Práctica evaluable de MapReduce

# Grupo

Alumno: Adolfo González Blázquez

Email: adolfo.gonzalez@elcorteingles.es

Teléfono: +34 609 964 414

Alumno: José María Álvarez Fernández

Email: josemaria.alvarezfernandez@elcorteingles.es

Teléfono: +34 696 667 194

Alumno: César Colado Rodríguez

email: cesar.colado@elcorteingles.es

Teléfono: +34 661 415 555

# Enunciado

Partiendo del notebook *pig-indice-invertido-estudiantes.ipynb* que se proporciona, implementa un índice invertido sobre el dataset de los fórum posts utilizado en sesiones anteriores. 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. Carga el fichero de los posts forum\_node.tsv, recuerda que está separado por tabuladores.
2. 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. Separa el body en palabras y júntalas con el identificador.
4. Elimina duplicados (palabras que aparecen más de una vez en un post).
5. Agrupa las palabras iguales.
6. 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. 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)

# Propuesta

Para la ejecución de esta práctica, se ha seguido la siguiente metodología:

* Crear un cluster Hadoop usando Docker, tal y como se describió en clase
* Usar un Notebook de Jupyter para elaborar el ejercicio, que se ha ejecutado dentro del cluster Hadoop descrito anteriormente. Los pasos dados han sido:
  + Establecer el entorno de trabajo, creando directorios y copiando los datasets pertinentes al entorno Hadoop.
  + Crear el fichero con el código Pig
  + Ejecutar el código Pig en el entorno local y dentro del cluster Hadoop.

## Crear cluster Hadoop

Para crear y ejecutar el cluster Hadoop, se han usado las imágenes Docker presentadas en clase. En concreto, se han usado las siguientes:

* accaminero/namenode01
* swapnillinux/cloudera-hadoop-yarnmaster
* swapnillinux/cloudera-hadoop-datanod

Para facilitar la ejecución, se han creado una serie de scripts (localizados dentro de la carpeta “docker”) que se pueden usar para construir el cluster, arrancarlo y lanzar Jupyter.

* docker/build.sh – Construye los contenedores necesarios para crear el cluster Hadoop. El script compartirá la carpeta de la práctica con el Docker en el directorio /media/notebooks
* docker/start.sh – Si el cluster estuviese parado (una vez creado), este script lo inicia.
* docker/stop.sh – Para la ejecución del cluster Hadoop.
* docker/jupyter.sh – Inicia el entorno de ejecución de Jupyter dentro del cluster Hadoop. Para acceder a Jupyter – <http://localhost:8889>

Para ejecutar el Notebook, dentro de la carpeta de la práctica:

* docker/build.sh
* docker/jupyter.sh
* <http://localhost:8889/pig-indiceinvertido-estudiantes.ipynb>

## Código Pig

A continuación, se describe el código Pig creado para ejecutar el algoritmo de índice invertido propuesto en el ejercicio. Dicho código se puede encontrar en el fichero **students-inverted-index.pig**

1. Carga el fichero de los posts forum\_node.tsv, recuerda que está separado por tabuladores. Para cargar el fichero, usamos una extensión de *piggybank* que lee desde un fichero CSV. En concreto, los comandos relevantes son:
   1. Registrar *PiggyBank*  
      REGISTER /usr/lib/pig/piggybank.jar;
   2. Cargar el fichero usando CSVExcelStorage, separando los campos por tabulador:  
      load 'forum\_node.tsv' using org.apache.pig.piggybank.storage.CSVExcelStorage('\t', 'YES\_MULTILINE', 'NOCHANGE', 'SKIP\_INPUT\_HEADER')
2. 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.
   1. Para cada línea de datos cargada en el paso anterior, ejecutamos las siguientes sustituciones:  
      cleandata = foreach data generate  
      REPLACE(pid, '[a-zA-Z]+', '') as post\_id, LOWER(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(body, '\\\\n\\\\r', ''), '\\\\r', ''), '\\\\n', ''), '<\*>', ''), '[^a-zA-Z0-9\'\\s]+', ' ')) AS clean\_body;
3. Separa el body en palabras y júntalas con el identificador.
4. Elimina duplicados (palabras que aparecen más de una vez en un post).
5. Agrupa las palabras iguales.
6. 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. Almacena los resultados en HDFS.