viu Universidad Internacional de Valencia

Actividad 2
Programación Apache Spark

Procesamiento de datos masivos

Òscar Garibo

Introducción:

El procesamiento masivo de datos no siempre se puede realizar con tecnologías tradicionales. En muchas ocasiones se tienen que utilizar tecnologías Big Data como Hadoop MapReduce y Spark.

Objetivo:

Conocer el modelo de procesamiento MapReduce y las principales herramientas Big Data.

Desarrollar programas Big Data utilizando el framework Apache Spark.

Trabajo previo:

Lectura del material docente de la parte específica que se encuentra disponible desde el comienzo del curso en la carpeta: Recursos y materiales>1. Materiales docentes:

Visualización de las videoconferencias teóricas (VC), es decir las sesiones de clases.

Metodología:

En las videoconferencias teóricas (VC) se expondrá al alumno conocimientos, material e indicaciones suficientes para que pueda elaborar una unidad didáctica basada en el aprendizaje y enseñanza por competencias en matemáticas e informática. Las actividades se centrarán en poner en práctica y asentar los conocimientos adquiridos en las videoconferencias teóricas relacionadas.

Actividades a elaborar:

Desarrollo de tres programas Big Data utilizando el framework Spark. Se podrá utilizar el lenguaje Python o Java.

1. (6 ptos) Dado un dataset que contenga entradas con la forma "persona;método_pago;dinero_gastado", crea un programa llamado personaGastosSinTarjetaCredito que para cada persona indique la suma del dinero gastado con cualquier forma de pago exceptuando tarjeta de crédito, con el formato persona;gastosinTDC. Ejemplo:

Entrada

```
Alice;Tarjeta de crédito;100
Alice;Tarjeta de crédito;150
Alice;Bizum;200
Luis;300
Bob;Tarjeta de crédito;201
Luis;Bizum;300
```

Salida

```
Alice;200
Bob;0
```

Notar que Alice gasta en total 450 euros, pero sólo 200 son con medios distintos a tarjeta de crédito (Bizum).
Se valorará positivamente la eficiencia del programa, por ejemplo no usar transformaciones innecesarias.

2. (7 ptos) Dado un dataset que contiene información sobre los videos de Youtube (https://netsg.cs.sfu.ca/youtubedata/), crear un programa llamado CategoriaDeVideosMasVista que obtenga cuál es la categoría de videos más vista de la plataforma Youtube y el número total de visualizaciones que hay en esa categoría. El programa debe recibir dos parámetros de entrada:
la carpeta en la que está el dataset y la carpeta en la que se guardará el resultado. En la carpeta donde está el dataset se tienen que descomprimir UNO de los archivos 0222.zip, 0301.zip, etc., que se encuentran en el enlace anterior. Importante: si la persona que hace la actividad dispone de pocos recursos computaciones, entonces se recomienda que únicamente descomprima algún .zip pequeño para que pueda desarrollar el programa. La carpeta de datos de entrada debería quedar como se ve en la Figura 1.

Los datos de entrada están en los archivos 0.txt, 1.txt, etc y cada fila contiene la información de un video tabulada con el siguiente formato: id del video de youtube, usuario que subió el video, número de días desde que se subió el video y la fecha en la que obtuvieron los datos, categoría del video, longitud del video, número de visitas del video, puntuación del video, número de puntuaciones del video, número de comentarios del video, y una lista de ids de videos relacionados.

Se valorará positivamente la eficiencia del programa, por ejemplo no usar transformaciones innecesarias.

Ejemplo:

Entrada	Salida
... Gadgets & Games ... 30 ... Gadgets & Games ... 10 ... Music ... 90 ... Sports ... 20 ... Music ... 50 ... Gadgets & Games ... 95	Music;140

Notar que la categoría “Sports” es la que menos visitas tiene: 20 en un único vídeo. “Music” es la que más visitas tiene: 90 en un video + 50 en otro video, es decir, en total 140 visitas, y la categoría “Gadgets & Games” tiene en total 135 visitas obtenidas de 30 + 10 + 95.

El programa debe funcionar independientemente del número de categorías, para cualquier cantidad de filas que se pueda llegar a tener, para cualquier cantidad de ficheros e ignorar el log.txt.

3. (7 ptos) Dado un dataset que contenga entradas con la forma “persona;método_pago;dinero_gastado”, crea un programa llamado personaYMetodosDePago que:

a) Por cada persona indique en cuántas compras pagó más de 1500 euros con tarjeta de crédito. La solución se tiene que guardar en una carpeta llamada comprasConTDCMayorDe1500.
b) Por cada persona indique en cuántas compras pagó menos o igual a 1500 euros con tarjeta de crédito. La solución se tiene que guardar en una carpeta llamada comprasConTDCMenoroIgualDe1500.

Se valorará positivamente la eficiencia del programa, por ejemplo no usar transformaciones innecesarias.

Ejemplo:

Entrada

```
Alice;Tarjeta de crédito;1000
Alice;Tarjeta de crédito;1800
Alice;Tarjeta de crédito;2100
Bob;Bizum;2000
Alice;Bizum;1000
Bob;Tarjeta de crédito;1100
```

Salida (a)

```
Alice;1
Bob;1
```

Salida (b)

```
Alice;2
Bob;0
```

Notar que Alice realiza dos compras con tarjeta de crédito mayores a 1500 y una menor a 1500; mientras que Bob solo hace una compra con tarjeta de crédito menos a 1500.

Sobre la entrega:

- La tarea se entregará en algún formato comprimido (gzip, zip, etc.)
- Esta actividad debe realizarse individualmente.
- **Cada carpeta en el fichero comprimido debe contener documentos en PDF (con la explicación de la solución y los *print screen* de las ejecuciones), códigos en los lenguajes de programación solicitados, instrucciones para la compilación y ejecución de dichos códigos**

estándares y los ficheros de entrada y resultados obtenidos, de acuerdo al caso de cada ejercicio.

El fichero comprimido debe tener el siguiente formato:

AG_2-03MBID-Apellido-Nombre.gz

- Esta actividad tiene un peso de 20%