MÁSTER

INFRAESTRUCTURAS COMPUTACIONALES PARA EL PROCESAMIENTO DE DATOS MASIVOS

PRÁCTICA DEL MÓDULO 1: ECOSISTEMA DE PROCESAMIENTO PARALELO PARA DATOS MASIVOS: APACHE HADOOP

Versión 1.0

Dr. Agustín C. Caminero Herráez — Dr. Rafael Pastor Vargas
Dr. Salvador Ros Muñoz

MÁSTER UNIVERSITARIO EN INGENIERÍA Y CIENCIA DE DATOS

Contenido

Introducción 2

Ejercicio 1: MapReduce. 3

Ejercicio 1.1: Contador de clientes valorados por países. 4

Ejercicio 1.2: País con mejores clientes. 5

Ejercicio 1.3: Mejorando el país con mejores clientes. 6

Ejercicio 2: Hive 7

Ejercicio 3: Pig 10

Introducción

En este documento se presenta el Trabajo Práctico (TP) del módulo 1 de la asignatura "INFRAESTRUCTURAS COMPUTACIONALES PARA EL PROCESAMIENTO DE DATOS MASIVOS", del "MÁSTER UNIVERSITARIO EN INGENIERÍA Y CIENCIA DE DATOS" de la UNED.

Este trabajo se realiza de forma individual. En las siguientes secciones se exponen los diferentes ejercicios que es necesario implementar para este trabajo.

Se proponen tres ejercicios diferentes. El primer ejercicio consiste en la implementación de un trabajo MapReduce con la librería MRJob. El segundo ejercicio consiste en la implementación de un trabajo utilizando Hive y el tercer ejercicio consiste en utilizar la herramienta Pig.

ES NECESARIO REALIZAR EL EJERCICIO 1 DE FORMA OBLIGATORIA, Y UN EJERCICIO MÁS A ELEGIR ENTRE LOS EJERCICIOS 2 Y 3.

El ejercicio 1 se valorará con un máximo de 8 puntos sobre 10, mientras que los ejercicios 2 y 3 se valorarán con un máximo de 2 puntos sobre 10.

La forma de evaluar el trabajo se hará en base a lo siguiente:

- **Jupyter notebook** con el código realizado. Se debe incluir no solamente el código sino también las explicaciones necesarias, imágenes, ... de forma que el notebook sea autocontenido. Al principio del notebook se debe indicar el nombre del/la estudiante. Los notebooks tienen que estar ejecutados (deben mostrar la salida de la ejecución del código realizado), y se deben incluir las órdenes necesarias para demostrar el correcto funcionamiento de los desarrollos. Por ejemplo, para demostrar que una tabla se ha creado y cargado de datos correctamente, habría que ejecutar un select que devuelva algunos de los contenidos de la tabla. Los notebooks no deben contener órdenes que ralenticen su carga de forma innecesaria, por ejemplo no se deben utilizar consultas select * from tabla para tablas con más de una decena de filas, habría que utilizar la cláusula limit.

- De forma optativa, se puede incluir una **memoria explicativa**.

Se valorará positivamente se incluya un apartado final donde se explique la opinión del/la estudiante sobre este trabajo, puntos fuertes y/o débiles, recomendaciones para el futuro, así como una valoración general de este módulo.

Este material se deberá incluir en un fichero comprimido y enviado a través del curso virtual dentro de los plazos establecidos para su entrega. El nombre de dicho fichero comprimido deberá tener la estructura *MR-ApellidosNombre.zip*, donde *Apellidos* y *Nombre* deben sustituirse por los valores correspondientes para el/la estudiante que realiza el envío. Un nombre correcto es, por ejemplo, el siguiente:

- MR-CamineroHerraezAgustin.zip

A continuación se detallan los ejercicios a completar.

Ejercicio 1: MapReduce.

Este ejercicio consta de 3 partes, que se detallan en este apartado. Para cada uno de las partes es necesario presentar lo siguiente:

- El diseño del programa MapReduce. Este diseño debe contener al menos respuestas a las siguientes preguntas:
  - ¿Cuántos pasos MapReduce son necesarios?
  - ¿Qué hace cada función de cada paso? (No es necesario código ejecutable, una descripción en texto o en pseudocódigo es suficiente).
  - ¿Qué datos se pasan de una función a la siguiente?
- Implementación de este diseño. Este código debe ejecutarse en el entorno de desarrollo propuesto para el tema de MapReduce.

El detalle de cada apartado se encuentra a continuación.

Ejercicio 1.1: Contador de clientes valorados por países.

Partiendo de los ficheros de datos de países y clientes y del código visto en el ejemplo `mrjob-join`, mejora dicho código para que el programa devuelva cuántos clientes con valoración "bueno" hay en cada país. En concreto, la salida del programa MapReduce debe ser un fichero con el contenido que se muestra en la Figura 1 . Este ejercicio tiene una puntuación de hasta 3 puntos sobre 10.

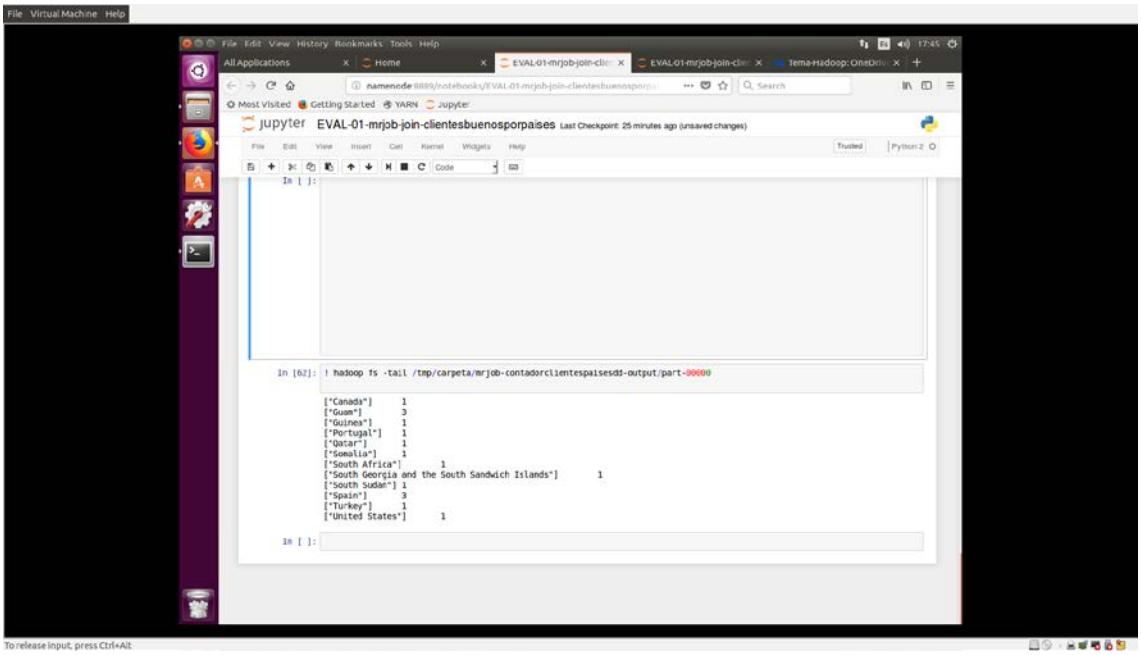

Figura 1. Fichero de salida del contador de buenos clientes.

Ejercicio 1.2: País con mejores clientes.

Partiendo del código implementado en el ejercicio anterior, extiéndelo para que devuelva el país en el que hay más clientes valorados como "bueno". En el caso de que haya más de un país con el mismo número de clientes buenos empatados en el primer lugar, se devolverá solamente uno de ellos. El resultado de este ejercicio se muestra en la Figura 2. Este ejercicio tiene una puntuación de hasta 3 puntos sobre 10.

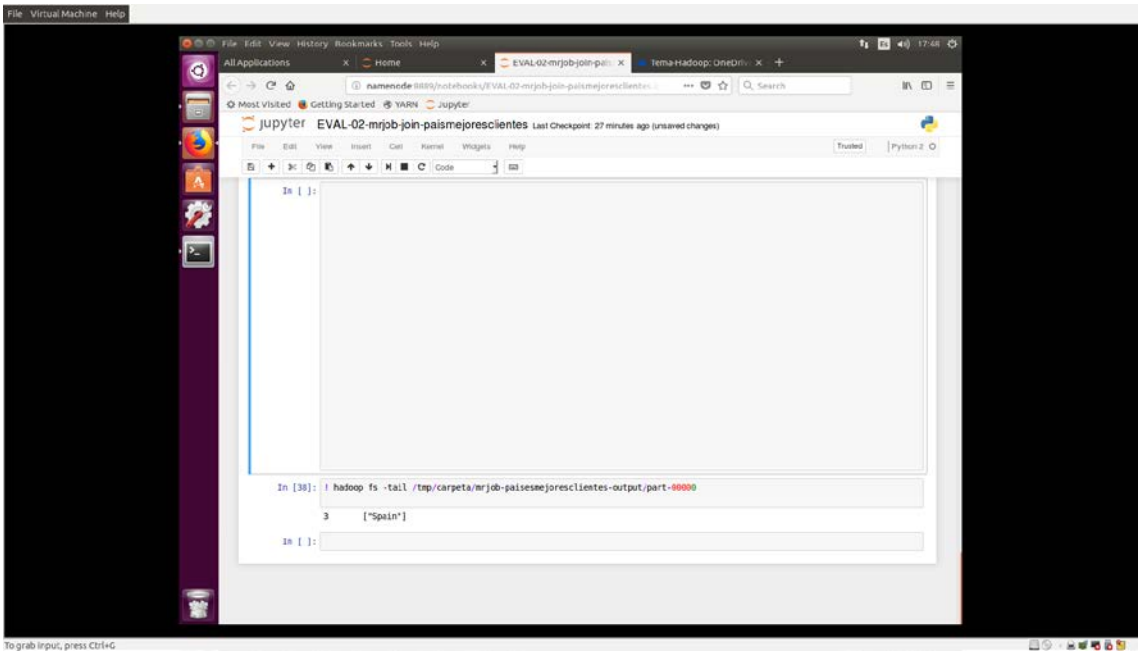

Figura 2. Fichero de salida con los mejores clientes.

Ejercicio 1.3: Mejorando el país con mejores clientes.

Partiendo del código implementado para el ejercicio anterior, mejóralo para que, en el caso de que haya más de un país empatado con el mayor número de buenos clientes, se devuelvan todos esos países. Utilizando los ficheros de datos sobre países y clientes vistos anteriormente, la salida de este programa MapReduce debería ser la que se muestra en la Figura 3. Hay que tener en cuenta que la entrada del reducer es una clave y un *generator* que contiene los valores que comparten la misma clave. Este ejercicio tiene una puntuación de hasta 4 puntos sobre 10.

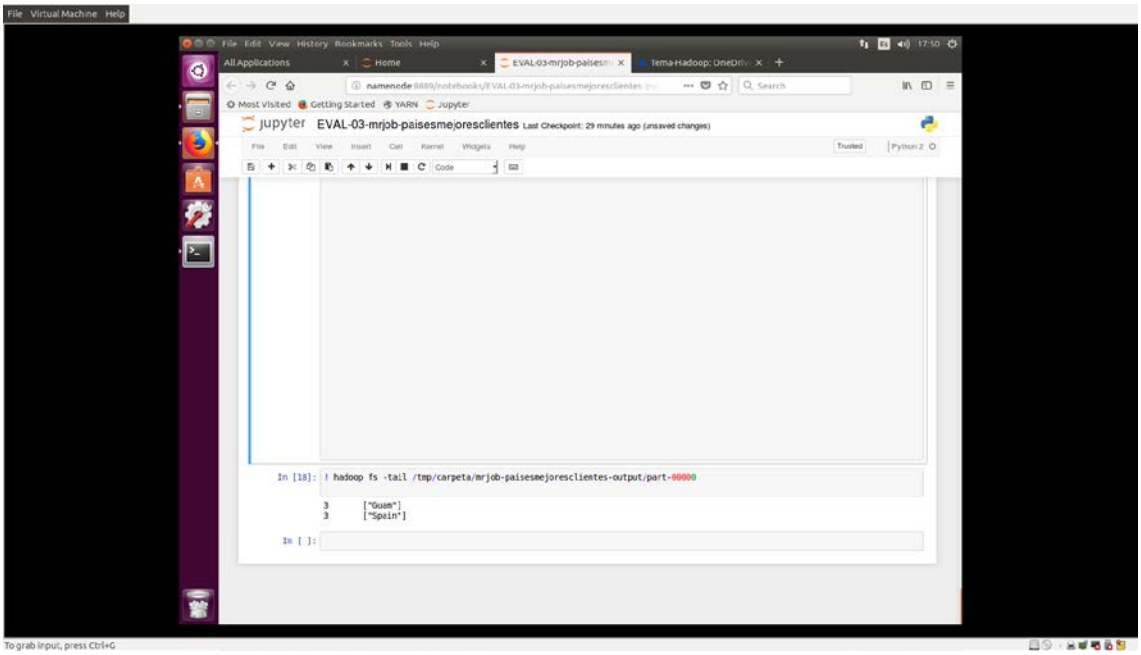

Figura 3. Fichero de salida con los mejores clientes, mejorado.

Ejercicio 2: Hive

En este trabajo se van a utilizar ficheros (**disponibles en el curso virtual**) descargados de la web del Banco Mundial que contienen información sobre inversión en investigación y desarrollo (I+D), y sobre esperanza de vida en el momento del nacimiento. En cada archivo se diferencia por años desde 1960 y por país. Son los archivos siguientes:

- API_GB.XPD.RSDV.GD.ZS_DS2_en_csv_v2_4353310.csv
- API_SP.DYN.LE00.IN_DS2_en_csv_v2_4330149.csv

Cada uno de los ficheros de arriba tiene los campos siguientes:

- Country Name: Nombre del país.
- Country Code: Código identificador del país.
- Indicator Name: Nombre del indicador que se muestra en el fichero.
- Indicator Code: Código identificador del indicador.

Abajo se muestra un fragmento del fichero API_SP.DYN.LE00.IN_DS2_en_csv_v2_4330149.csv, el otro tiene una estructura similar.

Country Name	Country Code	Indicator Name	Indicator Code	1960	1961
Aruba	ABW	Life expectancy at birth, total (years)	SP.DYN.LE00.IN	65.662	66.074

En los siguientes enlaces se puede encontrar información sobre estos datos:

1. Datos del Banco Mundial sobre esperanza de vida en el momento del nacimiento. Disponible en https://data.worldbank.org/indicator/SP.DYN.LE00.IN?view=chart
2. Datos del Banco Mundial sobre inversión en investigación y desarrollo. Disponible en https://data.worldbank.org/indicator/GB.XPD.RSDV.GD.ZS?view=chart

Partiendo de estos datos, desarrolla las órdenes de HiveQL que implementen las siguientes tareas.

1. (1 puntos) Crea las tablas necesarias para almacenar datos. Pueden ser internas o externas en función de los datos que se desee. La decisión de interna o externa debe estar razonada.

2. (1 puntos) Importa los datos en las tablas creadas.

3. (1 puntos) Crea una vista sobre las tablas creadas. Esta vista tendrá para cada país, su nombre, código, esperanza de vida en 2018 y su inversión en investigación y desarrollo en 2018. Esta vista se deberá utilizar en las consultas siguientes cuando sea necesario.

4. (4 puntos) Crea las consultas de Hive necesarias para responder las siguientes cuestiones. Nota: El/La estudiante debe decidir cómo tratar los datos inexistentes:
    - ¿Cuál es la esperanza de vida de España en 1960?
    - ¿Cuál es la esperanza de vida del país que tiene menor inversión en investigación y desarrollo en 2010? Nota: Solamente tener en cuenta los países cuyo dato está presente en el archivo.
    - ¿Cuáles son los cinco países con mayor inversión en investigación en 2010?
    - ¿Cuál es la esperanza de vida del país que tiene la mayor inversión media en I+D durante la década de 2000?

5. (3 puntos) Selecciona un dataset, impórtalo en las tablas de Hive necesarias e implementa al menos dos consultas sobre dicho dataset. Para realizar este ejercicio, es necesario seguir los siguientes pasos:
    - Selecciona una fuente de datos públicamente disponible en Internet. Puede ser un dataset o alguna otra fuente de datos como por ejemplo una red social. Si es una fuente de datos, deberás utilizar alguna de las herramientas de inyección de datos y serdes vistos en la asignatura para capturar datos para realizar este ejercicio. Si se decide utilizar un dataset, en los siguientes enlaces se pueden encontrar algunos de libre acceso (existen más repositorios de datasets):
        - https://www.kaggle.com/datasets
        - https://data.worldbank.org/
        - https://datasetsearch.research.google.com/
        - https://console.cloud.google.com/marketplace/browse?filter=solution-type:dataset&_ga=2.67976598.645382453.1601640482-195064274.1601640482

    - Una vez seleccionada la fuente de datos, el primer paso será publicar un mensaje en el foro del módulo indicando el enlace a dicha fuente de datos. Además, revisa si hay otro/a estudiante que haya publicado previamente dicha fuente de datos, en cuyo caso deberás elegir otra distinta. El objetivo es evitar que dos o más estudiantes utilicen los mismos datos en sus trabajos. En el caso de más de un estudiante haya utilizado los mismos datos, solamente se valorará el que lo haya escrito el en foro en primer lugar.

- Utilizando la fuente de datos seleccionada, importa los datos en Hive e implementa al menos tres consultas sobre dichos datos.

Este apartado se evaluará de la siguiente forma:

  - (1.5 puntos) Selección de dataset, creación de las tablas e importación de los datos.
  - (1.5 puntos) Creación de consultas.

Se valorará positivamente la complejidad de los datos utilizados así como la de las consultas implementadas. Se deberá proporcionar una breve explicación del dataset utilizado. En el caso de que no sea necesario utilizar herramientas de inyección de datos y serdes, se podrá conseguir la máxima puntuación en este apartado si las consultas tienen complejidad suficiente.

Ejercicio 3: Pig

Partiendo del notebook pig-indice-invertido-estudiantes.ipynb que se proporciona, implementa un índice invertido sobre el dataset de los fórum posts utilizado en el curso virtual. Este índice debe contener para cada palabra, un listado de los identificadores de los posts en los que aparece, así como un contador que indique en cuántos posts aparece.

En detalle, los pasos a seguir serían los siguientes:

1. (1 punto) Carga el fichero de los posts forum_node.tsv, recuerda que está separado por tabuladores.
2. (2 puntos) Limpia el fichero: elimina del body caracteres que no sean letras o números, pásalo a minúsculas, confirma que el identificador es numérico, entre otras opciones. En el identificador de post, elimina caracteres que no sean numéricos.
3. (2 puntos) Separa el body en palabras y júntalas con el identificador.
4. (2 puntos) Elimina duplicados (palabras que aparecen más de una vez en un post).
5. (1 puntos) Agrupa las palabras iguales.
6. (1 puntos) Prepara los resultados. Para cada palabra, hay que mostrar en orden ascendente el identificador de los posts donde aparece y la cuenta de los posts.
7. (1 punto) Almacena los resultados en HDFS.

Una parte de la salida de este programa es el siguiente:

```
(7001187)},13)
(zyrcster's,{(9247)},1)
(zyrcter,{(11610)},1)
(zytrax,{(6028725),(7002663)},2)
(zyx,{(8004310)},1)
(zz,{(1007745),(10011348)},2)
(zzz,{(8385)},1)
(zzzz,{(14790),(30278)},2)
(zzzzz,{(1007093),(5006080)},2)
(zzzzzzzzzzzzzzz,{(8353)},1)
```

Es interesante saber lo siguiente:

- La siguiente función comprueba si un dato X es numero: org.apache.pig.piggybank.evaluation.IsNumeric(X). Recuerda que para utilizar una función de Piggybank es necesario registrar la librería.
- Para limpiar el body se puede utilizar esta expresión regular: '[^a-zA-Z0-9\']+'