Universidad de Granada

Master Profesional en Ingeniería Informática

Práctica 2

Hadoop

Autor:
Manuel Jesús García Manday
(nickter@correo.ugr.es)

Master en Ingeniería Informática

19 de mayo de 2017

Índice

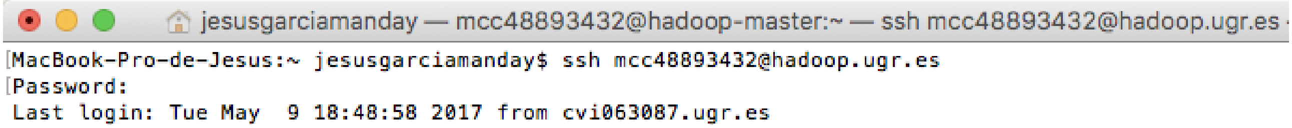
1. Objetivo.

El objetivo de esta práctica es realizar programas escalables para mejorar la eficiencia en entornos Big Data.

2. Introduccíon.

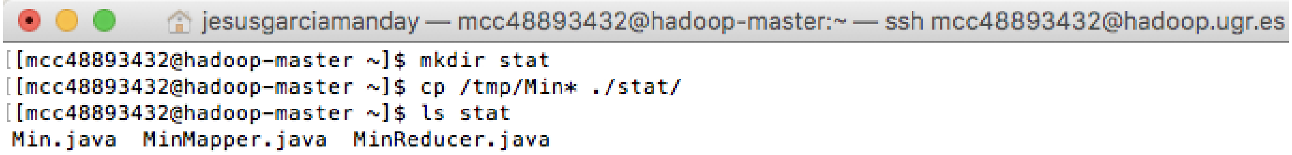
Para comenzar a realizar las tareas que se piden en esta práctica, es necesario en primer lugar realizar una serie de pasos iniciales que se describen a continuación.

Realizamos una conexión remota hacía el servidor **haddop.ugr.es** y una vez dentro creamos una carpeta nueva donde descargaremos el código Java de los programas. Comprobamos tambien que los datos de entrada se encuentran disponibles.


jesusgarciamanday — mcc48893432@hadoop-master:~ — ssh mcc48893432@hadoop.ugr.es

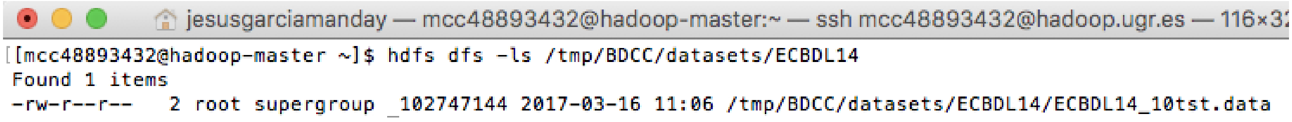
```
[MacBook-Pro-de-Jesus:~ jesusgarciamanday$ ssh mcc48893432@hadoop.ugr.es
[Password:
 Last login: Tue May  9 18:48:58 2017 from cvi063087.ugr.es
```

Figura 1: Conexion remota a **hadoop.ugr.es**.


jesusgarciamanday — mcc48893432@hadoop-master:~ — ssh mcc48893432@hadoop.ugr.es

```
[[mcc48893432@hadoop-master ~]$ mkdir stat
[[mcc48893432@hadoop-master ~]$ cp /tmp/Min* ./stat/
[[mcc48893432@hadoop-master ~]$ ls stat
 Min.java  MinMapper.java  MinReducer.java
```

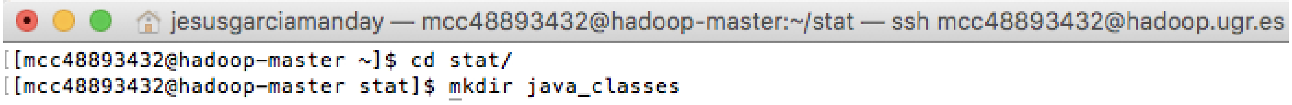
Figura 2: Copiando los ficheros Java.


jesusgarciamanday — mcc48893432@hadoop-master:~ — ssh mcc48893432@hadoop.ugr.es — 116×3

```
[[mcc48893432@hadoop-master ~]$ hdfs dfs -ls /tmp/BDCC/datasets/ECBDL14
 Found 1 items
 -rw-r--r--   2 root supergroup  102747144 2017-03-16 11:06 /tmp/BDCC/datasets/ECBDL14/ECBDL14_10tst.data
```

Figura 3: Datos de entrada.

Viendo que están disponibles, ahora nos creamos un directorio local para las clases de java.


jesusgarciamanday — mcc48893432@hadoop-master:~/stat — ssh mcc48893432@hadoop.ugr.es

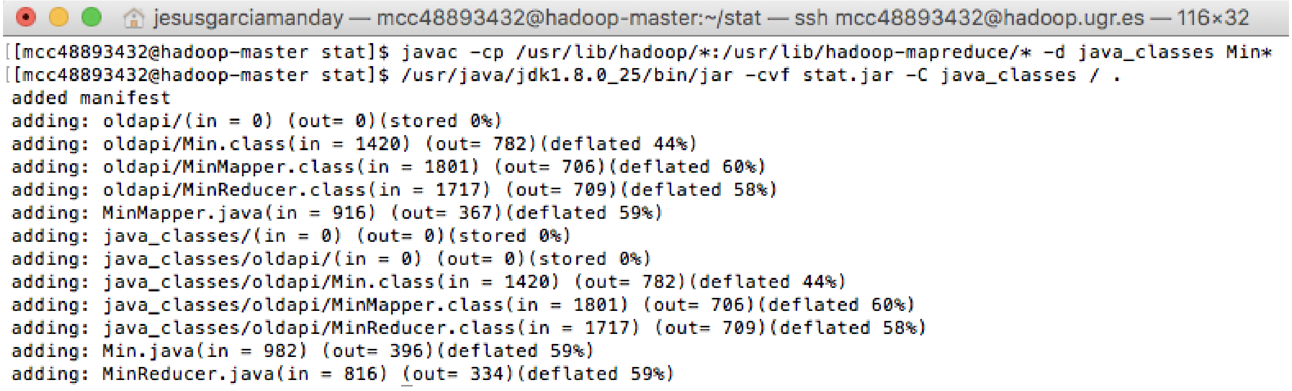
```
[[mcc48893432@hadoop-master ~]$ cd stat/
[[mcc48893432@hadoop-master stat]$ mkdir java_classes
```

Figura 4: Directorio local.

Con la preconfiguración realizada pasamos a realizar las diferentes tareas que es exponen en la práctica.

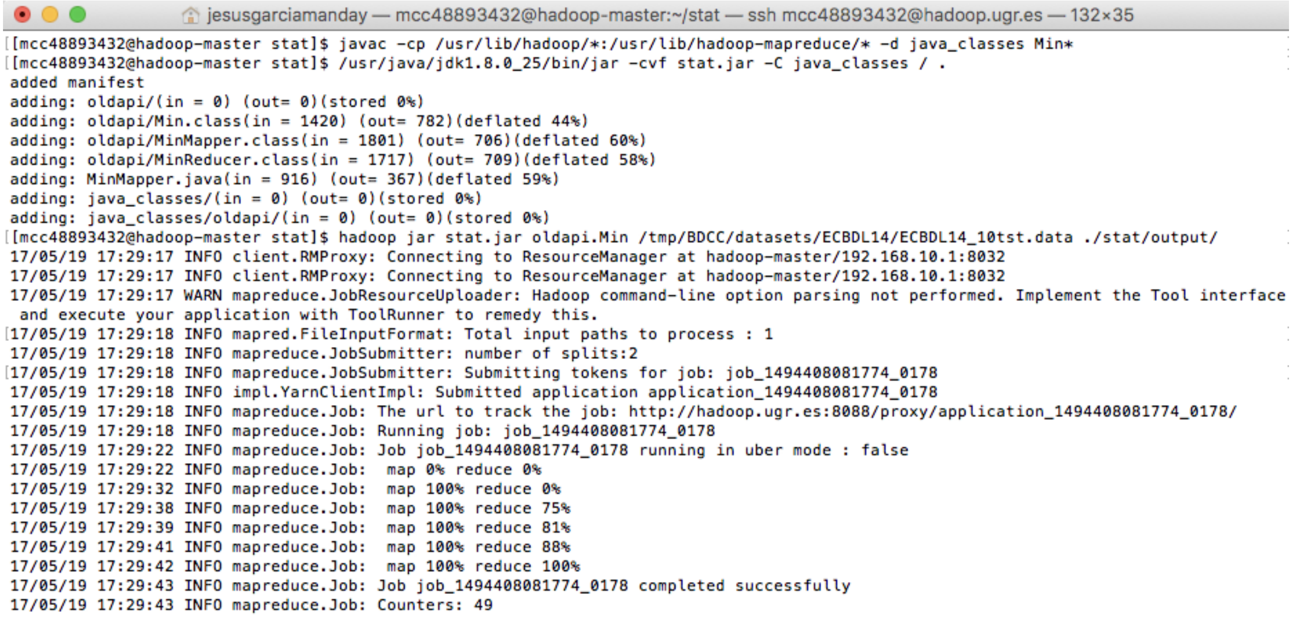
3. Calcula el valor mínimo de la variable (columna) 5.

La primera de ellas es calcular el mínimo sobre el conjunto de valores del dataset, por lo que una vez que tenemos los ficheros java ahora toca compilarlos para crear el fichero **.jar** a continuación y ejecutarlo en **hadoop**.


jesusgarciamanday — mcc48893432@hadoop-master:~/stat — ssh mcc48893432@hadoop.ugr.es — 116×32

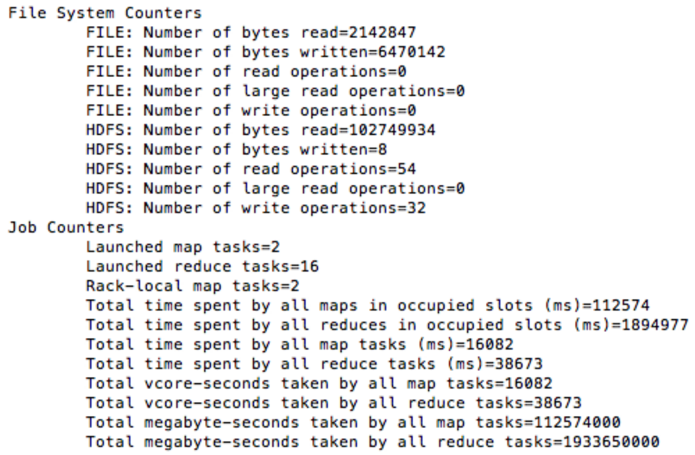
```
[[mcc48893432@hadoop-master stat]$ javac -cp /usr/lib/hadoop/*:/usr/lib/hadoop-mapreduce/* -d java_classes Min*
[[mcc48893432@hadoop-master stat]$ /usr/java/jdk1.8.0_25/bin/jar -cvf stat.jar -C java_classes / .
 added manifest
 adding: oldapi/(in = 0) (out= 0)(stored 0%)
 adding: oldapi/Min.class(in = 1420) (out= 782)(deflated 44%)
 adding: oldapi/MinMapper.class(in = 1801) (out= 706)(deflated 60%)
 adding: oldapi/MinReducer.class(in = 1717) (out= 709)(deflated 58%)
 adding: MinMapper.java(in = 916) (out= 367)(deflated 59%)
 adding: java_classes/(in = 0) (out= 0)(stored 0%)
 adding: java_classes/oldapi/(in = 0) (out= 0)(stored 0%)
 adding: java_classes/oldapi/Min.class(in = 1420) (out= 782)(deflated 44%)
 adding: java_classes/oldapi/MinMapper.class(in = 1801) (out= 706)(deflated 60%)
 adding: java_classes/oldapi/MinReducer.class(in = 1717) (out= 709)(deflated 58%)
 adding: Min.java(in = 982) (out= 396)(deflated 59%)
 adding: MinReducer.java(in = 816) (out= 334)(deflated 59%)
```

Figura 5: Compilamos y ejecutamos (I).


jesusgarciamanday — mcc48893432@hadoop-master:~/stat — ssh mcc48893432@hadoop.ugr.es — 132×35

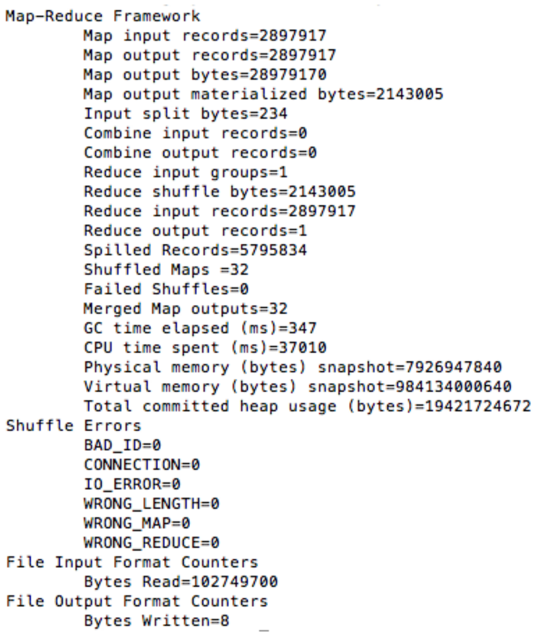
```
[[mcc48893432@hadoop-master stat]$ javac -cp /usr/lib/hadoop/*:/usr/lib/hadoop-mapreduce/* -d java_classes Min*
[[mcc48893432@hadoop-master stat]$ /usr/java/jdk1.8.0_25/bin/jar -cvf stat.jar -C java_classes / .
 added manifest
 adding: oldapi/(in = 0) (out= 0)(stored 0%)
 adding: oldapi/Min.class(in = 1420) (out= 782)(deflated 44%)
 adding: oldapi/MinMapper.class(in = 1801) (out= 706)(deflated 60%)
 adding: oldapi/MinReducer.class(in = 1717) (out= 709)(deflated 58%)
 adding: MinMapper.java(in = 916) (out= 367)(deflated 59%)
 adding: java_classes/(in = 0) (out= 0)(stored 0%)
 adding: java_classes/oldapi/(in = 0) (out= 0)(stored 0%)
[[mcc48893432@hadoop-master stat]$ hadoop jar stat.jar oldapi.Min /tmp/BDCC/datasets/ECBDL14/ECBDL14_10tst.data ./stat/output/
 17/05/19 17:29:17 INFO client.RMProxy: Connecting to ResourceManager at hadoop-master/192.168.10.1:8032
 17/05/19 17:29:17 INFO client.RMProxy: Connecting to ResourceManager at hadoop-master/192.168.10.1:8032
 17/05/19 17:29:17 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface
  and execute your application with ToolRunner to remedy this.
[17/05/19 17:29:18 INFO mapred.FileInputFormat: Total input paths to process : 1
 17/05/19 17:29:18 INFO mapreduce.JobSubmitter: number of splits:2
[17/05/19 17:29:18 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1494408081774_0178
 17/05/19 17:29:18 INFO impl.YarnClientImpl: Submitted application application_1494408081774_0178
 17/05/19 17:29:18 INFO mapreduce.Job: The url to track the job: http://hadoop.ugr.es:8088/proxy/application_1494408081774_0178/
 17/05/19 17:29:18 INFO mapreduce.Job: Running job: job_1494408081774_0178
 17/05/19 17:29:22 INFO mapreduce.Job: Job job_1494408081774_0178 running in uber mode : false
 17/05/19 17:29:22 INFO mapreduce.Job:  map 0% reduce 0%
 17/05/19 17:29:32 INFO mapreduce.Job:  map 100% reduce 0%
 17/05/19 17:29:38 INFO mapreduce.Job:  map 100% reduce 75%
 17/05/19 17:29:39 INFO mapreduce.Job:  map 100% reduce 81%
 17/05/19 17:29:41 INFO mapreduce.Job:  map 100% reduce 88%
 17/05/19 17:29:42 INFO mapreduce.Job:  map 100% reduce 100%
 17/05/19 17:29:43 INFO mapreduce.Job: Job job_1494408081774_0178 completed successfully
 17/05/19 17:29:43 INFO mapreduce.Job: Counters: 49
```

Figura 6: Compilamos y ejecutamos (II).



```
File System Counters
        FILE: Number of bytes read=2142847
        FILE: Number of bytes written=6470142
        FILE: Number of read operations=0
        FILE: Number of large read operations=0
        FILE: Number of write operations=0
        HDFS: Number of bytes read=102749934
        HDFS: Number of bytes written=8
        HDFS: Number of read operations=54
        HDFS: Number of large read operations=0
        HDFS: Number of write operations=32
Job Counters
        Launched map tasks=2
        Launched reduce tasks=16
        Rack-local map tasks=2
        Total time spent by all maps in occupied slots (ms)=112574
        Total time spent by all reduces in occupied slots (ms)=1894977
        Total time spent by all map tasks (ms)=16082
        Total time spent by all reduce tasks (ms)=38673
        Total vcore-seconds taken by all map tasks=16082
        Total vcore-seconds taken by all reduce tasks=38673
        Total megabyte-seconds taken by all map tasks=112574000
        Total megabyte-seconds taken by all reduce tasks=1933650000
```

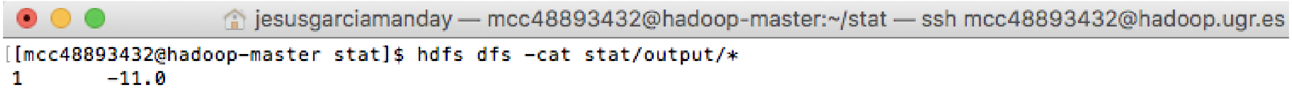
Figura 7: Compilamos y ejecutamos (III).



```
Map-Reduce Framework
        Map input records=2897917
        Map output records=2897917
        Map output bytes=28979170
        Map output materialized bytes=2143005
        Input split bytes=234
        Combine input records=0
        Combine output records=0
        Reduce input groups=1
        Reduce shuffle bytes=2143005
        Reduce input records=2897917
        Reduce output records=1
        Spilled Records=5795834
        Shuffled Maps =32
        Failed Shuffles=0
        Merged Map outputs=32
        GC time elapsed (ms)=347
        CPU time spent (ms)=37010
        Physical memory (bytes) snapshot=7926947840
        Virtual memory (bytes) snapshot=984134000640
        Total committed heap usage (bytes)=19421724672
Shuffle Errors
        BAD_ID=0
        CONNECTION=0
        IO_ERROR=0
        WRONG_LENGTH=0
        WRONG_MAP=0
        WRONG_REDUCE=0
File Input Format Counters
        Bytes Read=102749700
File Output Format Counters
        Bytes Written=8
```

Figura 8: Compilamos y ejecutamos (IV).

Por último comprobamos el resultado para ver si se ha realizado correctamente.


jesusgarciamanday — mcc48893432@hadoop-master:~/stat — ssh mcc48893432@hadoop.ugr.es

```
[[mcc48893432@hadoop-master stat]$ hdfs dfs -cat stat/output/*
 1       -11.0
```

Figura 9: Comprobando resultado.

4. **Calcula el valor máximo de la variable (columna) 5.**

5. **Calcula al mismo tiempo los valores máximo y mínimo de la variable 5.**

6. **Calcula los valores máximo y mínimo de todas las variables (salvo la última, que es la etiqueta de la clase).**

7. **Realizar la media de la variable 5.**

8. **Obtener la media de todas las variables (salvo la clase).**

9. **Comprobar si el conjunto de datos ECBDL es balanceado o no balanceado, es decir, que el ratio entre clases sea menor o mayor que 1.5 respectivamente.**

10. **Cálculo del coeficiente de correlación entre todas las parejas de variables.**