - cloudera cloudera          0 2024-04-26 09:41 count
drwxr-xr-x   - cloudera cloudera          0 2024-04-27 01:45 customers
-rw-r--r--   1 cloudera cloudera     292231 2024-04-19 10:56 customers.csv
drwxr-xr-x   - cloudera cloudera          0 2024-04-19 11:31 data
drwxr-xr-x   - cloudera cloudera          0 2024-04-27 01:47 orders
-rw-r--r--   1 cloudera cloudera    2999944 2024-04-19 10:37 orders.csv
-rw-r--r--   1 cloudera cloudera          0 2024-04-19 10:39 prueba_inti
drwxr-xr-x   - cloudera cloudera          0 2024-04-27 01:52 test-all
[cloudera@quickstart dh-course]$
```

Creamos directorio para proyecto
>hadoop fs -mkdir proyecto

Se pasa carpeta “sample_dataset_hadoop”a hdfs con:
>hadoop fs -put sample_dataset_hadoop proyecto/sample_dataset_hadoop

Listamos elementos en proyecto:
>hadoop fs -ls proyecto

```
[cloudera@quickstart dh-course]$ hadoop fs -put sample_dataset_hadoop proyecto/sample_dataset_hadoop
[cloudera@quickstart dh-course]$ hadoop fs -ls proyecto
Found 1 items
drwxr-xr-x   - cloudera cloudera          0 2024-05-04 03:28 proyecto/sample_dataset_hadoop
[cloudera@quickstart dh-course]$ hadoop fs -ls proyecto/sample_dataset_hadoop
Found 5 items
drwxr-xr-x   - cloudera cloudera          0 2024-05-04 03:28 proyecto/sample_dataset_hadoop/.git
-rw-r--r--   1 cloudera cloudera       5577 2024-05-04 03:28 proyecto/sample_dataset_hadoop/README
-rw-r--r--   1 cloudera cloudera     171308 2024-05-04 03:28 proyecto/sample_dataset_hadoop/movies.dat
-rw-r--r--   1 cloudera cloudera   24594131 2024-05-04 03:28 proyecto/sample_dataset_hadoop/ratings.dat
-rw-r--r--   1 cloudera cloudera     134368 2024-05-04 03:28 proyecto/sample_dataset_hadoop/users.dat
[cloudera@quickstart dh-course]$
```

Verificamos en folder de user:
>hadoop fs -ls /user/cloudera/proyecto/sample_dataset_hadoop

```
[cloudera@quickstart ~]$ hadoop fs -ls /user/cloudera/proyecto/sample_dataset_hadoop
Found 5 items
drwxr-xr-x   - cloudera cloudera          0 2024-05-04 03:28 /user/cloudera/proyecto/sample_dataset_hadoop/.git
-rw-r--r--   1 cloudera cloudera       5577 2024-05-04 03:28 /user/cloudera/proyecto/sample_dataset_hadoop/README
-rw-r--r--   1 cloudera cloudera     171308 2024-05-04 03:28 /user/cloudera/proyecto/sample_dataset_hadoop/movies.dat
-rw-r--r--   1 cloudera cloudera   24594131 2024-05-04 03:28 /user/cloudera/proyecto/sample_dataset_hadoop/ratings.dat
-rw-r--r--   1 cloudera cloudera     134368 2024-05-04 03:28 /user/cloudera/proyecto/sample_dataset_hadoop/users.dat
[cloudera@quickstart ~]$
```

HIVE

Hive requiere definir las tablas y delimitador para abrilos correctamente. Para ello exploramos los datos de README file y en linea de comando con comando “head”.
> head file.dat

```
[cloudera@quickstart sample_dataset_hadoop]$ head movies.dat
1::Toy Story (1995)::Animation|Children's|Comedy
2::Jumanji (1995)::Adventure|Children's|Fantasy
3::Grumpier Old Men (1995)::Comedy|Romance
4::Waiting to Exhale (1995)::Comedy|Drama
5::Father of the Bride Part II (1995)::Comedy
6::Heat (1995)::Action|Crime|Thriller
7::Sabrina (1995)::Comedy|Romance
8::Tom and Huck (1995)::Adventure|Children's
9::Sudden Death (1995)::Action
10::GoldenEye (1995)::Action|Adventure|Thriller
[cloudera@quickstart sample_dataset_hadoop]$ head ratings.dat
1::1193::5::978300760
1::661::3::978302109
1::914::3::978301968
1::3408::4::978300275
1::2355::5::978824291
1::1197::3::978302268
1::1287::5::978302039
1::2804::5::978300719
1::594::4::978302268
1::919::4::978301368
[cloudera@quickstart sample_dataset_hadoop]$ head users.dat
1::F::1::10::48067
2::M::56::16::70072
3::M::25::15::55117
4::M::45::7::02460
5::M::25::20::55455
6::F::50::9::55117
7::M::35::1::06810
8::M::25::12::11413
9::M::25::17::61614
10::F::35::1::95370
[cloudera@quickstart sample_dataset_hadoop]$
```

Definimos las tablas en HIVE

Entramos a hive en terminal:

>hive

Definimos tabla para ratings:

Tabla para `ratings.dat`:

```
CREATE TABLE ratings (
    user_id INT,
    movie_id INT,
```

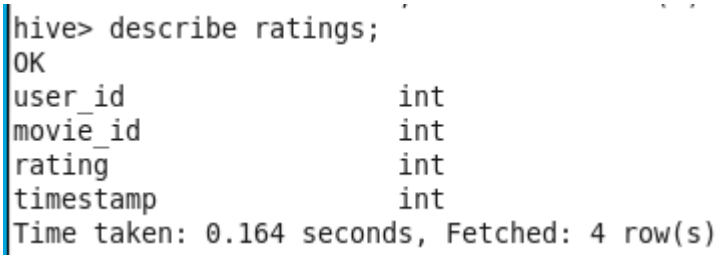
```
    rating INT,
    timestamp INT
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '::'
STORED AS TEXTFILE
LOCATION '/proyecto/sample_dataset_hadoop/ratings.dat';
```

mostramos tablas

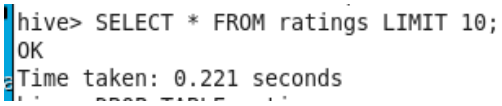
>show tables;

describimos tabla ratings

> describe ratings;



Probamos que hay datos con una query:



No hay datos, asi que verificamos. Nos damos cuenta que LOCATION no esta bien definido, asi que eliminamos tabla con > DROP TABLE ratings; y redefinimos la tabla (sin location) y luego movemos ficheros a warehouse.

```
CREATE TABLE ratings (
    user_id INT,
    movie_id INT,
    rating INT,
    timestamp INT
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '::';
```

```
hive> CREATE TABLE ratings (
    >     user_id INT,
    >     movie_id INT,
    >     rating INT,
    >     timestamp INT
    > )
    > ROW FORMAT DELIMITED
    > FIELDS TERMINATED BY '::';
OK
Time taken: 0.162 seconds
hive> dfs -ls /user/hive/warehouse;
Found 6 items
drwxrwxrwx   - cloudera supergroup          0 2024-04-27 03:30 /user/hive/warehouse/cust2
drwxrwxrwx   - cloudera supergroup          0 2019-05-02 21:46 /user/hive/warehouse/cust_part
drwxrwxrwx   - cloudera supergroup          0 2019-04-06 03:22 /user/hive/warehouse/customers
-rw-r--r--   1 cloudera supergroup    2999944 2024-04-27 03:38 /user/hive/warehouse/orderdecsv
drwxrwxrwx   - cloudera supergroup          0 2019-04-06 03:27 /user/hive/warehouse/orders
drwxrwxrwx   - cloudera supergroup          0 2024-05-04 23:07 /user/hive/warehouse/ratings
```

Se verifica data size:
-rw-r--r-- 1 cloudera cloudera 24594131 2024-05-04 03:28 /user/hive/warehouse/ratings/ratings.dat

Se verifican datos concuerden con los obtenidos en terminal con >head ratings:

Pero no concuerdan. Se intenta usar código octal en base a documentación:
“note
The `CREATE TABLE` clauses `FIELDS TERMINATED BY`, `ESCAPED BY`, and `LINES TERMINATED BY` have special rules for the string literal used for their argument, because they all require a single character. You can use a regular character surrounded by single or double quotation marks, an octal sequence such as `'\054'` (representing a comma), or an integer in the range '-127'..'128' (with quotation marks but no backslash), which is interpreted as a single-byte ASCII character. Negative values are subtracted from 256; for example, `FIELDS TERMINATED BY '-2'` sets the field delimiter to ASCII code 254, the Icelandic Thorn character used as a delimiter by some data formats. ”
Referencia: https://docs.cloudera.com/cdw-runtime/cloud/impala-sql-reference/topics/impala-create-table.html

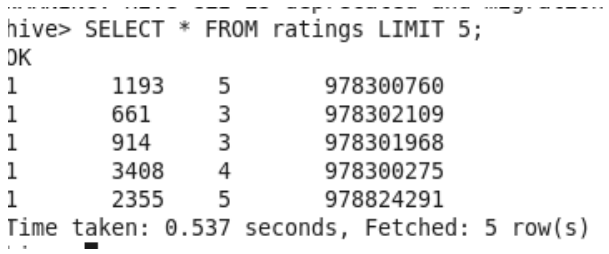
Se probo con codigo \072 que es la representacion octal de “:” pero no funciono.

>ALTER TABLE ratings SET SERDEPROPERTIES ('field.delim'='\072\072');

Referencia: https://www.ibm.com/docs/en/aix/7.2?topic=adapters-ascii-decimal-hexadecimal-octal-binary-conversion-table

Asi que para avanzar se modifica ficheros para reemplazar “::” por “,” en terminal con:
>sed -i 's/::/,/g' ratings.dat

Se actualiza ficheros y tabla. Se comprueba en hive que los datos se abren bien:



Se crean tablas restantes y se cargan datos respectivamente a warehouse:

```
# Tabla para users.dat:
CREATE TABLE users (
    user_id INT,
    gender STRING,
    age_group STRING,
    occupation STRING,
    zip_code STRING
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ',';

# Tabla para movies.dat:
CREATE TABLE movies (
    movie_id INT,
    title STRING,
    genres ARRAY<STRING>
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ',';
```

De cara a definir por qué genero apostar, identificar los influencers que pueden potenciar el marketing de MovieBuster y definir una estrategia de publicidad, el CEO te pide averiguar los siguientes datos del momento de mercado actúal:

¿Cuál es la película con más opiniones?

Consulta

```
SELECT movie_id, COUNT(*) AS total_ratings
FROM ratings
GROUP BY movie_id
ORDER BY total_ratings DESC
LIMIT 1;
```

Respuesta

movie_id: 2858
movie_title =American Beauty (1999) 3428
ratings:3428

Para obtener nombre:

```
SELECT m.title AS movie_title, COUNT(*) AS total_ratings
FROM ratings r
JOIN movies m ON r.movie_id = m.movie_id
GROUP BY m.title
ORDER BY total_ratings DESC
LIMIT 1;
```

¿Qué 10 usuarios son los más activos a la hora de puntuar películas?

Consulta:

```
SELECT user_id, COUNT(*) AS total_ratings
FROM ratings
GROUP BY user_id
ORDER BY total_ratings DESC
LIMIT 10;
```

Respuesta: user_id/total_ratings

4169 2314
1680 1850
4277 1743
1941 1595
1181 1521
889 1518
3618 1344
2063 1323
1150 1302
1015 1286

¿Cuáles son las tres mejores películas según los scores? Y las tres peores?

Consulta mejores:

```
SELECT r.movie_id, m.title, AVG(r.rating) AS avg_rating
FROM ratings r
JOIN movies m ON r.movie_id = m.movie_id
GROUP BY r.movie_id, m.title
ORDER BY avg_rating DESC
LIMIT 3;
```

Respuesta mejores:
1830 Follow the Bitch (1998) 5.0
3233 Smashing Time (1967) 5.0
3607 One Little Indian (1973) 5.0

Consulta peores:

```
SELECT r.movie_id, m.title, AVG(r.rating) AS avg_rating
FROM ratings r
JOIN movies m ON r.movie_id = m.movie_id
GROUP BY r.movie_id, m.title
ORDER BY avg_rating ASC
LIMIT 3;
```

Respuesta peores:
3460 Hillbillys in a Haunted House (1967) 1.0
2217 Elstree Calling (1930) 1.0
641 Little Indian 1.0

¿Hay alguna profesión en la que deberíamos enfocar nuestros esfuerzos en publicidad?

Consulta:

```
SELECT occupation, COUNT(user_id) AS total_users
FROM users
GROUP BY occupation
ORDER BY total_users DESC;
```

Respuesta:
Los estudiantes son los usuarios mas activos y una parte de la publicidad podria estar dirigida a los grupos mas activos (other, executive, academic y technician-engineer) para mantener y atraer más. Por otra parte se podría dirigir publicidad tambien hacia otros grupos sabiendo que sus numeros podrian crecer (farmer, trademan, customer service).

Salida de consulta:
4 759 / college grad students
0 711 / other
7 679 /executive/managerial
1 528 /academic/educator
17 502 /technician/engineer
12 388
14 302
20 281
2 267
16 241
6 236
10 195
3 173
15 144

13 142
11 129
5 112 /customer service
9 92 /homemaker
19 72 /unemployed
18 70 /tradesman/craftsman
8 17 /farmer

1. Se te ocurre algún otro insight valioso que pudiéramos extraer de los datos procesados? Cómo?

Obtener los grupos por genero y edad seria valioso. Por ejemplo, según datos los hombres entre 18 y 40 son los usuarios mas activos. Asi que una estrategia seria encontrar como atraer a mas mujeres. Entendiendo por ejemplo que genero les atrae mas e invertir mas en estos.

Consulta para obtener usuarios por genero y edad:

```
SELECT u.gender, u.age_group, COUNT(DISTINCT u.user_id) AS total_users
FROM ratings r
JOIN users u ON r.user_id = u.user_id
GROUP BY u.gender, u.age_group
ORDER BY total_users DESC;
```

Resultado: genero/grupo_etario/cantidad

M 25 1538
M 35 855
M 18 805
F 25 558
M 45 361
M 50 350
F 35 338
F 18 298
M 56 278
F 45 189
F 50 146
M 1 144
F 56 102
F 1 78

Tabla para users.dat:

```
CREATE TABLE users (
    user_id INT,
    gender STRING,
    age_group STRING,
    occupation STRING,
    zip_code STRING
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ',';
```

Tabla para movies.dat:

```
CREATE TABLE movies (
    movie_id INT,
    title STRING,
    genres ARRAY<STRING>
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ',';
```

1.2 Ejercicio práctico

El CEO está preocupado con la eficiencia de las queries usadas para extraer los datos de los ejercicios prácticos y exige poder ver estos resultados desde una web.
Implementa, a través de Sqoop, una BBDD relacional en MySQL que contenga al menos los datos de uno de los insights extraídos en el ejercicio práctico #1

Consula a usar:

```
INSERT OVERWRITE DIRECTORY '/user/cloudera/proyecto/resultado'
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
SELECT user_id, COUNT(*) AS total_ratings
FROM ratings
GROUP BY user_id
ORDER BY total_ratings DESC
LIMIT 10;
```

Paso 1. Crear directorio de resultado de consulta
>hadoop fs -mkdir /user/cloudera/proyecto/resultado
>hive

En hive

```
hive> INSERT OVERWRITE DIRECTORY '/user/cloudera/proyecto/resultado'
    > SELECT user_id, COUNT(*) AS total_ratings
    > FROM ratings
    > GROUP BY user_id
    > ORDER BY total_ratings DESC
    > LIMIT 10;
Query ID = cloudera_20240506085454_371e085a-9240-4961-9f7f-7a5883739b3e
Total jobs = 2
Launching Job 1 out of 2
Number of reduce tasks not specified. Estimated from input data size: 1
```

Se revisa y se imprime en consola:
> hadoop fs -cat /user/cloudera/proyecto/resultado/000000_0

```
[cloudera@quickstart dh-course]$ hadoop fs -cat /user/cloudera/proyecto/resultado/000000_0
4169,2314
1680,1850
4277,1743
1941,1595
1181,1521
889,1518
3618,1344
2063,1323
1150,1302
1015,1286
[cloudera@quickstart dh-course]$
```

Revisamos en mysql
>mysql -u retail_dba -p
ingresamos “cloudera”
>show databases;
con este usuario no puedo crear database con > CREATE DATABASE resultado_db; por falta de permisos

Creamos tabla resultado en mysql con:

```
CREATE TABLE resultado (
   user_id INT,
   total_ratings INT
);
```

```
mysql> CREATE TABLE resultado (
    ->     user_id INT,
    ->     total_ratings INT
    -> );
Query OK, 0 rows affected (0.02 sec)
```

Verificamos tabla resultado:

```
Database changed
mysql> show tables;
+---------------------+
| Tables_in_retail_db |
+---------------------+
| categories          |
| customers           |
| departments         |
| order_items         |
| orders              |
| products            |
| resultado           |
+---------------------+
7 rows in set (0.00 sec)

mysql> describe resultado;
+---------------+---------+------+-----+---------+-------+
| Field         | Type    | Null | Key | Default | Extra |
+---------------+---------+------+-----+---------+-------+
| user_id       | int(11) | YES  |     | NULL    |       |
| total_ratings | int(11) | YES  |     | NULL    |       |
+---------------+---------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

Salimos de mysql

#probamos pasar de HDFS a MYSQL
sqoop export --connect jdbc:mysql://localhost/retail_db \
  --username retail_dba --password cloudera \
  --table resultado \
  --export-dir /user/cloudera/proyecto/resultado \
  --input-fields-terminated-by ','

Se revisa en mysql:

```
Database changed
mysql> select * from resultado;
+---------+---------------+
| user_id | total_ratings |
+---------+---------------+
|    1150 |          1302 |
|    1015 |          1286 |
|    4169 |          2314 |
|    1680 |          1850 |
|    4277 |          1743 |
|    1941 |          1595 |
|    1181 |          1521 |
|     889 |          1518 |
|    3618 |          1344 |
|    2063 |          1323 |
+---------+---------------+
10 rows in set (0.00 sec)
```

2. Dimensionamiento clúster Hadoop

El CEO está muy contento con el trabajo realizado y quiere apostar aún más por tecnologías Big Data. Está pensando en montar otra infraestructura Hadoop que procese eventos de películas provenientes de distintas fuentes (cines, plataformas de streaming, etc..) y necesita estimación del tamaño plataforma teniendo en cuenta que:

Media eventos (por día)	tamaño por evento (KB)	tamaño (bytes)
10000	15	15360
120000	0.29	300
150000	100	102400
170000	800	819200
2000	1500	1536000

Especificaciones de maquinas	Cantidad	Unidad
Discos	22	
Capacidad x disco	2	TB
capacidad total por maquina	44	TB

Primero se pasa los datos de eventos diarios en tamaño a TB y luego se multiplica por los eventos en 1 año (365 dias) y se suman obteniendo :57.63 TB.

Ahora los datos se triplican ya que estaran en sistema HDFS (x3) y se toma un margen de 20% por seguridad de variaciones de eventos y tamaño se obtiene una demanda de 207.46 TB. Al dividir entre los 44 TB de la capacidad total de maquina se obtiene 4.7 maquinas asi que se redondea a 5 maquinas las necesarias para operar con el sistema un año.

Detalle de calculo en fichero calculo_dimensionamiento.ods

Estimación arquitecturas Hadoop para distintos casos de uso

En la empresa están también pensando en conectar su plataforma Big Data con otras herramientas de la empresa y nos piden consejo sobre cómo podría integrarse/ejecutarse:

-Herramienta de BI (p.ej.: Microstrategy)

-Web de consultas sobre pedidos realizados

-Generación de informes SQL usando R que se ejecutan mensualmente

-Recopilación de información de redes sociales

Para cada una de estas tareas indica que posibles herramientas del ecosistema Hadoop aplicarían por requisitos de casuística teniendo en cuenta las ventajas e inconvenientes de cada una de ellas (por ejemplo, uso de Impala consume mucha RAM).

Caso	Herramienta	Ventajas	Inconvenientes
-Herramienta de BI (p.ej.: Microstrategy)	HDFS,Hive y Spark	HDFS tiene prestaciones de gran almacenamiento de datos. Hive sirve para explorar y análisis usando lenguaje similar a SQL. Spark trabaja en memoria y es mucho más rápido.	Para Hive, en dependencia de datos y procesos, la latencia puede ser problema y no se recomienda para consultas interactivas de baja latencia. Spark requiere más recursos de memoria.
-Web de consultas sobre pedidos realizados	HDFS, Spark	Si los volumenes de datos son grandes HDFS puede ser una buena solución. Spark es rápido para consultas interactivas.	Requiere recursos de memoria.
-Generación de informes SQL usando R que se ejecutan mensualmente	Hive, Spark y MapReduce	Hive puede procesar todos los datos en paralelo y tiene escalabilidad horizontal. Existen herramientas de integración de R con Spark.	La latencia puede ser un problema pero al ser procesos mensuales no hay inconveniente.
-Recopilación de información de redes sociales	Flume, Spark, Hive y HBase	Flume puede recopilar grandes volumenes de datos en tiempo	Configuracion de Flume puede ser compleja.

real de diferentes fuentes hacia sistemas distribuidos de datos como HDFS y HBASE. Y con Hive o Spark se podrían analizar de ser necesario.