viu Universidad Internacional de Valencia

Actividad 1
Programación en Hadoop

Procesamiento de datos masivos

Y. Cardinale

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 Hadoop MapReduce.

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 “VC Procesamiento Big Data”, “AGHadoop MapReduce”, y “VC Hadoop MapReduce avanzado” que se encontrarán disponibles en: Videoconferencias>grabaciones

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. En la videoconferencia de actividad guiada (AG) se establecerá las pautas concretas y la dinámica que el alumnado deberá seguir para realizar la actividad propuesta. Las actividades se centrarán en poner en práctica y asentar los conocimientos adquiridos en la videoconferencia teórica anterior.

Actividades a elaborar:

Desarrollo de dos programas Big Data utilizando el framework Hadoop MapReduce. Se podrá utilizar el lenguaje Python o Java.

1. Dado un dataset que contenga entradas con la forma “persona;producto;cantidad;esDevolucion”, crea un programa llamado dineroGastadoPorClientes que indique para cada cliente cuánto dinero gastó en total. Se valorará positivamente la optimización del programa, por ejemplo a través de la funcionalidad Combiner. Ejemplo:

Entrada	Salida
`Alice;Camiseta;10;Falso`	`Alice;20`
`Alice;Camiseta;10;Cierto`	`Bob;12`
`Alice;Pantalones;20;Falso`	
`Bob;Camiseta;5;Falso`	
`Bob;Camiseta;7;Falso`	

Notar que Alicia gastó en total 20 ya que hizo dos compras gastando 30 euros (10+20), pero luego hizo una devolución de 10 euros.

2. Para este ejercicio utiliarán el fichero de entrada cite75_99.txt que puede ser descargado del National Bureau of Economic Research (NBER) de EEUU (http://www.nber.org/patents/).
Una descripción detallada de este fichero puede encontrarse en:
Hall, B. H., A. B. Jaffe, and M. Trajtenberg (2001). "The NBER Patent Citation Data File: Lessons, Insights and Methodological Tools." NBER Working Paper 8498.

Este fichero contiene citas de patentes emitidas entre 1975 y 1990 en los EEUU. Es un fichero CSV (*comma-separated values*) con más de 16,5 millones de filas, y las primeras líneas son como sigue:

```
"CITING","CITED"
3858241,956203
3858241,1324234
3858241,3398406
3858241,3557384
3858241,3634889
3858242,1515701
3858242,3319261
3858242,3668705
.........
```

La primera línea contiene una cabecera con la descripción de las columnas. Cada una de las otras líneas indica una cita que la patente con el número de la primera columna ha hecho a la patente con el número en la segunda. Por ejemplo, la segunda fila indica que la patente nº 3858241 ("citing" o *citante*) hace una cita a la patente nº 956203 ("cited" o citada).

El fichero está ordenado por las patentes citantes. Así podemos ver que la patente nº 3858241 cita a otras 5 patentes.

Deben implementar un programa MapReduce escrito en Java o Python (a elegir) que, para cada patente de cite75_99.txt, obtenga la lista de las que la citan:

- Posible implementación:
  - El mapper obtiene cada línea del fichero de entrada, separar los campos y los invierte (para obtener como clave intermedia la patente citada y como valor intermedio la patente que la cita), por ejemplo:
    3858245, 3755824 → 3755824 3858245
  - El reducer, para cada patente recibe como valor una lista de las que la citan, ordena esa lista numéricamente y la convierte en un string de números separados por coma
    3755824 {3858245 3858247. . . } → 3755824 3858245, 3858247...

- Formato de salida salida: *patente patente1, patente2...* (la separación entre la clave y los valores debe ser un **tabulado**)
  - La salida debe de estar **ordenada** por la clave (patentes citadas).
  - Los valores se deben guardar separados por coma, sin espacios en blanco entre ellos.
  - Deben tener en cuenta la cabecera, para que no aparezca en la salida (el fichero de entrada no debe modificarse de ninguna manera)

Sobre la entrega:

- La tarea se entregará en algún formato comprimido (gzip, zip, etc.)
- Esta actividad puede realizarse en grupo de dos personas (preferible) o individual.
- Cada estudiante/grupo deberá explicar su solución y demostrarla al profesor (no necesariamente en horas de clase)
- El fichero comprimido debe contener tanto documentos en PDF (con la explicación de la solución y los print screen de las ejecuciones) como códigos en los lenguajes de programación solicitados, así como instrucciones para la compilación y ejecución de dichos códigos estándares y tendrá el siguiente formato:
  AG_X-02MBIG-Apellido-Nombre.gz
- Esta actividad tiene un peso de 20%