TALLER MAP-REDUCE

1.- En el menú de BigData vamos a Dataproc y creamos un clúster con los valores por defecto (1 nodo maestro y 2 workers), crearlo como regional y en una zona USA.

2.- una vez creado el clúster vamos a Compute y a VM Instances, ahí veremos 3 máquinas cuyo nombre se inicia con el de nuestro clúster, el nombre del nodo master **termina en "...m",** nos conectamos a él mediabte SSH

3.- Modificamos el fichero .bashrc con el editor nano y añadimos las siguientes líneas al final del fichero.

run_mapreduce() { hadoop jar /usr/lib/hadoop-mapreduce/hadoop-streaming.jar -mapper $1 -reducer $2 -file $1 -file $2 -input $3 -output $4 }

alias hs=run_mapreduce

4.- Hacemos

$ source .bashrc para actualizar

5.- Subir al master (en la esquina superior derecha de la consola tenemos la opción para subir ficheros) los ficheros ***mapper1.py, reducer1.py, mapper2.py, reducer2.py, comedies.txt y purchases.txt*** que están en el campus virtual. Podemos ver el código de los ficheros python y hacernos una idea de lo que hacen. Podemos utilizar el comando more o cat

$ cat mapper1.py

$ cat reducer1.py

6.- instalar en el master la utilidad dos2unix

$ sudo apt-get install dos2unix

7.- cambiar los permisos de los ficheros python

$ chmod 0777 *.py

8.- Crear el directorio de entrada en HDFS, desde donde se leerán los ficheros a procesar. <vuestro_nombre_usuario> es el nombre con el que aparecéis en el directorio /home/..

$ hadoop fs -mkdir /user/<vuestro_nombre_usuario>

$ hadoop fs -mkdir /user/<vuestro_nombre_usuario>/input

9.- copiar el fichero comedies.txt y el purchases.txt a hdfs

$hadoop fs -put comedies.txt /user/vuestro_nombre_usuario/input

$hadoop fs -put comedies.txt /user/vuestro_nombre_usuario/input

10.- Convertir los picheros python a formato Unix

$ dos2unix *.py

11.- Ejecutar el comando de mapreduce

$ hs {mapper script} {reducer script} {input_directory} {output_directory}

Es decir para contar las palabras del fichero ***comedies.txt*** sería

$ hs mapper.py reducer1.py input/comedies.txt output

Observamos la salida que produce la ejecución

12.- Vemos que exista una salida en el directorio output que debe haberse creado

$hadoop fs -ls /user/vuestro_nombre_usuario/output

13.- Si hay ficheros en este directorio vemos lo que contienen, por ejemplo

$hadoop fs -cat /user/vuestro_nombre_usuario/output/PART-00000

14 .- Borramos el directorio output de hdfs

$hadoop fs -rm -r /user/vuestro_nombre_usuario/output

15.- Repetimos desde el paso 11 pero utilizando mapper2.py y reducer2.py y el fichero de entrada **purchases.txt**. Observamos lo que hacen mapper2.py y reducer2.py , así como lo que contiene el fichero purchases.txt con el comando **"cat"** y los resultados obtenidos.

$cat purchases.txt