TRABAJO CBD
Análisis de datos con MapReduce

Alumnos: Javier Meliá Sevilla,
Ismael Mira Hernandez

Desarrollo del Dataset

Un dataset es un conjunto de datos estructurados que se utilizan para analizar y comprender un fenómeno o proceso determinado. En este caso, el dataset se refiere al detalle del alumnado matriculado en las Escuelas Oficiales de Idiomas, y está estructurado por niveles de enseñanza.

El dataset que se describe en este caso es una herramienta que permite analizar y entender la situación del alumnado matriculado en las Escuelas Oficiales de Idiomas de la Comunidad Valenciana. Este conjunto de datos presenta información sobre diferentes aspectos relacionados con estas escuelas, como su ubicación geográfica, el idioma que se estudia en cada centro y los niveles de enseñanza que se imparten.

En concreto, el dataset se centra en un mismo año escolar y recoge información de diferentes escuelas de idiomas en toda la comunidad Valenciana, cada una identificada por un código que la identifica. Además, se distingue la separación entre las diferentes provincias de la comunidad, Alicante, Castellón y Valencia, con su correspondiente código postal, y los diferentes municipios dentro de cada provincia donde se encuentra cada escuela.

En relación al idioma, se pueden encontrar diferentes datos en este conjunto de datos, como los diferentes idiomas que se estudia en cada centro y el nivel que se trata, desde el más básico A1 hasta el más avanzado C2, marcando también los niveles desde básico hasta avanzado. Además, el dataset también especifica en qué curso se encuentra cada clase de cada idioma, junto con el número de grupo asignado y el total de alumnos que tiene cada clase.

Este tipo de información es muy útil para entender la demanda de enseñanza de idiomas en la Comunidad Valenciana, así como para planificar la oferta formativa de estas escuelas y mejorar la calidad de la enseñanza de idiomas en general. Por ejemplo, los datos sobre el número de alumnos matriculados en cada idioma y nivel pueden ser útiles para determinar la demanda de cada uno y adaptar la oferta educativa en consecuencia. Además, la información sobre la ubicación geográfica de las escuelas puede ser útil para determinar la accesibilidad de estas escuelas para diferentes poblaciones y planificar su ubicación estratégica en función de las necesidades de los estudiantes.

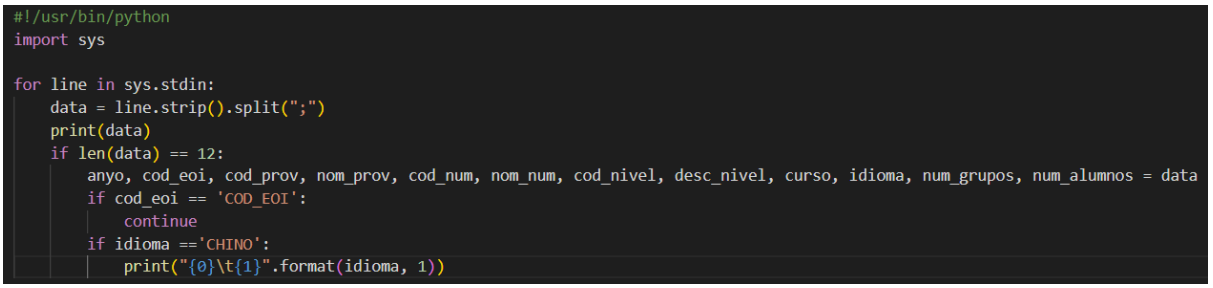
Metodología

Para realizar las consultas sobre este dataset, vamos a utilizar el paradigma de programación MapReduce y Hadoop Streaming, como hicimos en la práctica realizada previamente en la asignatura. Ya vimos que la etapa de mapping procesa los datos particionados y se los pasa a una segunda etapa (ordenados) para obtener los resultados finales. El clúster Hadoop nos permite trabajar con datasets grandes, al estar desplegado en la nube.

Consulta 1: Total de grupos que estudian Chino en toda la comunidad

Esta consulta es para encontrar, entre todas las escuelas oficiales de idiomas de toda la comunidad, cuántos grupos totales estudian chino. Para ello se han programado a partir de lo estudiado en clase un mapper y un reducer de la siguiente forma:

En el mapper podemos observar, en primer lugar, la separación de datos en 12 campos por punto y coma. Una vez tenemos los datos, quitamos la primera línea saliendo del bucle si el código es el del encabezado. Y posteriormente, tenemos un condicional que si el idioma es chino hace un print con un 1.



```
#!/usr/bin/python
import sys

for line in sys.stdin:
    data = line.strip().split(";")
    print(data)
    if len(data) == 12:
        anyo, cod_eoi, cod_prov, nom_prov, cod_num, nom_num, cod_nivel, desc_nivel, curso, idioma, num_grupos, num_alumnos = data
        if cod_eoi == 'COD_EOI':
            continue
        if idioma =='CHINO':
            print("{0}\t{1}".format(idioma, 1))
```

En el reducer queremos sumar para obtener el número de líneas que han encontrado que su idioma es chino. Por lo tanto, utilizamos el reducer de las prácticas en el que tenemos un sumatorio de enteros donde vamos almacenando en una variable el total.

```python
#!/usr/bin/python

import sys

countTotal = 0
oldKey = None

for line in sys.stdin:
    data_mapped = line.strip().split("\t")
    if len(data_mapped) != 2:
        continue

    thisKey, thisCount = data_mapped

    if oldKey and oldKey != thisKey:
        print countTotal
        oldKey = thisKey
        countTotal = 0

    oldKey = thisKey
    countTotal += int(thisCount)

if oldKey != None:
    print countTotal
```

Luego, para la comprobación de los códigos, se probó primero en local y después en Hadoop y se obtuvieron los mismos resultados como se puede observar en las siguientes imágenes:

En local lo que hicimos fue mostrar los resultados con un cat:

```
alucloud74@hadoopmaster:~/trabajo$ cat /home/alucloud74/eoii.txt | ./mapper.py | sort -k 1,1 | ./reducer.py
CHINO    20
```

Mientras que en Hadoop:

```
alucloud74@hadoopmaster:~/trabajo$ hs mapper.py reducer.py eoi/eoii.txt purchases/outpu
2023-05-09 10:03:35,852 WARN streaming.StreamJob: -file option is deprecated, please use generic option -files instead.
packageJobJar: [mapper.py, reducer.py] [] /tmp/streamjob8666167076786211700.jar tmpDir=null
2023-05-09 10:03:36,934 INFO impl.MetricsConfig: Loaded properties from hadoop-metrics2.properties
2023-05-09 10:03:37,019 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s).
2023-05-09 10:03:37,019 INFO impl.MetricsSystemImpl: JobTracker metrics system started
```

```
alucloud74@hadoopmaster:~/trabajo$ hadoop fs -text purchases/outpu/part-00000
CHINO    20
```

Podemos observar que el número de cursos de chino en la comunidad es de 20.

Consulta 2: Número de personas que estudian un idioma en Valencia

En la siguiente consulta vamos a obtener el número total de alumnos de cualquier idioma en Valencia capital, ordenados alfabéticamente según el lenguaje.

Para el mapper, es similar al anterior pero en este caso en vez de enviar un 1 que se añade al sumador, mandamos el número de alumnos junto al idioma, que será la clave.

```
#!/usr/bin/python
import sys

for line in sys.stdin:
    data = line.strip().split(";")
    print(data)
    if len(data) == 12:
        anyo, cod_eoi, cod_prov, nom_prov, cod_mun, nom_mun, cod_nivel, desc_nivel, curso, idioma, num_grupos, num_alumnos = data
        if cod_eoi == 'COD_EOI':
            continue
        if nom_mun == 'VALENCIA':
            print("{0}\t{1}".format(idioma, num_alumnos))
```

En el reducer no es necesario realizar ninguna modificación, ya que seguimos queriendo acumular datos en una variable.

```
#!/usr/bin/python

import sys

countTotal = 0
oldKey = None

for line in sys.stdin:
    data_mapped = line.strip().split("\t")
    if len(data_mapped) != 2:
        continue

    thisKey, thisCount = data_mapped

    if oldKey and oldKey != thisKey:
        print countTotal
        oldKey = thisKey
        countTotal = 0

    oldKey = thisKey
    countTotal += int(thisCount)

if oldKey != None:
    print countTotal
```

Ahora que hemos comprobado en la consulta anterior que se puede ejecutar tanto en local como en Hadoop, vamos a probar las siguientes consultas en el clúster.

```
alucloud74@hadoopmaster:~/trabajo$ hadoop fs -text purchases/output3/part-00000
ALEMÁN    794
CHINO    166
ESPAÑOL COMO LENGUA EXTRANJERA    589
EUSKERA        66
FINÉS    30
FRANCÉS        1505
GRIEGO    69
INGLÉS    5699
ITALIANO        770
JAPONÉS        205
NEERLANDÉS        49
POLACO    30
PORTUGUÉS        201
RUSO    141
VALENCIANO        1139
ÁRABE    189
```

En este caso, el resultado es una lista ordenada que nos indica el par idioma-alumnos para un lugar en concreto, lo que puede ser muy útil.

Consulta 3: Número de grupos cuyo número de alumnos está entre 10 y 50 personas

Para la siguiente columna, queremos obtener el número de grupos por idioma de tamaño mediano (lo hemos considerado entre 10 y 50 personas).

En el mapper tenemos la condición de que el número de alumnos esté entre 10 y 50, y por cada vez que la condición se cumpla añadimos un 1 al idioma que toque.

```python
#!/usr/bin/python
import sys

for line in sys.stdin:
    data = line.strip().split(";")
    print(data)
    if len(data) == 12:
        anyo, cod_eoi, cod_prov, nom_prov, cod_mun, nom_mun, cod_nivel, desc_nivel, curso, idioma, num_grupos, num_alumnos = data
        if cod_eoi == 'COD_EOI':
            continue
        if int(num_alumnos) >= 10 and int(num_alumnos) <= 50:
            print("{0}\t{1}".format(idioma, 1))
```

Por lo tanto el reducer va a ser similar al de la consulta 1, donde tengamos una variable entera en la que ir almacenando todos los resultados encontrados.

```python
#!/usr/bin/python

import sys

countTotal = 0
oldKey = None

for line in sys.stdin:
    data_mapped = line.strip().split("\t")
    if len(data_mapped) != 2:
        continue

    thisKey, thisCount = data_mapped

    if oldKey and oldKey != thisKey:
        print countTotal
        oldKey = thisKey
        countTotal = 0

    oldKey = thisKey
    countTotal += int(thisCount)

if oldKey != None:
    print countTotal
```

Echando un vistazo a los resultados, encontramos que para prácticamente todos los idiomas existen grupos medianos. El idioma para el que hay más es el francés, mientras que hay algunos con solo uno como el finés o polaco.

```
alucloud74@hadoopmaster:~/trabajo$ hadoop fs -text purchases/output5/part-00000
ALEMÁN    76
CHINO    9
ESPAÑOL COMO LENGUA EXTRANJERA    27
EUSKERA        4
FINÉS    1
FRANCÉS        110
GRIEGO    3
INGLÉS    74
ITALIANO        61
JAPONÉS        11
NEERLANDÉS        1
POLACO    1
PORTUGUÉS        8
RUSO    11
VALENCIANO        88
ÁRABE    9
```

Consulta 4: Curso más grande en toda la Comunitat Valenciana

En esta consulta queremos conocer dónde está el grupo más grande y cuántos alumnos pertenecen al mismo. Por lo tanto, la consulta es un poco diferente a las anteriores ya que se trata de un problema de máximos.

En el mapper vamos a leer todas las líneas sin ninguna condición, ya que queremos recorrer el archivo completo para obtener la información de todos los cursos.

```python
#!/usr/bin/python
import sys

for line in sys.stdin:
    data = line.strip().split(";")
    print(data)
    if len(data) == 12:
        anyo, cod_eoi, cod_prov, nom_prov, cod_num, nom_num, cod_nivel, desc_nivel, curso, idioma, num_grupos, num_alumnos = data
        if cod_eoi == 'COD_EOI':
            continue
        print("{0}\t{1}".format(nom_num, num_alumnos))
```

Y por lo tanto en el reducer tenemos que ir leyendo la información recibida y reemplazar una variable temporal con el máximo conocido hasta el momento, con un algoritmo típico de máximos.

```python
#!/usr/bin/python

import sys

maximo = 0
maxPath = ''

for line in sys.stdin:
    data_mapped = line.strip().split("\t")
    if len(data_mapped) != 2:
        # Something has gone wrong. Skip this line.
        continue

    thisKey, thisCount = data_mapped

    if int(thisCount) > maximo:
        maxPath = thisKey
        maximo = int(thisCount)

print(maxPath, "\t", maximo)
```

El resultado es que el grupo más grande se encuentra en Castellón, y tiene 727 alumnos. Lo hemos ejecutado en local ya que para esta consulta nos decía que no había recursos suficientes en Hadoop.

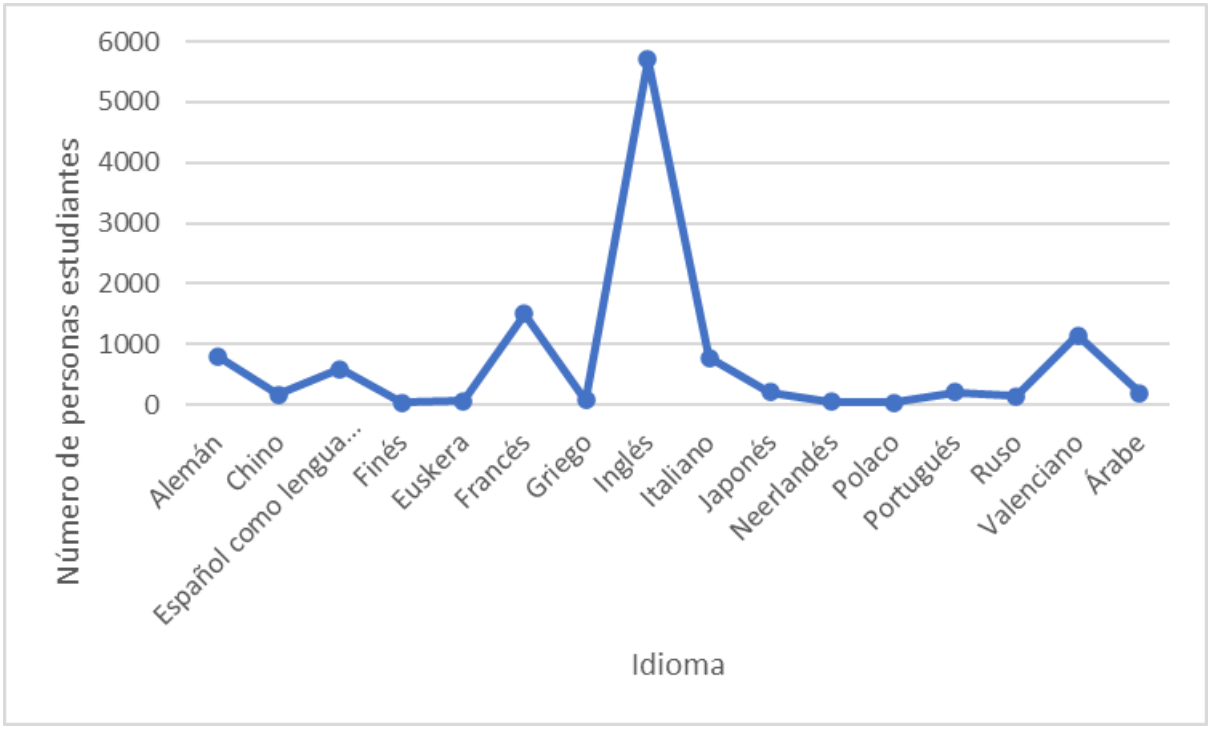
```
alucloud74@hadoopmaster:~/trabajo$ cat /home/alucloud74/eoii.txt | ./mapper.py | sort -k 1,1 | ./reducer.py
CASTELLÓ DE LA PLANA     727
```

Visualización

En este apartado solo se van a visualizar los resultados obtenidos de las consultas 2 y 3, ya que la 1 y 4 son simplemente un resultado. Entonces quedarían de la siguiente manera:

Consulta 2:

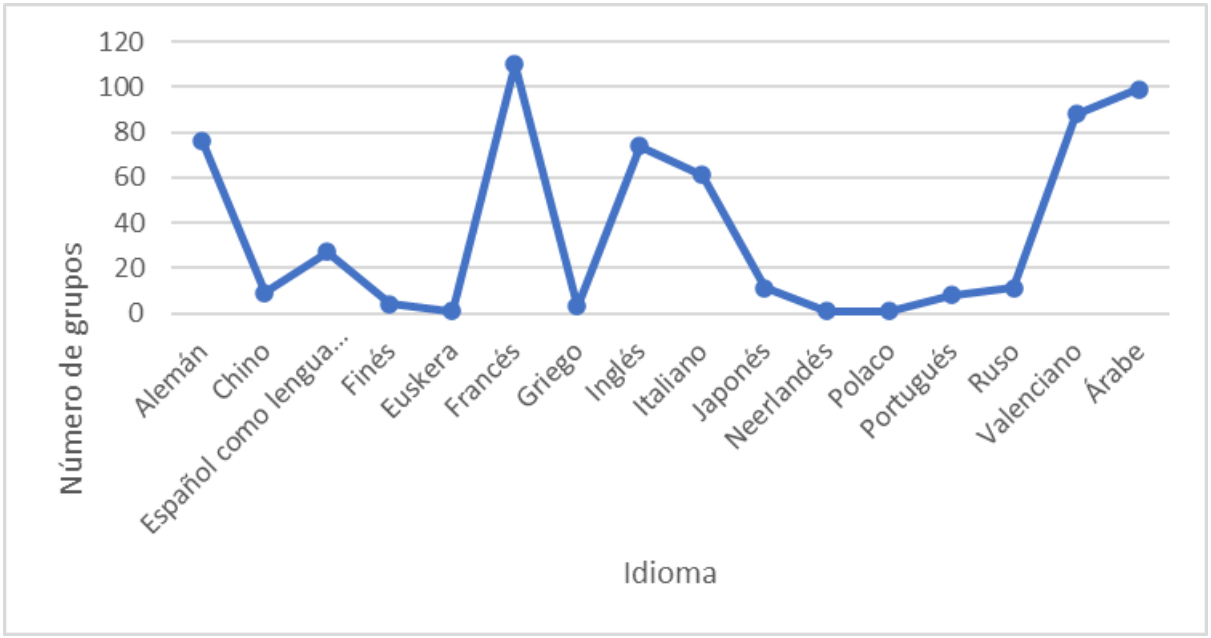
País	Número de personas estudiantes
Alemán	794
Chino	166
Español como lengua extranjera	589
Finés	30
Euskera	66
Francés	1505
Griego	69
Inglés	5699
Italiano	770
Japonés	205
Neerlandés	49
Polaco	30
Portugués	201
Ruso	141
Valenciano	1139
Árabe	189



En Valencia, se observa una tendencia hacia el estudio de las lenguas más utilizadas en la Comunidad, tales como el inglés, valenciano y francés. Es notable que estas lenguas se encuentran geográficamente más próximas a sus países de origen. Por otro lado, se evidencia una menor demanda en el estudio de lenguas del norte de Europa, como el polaco, finés y neerlandés.

Consulta 3:

País	Número de grupos
Alemán	76
Chino	9
Español como lengua extranjera	27
Finés	4
Euskera	1
Francés	110
Griego	3
Inglés	74
Italiano	61
Japonés	11
Neerlandés	1
Polaco	1
Portugués	8
Ruso	11
Valenciano	88
Árabe	99



Tras analizar los resultados, se puede observar que la mayoría de los idiomas cuentan con grupos de tamaño mediano. El francés es el que tiene la mayor cantidad de grupos, mientras que otros idiomas como el finés o polaco sólo tienen uno.

Conclusiones

El uso de MapReduce y Apache Hadoop para realizar consultas sobre un dataset ofrece una solución escalable y eficiente para el procesamiento de grandes volúmenes de datos. Para nuestros trabajo, quizás no es especialmente útil ya que el dataset no es muy grande, pero en entornos académicos y científicos donde el volumen de datos sea enorme puede ser una gran herramienta, ya que como se puede escalar horizontalmente, se pueden agregar más nodos de procesamiento para acelerar el proceso de procesamiento de datos.

En nuestro proyecto, hemos conseguido obtener estadísticas sobre el alumnado de las escuelas de idiomas de forma muy rápida. Tardamos un poco más de tiempo en realizar los mapper y los reducer, pero como teníamos ejemplos de las actividades de prácticas no tuvimos que empezar de cero. Posteriormente, modificando los existentes puedes realizar prácticamente cualquier consulta.

Tras ejecutar estos trabajos, con los resultados se pueden crear métodos de visualización de los mismos muy útiles en algunos casos, como por ejemplo para presentar los datos a un público general poco especializado.

En resumen, el uso de MapReduce y Apache Hadoop para realizar consultas ofrece una solución escalable para procesar conjuntos de datos, y aunque esto puede estar dirigido en principio a organizaciones grandes con enormes volúmenes de información, le puede ser útil a cualquiera que necesite realizar consultas.