Máster en Big Data

Tecnologías de Almacenamiento

4. Hands-On: Desarrollo MapReduce

Presentado por:

Albert Ripoll y Jose David Angulo

laSalle ENG
Universitat Ramon Llull

Índice

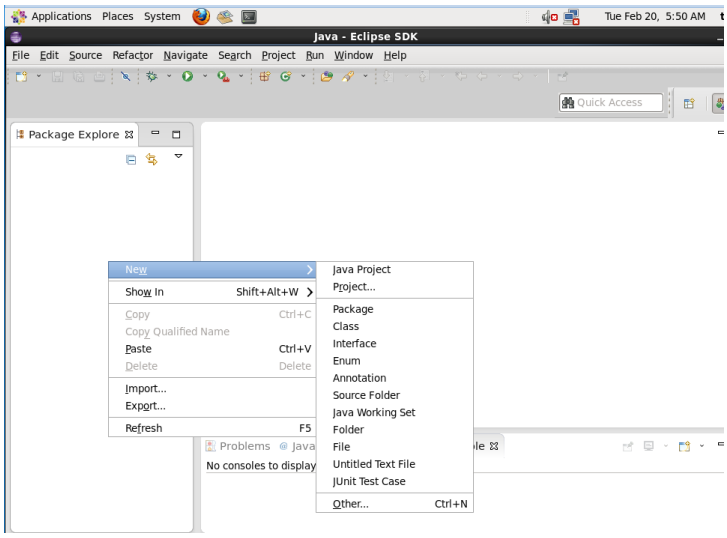
1. Introducción

El objetivo de este Hands-On es realizar el desarrollo, en Java, de dos Jobs sencillos de MapReduce y ejecutarlos.

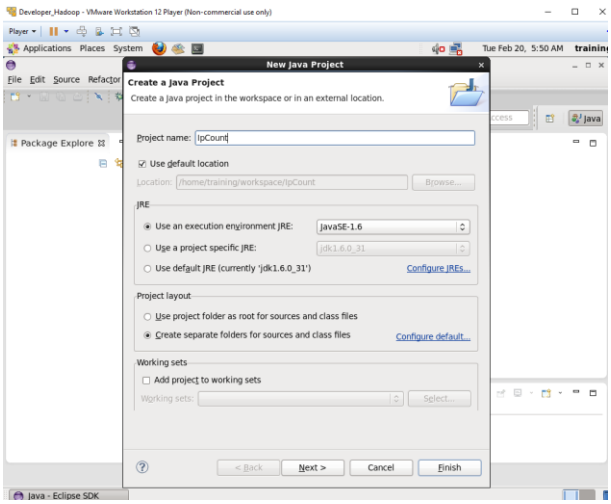
2. Entorno de desarrollo

Para realizar el desarrollo lo haremos mediante el IDE Eclipse de la máquina virtual importada en ejercicios anteriores.

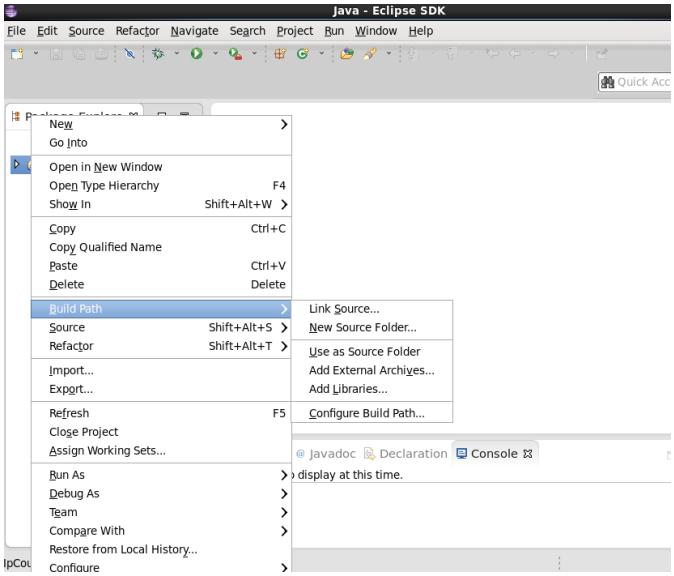
Para crear un nuevo proyecto, haremos click derecho sobre el package explorer New → Java Project



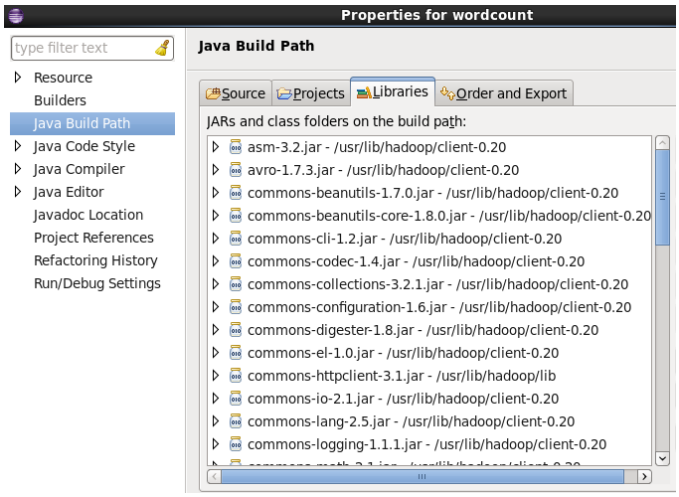
Introducimos el nombre del proyecto y click en Finish



Importamos manualmente las lilbrerías necesarias haciendo click derecho sobre el proyecto que acabamos de crear y seleccionamos Build Path → Configure Build Path



En la pestaña de libraries, seleccionamos Add Extertnal Jars e importamos todo el contenido de la carpeta /usr/lib/hadoop/client-0.20/



3. IpCount

Desarrollar y ejecutar el siguiente MapReduce:

Contar el número de veces que una misma Ip hace una petición en los logs contenidos en /weblog/ importado anteriormente

3.1. Código del Driver (diapositiva 23 + cambios resaltados)

/home/training/workspace/IpCount/src/lab_4/IpMap.java

```
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.mapreduce.Job;

public class WordCount { → Cambiado por IpDriver
public static void main(String[] args) throws Exception {
if (args.length != 2) {
System. out.printf ("Usage: WordCount <input dir> <output dir>n");
System.exit(-1);
}
Job job = new Job();
job.setJarByClass (WordCount.class); → Cambiado por IpDriver
job.setJobName ("Word Count"); → Cambiado por IpDriver
```

Comentario [ARB1]: IpDriver

Comentario [ARB2]: Was missing

Comentario [ARB3]: IpDriver

Comentario [ARB4]: Ip Driver

```
FileInputFormat.setInputPaths (job, new Path(args[0]));
FileOutputFormat.setOutputPath (job, new Path(args[1]));

job.setMapperClass (WordMapper.class) ; → Cambiado por IpMap
job.setReducerClass (SumReducer.class) ; → Cambiado por IpReducer

job.setMapOutputKeyClass (Text.class);
job.setMapOutputValueClass (IntWritable.class);

job.setOutputKeyClass (Text.class) ;
job.setOutputValueClass (IntWritable.class);

Boolean success = job.waitForCompletion(true);
System.exit(success ? 0 : 1);
}
}
```

Comentario [ARB5]: IpMap

Comentario [ARB6]: IpReduce r

3.2. Código del Maper (diapositiva 35 + cambios resaltados)

```
import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce .Mapper;

public class WordMapper extends Mapper<LongWritable, Text, Text, ; → Cambiado por IpMap
IntWritable> {
@Override
public void map(LongWritable key, Text value, Context context)
throws IOException, InterruptedException {
```

Comentario [ARB7]: IpMap

```
String line = value.toString();
for (String word : line.split("\\W+")) {
if (word.length() > 0) {; → Cambiado por String[] parts = line.split(" - - "); String word = parts[0];
context.write(new Text (word), new IntWritable(1));
}
}
```

Comentario [ARB8]: String[] parts = line.split(" - - ");
String word = parts[0];

3.3. Código del Reducer (diapositiva 38 + cambios resaltados)

```
import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce .Reducer;

public class SumReducer extends Reducer<Text, IntWritable, Text, IntWritable>
{ → Cambiado por IpReducer
@Override
public void reduce (Text key, Iterable<IntWritable> values, Context context)
throws IOException, InterruptedException {
int wordCount = 0;

for (IntWritable value : values) {
wordCount += value.get();
}
context.write (key, new IntWritable (wordCount));
}
}
```

Comentario [ARB9]: IpReduce

3.4. Ejecución del Map-Reduce por consola

• Después de que con Eclipse creásemos las clases .java de IpDriver, IpMap y IpReduce, accedemos en local para comprovar su ruta:

```
[training@localhost ~]$ ls
Desktop                                 Music        Templates
Documents                               Pictures     training
Downloads                               Public       training_materials
eclipse                                 ruta         Videos
kiji-bento-albacore-1.0.5-release.tar.gz scripts     workspace
Lab                                     spark-1.3.1  workspace.save.dev
lib                                     src
[training@localhost ~]$ cd workspace
[training@localhost workspace]$ ls
hive  IpCount  oozie-labs
[training@localhost workspace]$ cd IpCount
[training@localhost IpCount]$ ls
bin  src
[training@localhost IpCount]$ cd src
[training@localhost src]$ ls
lab_4
[training@localhost src]$ cd lab_4
[training@localhost lab_4]$ ls
IpDriver.java  IpMap.java  IpReduce.java
[training@localhost lab_4]$
```

• Con el primer commando, creamos las distintas clases dentro de la carpeta “lab_4”. Y con el Segundo commando creamos el document comprimido Lab_4.jar

```
[training@localhost lab_4]$ javac -classpath `hadoop classpath` IpMap.java IpRed
uce.java IpDriver.java
[training@localhost lab_4]$ jar cf Lab_4.jar IpMap.class IpReduce.class IpDriver
.class
```

```
[training@localhost src]$ cd lab_4
[training@localhost lab_4]$ ls
IpDriver.class  IpMap.class  IpReduce.class  Lab_4.jar
IpDriver.java   IpMap.java   _IpReduce.java
```

• Entonces vamos al directorio de src (de fuera del lab) para ejecutar el siguiente comando. En este caso los nombres del comando están cambiados de IpDriver a IPDRIVER, IpMap a IPMAPER, IpReduce a IPREDUCER y la carpeta Lab_4 a LABORATORIO4_JOSE/MYLAB4 ya que la ejecución se realizó en otra computadora con archivos con otros nombres debido a que la anterior se colgaba todo el rato.

javac -classpath `hadoop classpath` IPMAPER.java IPREDUCE.java IPDRIVER.java

jar cf MYJAR.jar IPMAPER.class IPREDUCE.class IPDRIVER.class

```
[training@localhost src]$ jar cf /home/training/training/Jars/MYLAB4.jar LAB4/IP
MAPER.class LAB4/IPREDUCE.class LAB4/IPDRIVER.class
[training@localhost src]$ hadoop jar /home/training/training/Jars/MYLAB4.jar IPD
RIVER /user/training/weblog/access_log /user/training/OUTPUTS_LAB4/IPCOUNT
Exception in thread "main" java.lang.ClassNotFoundException: IPDRIVER
        at java.net.URLClassLoader$1.run(URLClassLoader.java:202)
        at java.security.AccessController.doPrivileged(Native Method)
        at java.net.URLClassLoader.findClass(URLClassLoader.java:190)
        at java.lang.ClassLoader.loadClass(ClassLoader.java:306)
        at java.lang.ClassLoader.loadClass(ClassLoader.java:247)
        at java.lang.Class.forName0(Native Method)
        at java.lang.Class.forName(Class.java:247)
        at org.apache.hadoop.util.RunJar.main(RunJar.java:201)
[training@localhost src]$ hadoop jar /home/training/training/Jars/MYLAB4.jar LAB
4.IPDRIVER /user/training/weblog/access_log /user/training/OUTPUTS_LAB4/IPCOUNT
24/03/20 15:00:47 WARN mapred.JobClient: Use GenericOptionsParser for parsing th
e arguments. Applications should implement Tool for the same.
24/03/20 15:00:47 INFO input.FileInputFormat: Total input paths to process : 1
24/03/20 15:00:47 INFO mapred.JobClient: Running job: job_202403201451_0001
24/03/20 15:00:48 INFO mapred.JobClient:  map 0% reduce 0%
24/03/20 15:00:56 INFO mapred.JobClient:  map 25% reduce 0%
24/03/20 15:01:05 INFO mapred.JobClient:  map 50% reduce 0%
24/03/20 15:01:07 INFO mapred.JobClient:  map 50% reduce 16%
24/03/20 15:01:12 INFO mapred.JobClient:  map 75% reduce 16%
24/03/20 15:01:16 INFO mapred.JobClient:  map 75% reduce 25%
24/03/20 15:01:17 INFO mapred.JobClient:  map 100% reduce 25%
24/03/20 15:01:21 INFO mapred.JobClient:  map 100% reduce 100%
```

```
24/03/20 15:01:21 INFO mapred.JobClient: Job complete: job_202403201451_0001
24/03/20 15:01:21 INFO mapred.JobClient: Counters: 32
24/03/20 15:01:21 INFO mapred.JobClient:   File System Counters
24/03/20 15:01:21 INFO mapred.JobClient:     FILE: Number of bytes read=18248095
0
24/03/20 15:01:21 INFO mapred.JobClient:     FILE: Number of bytes written=27545
0291
24/03/20 15:01:21 INFO mapred.JobClient:     FILE: Number of read operations=0
24/03/20 15:01:21 INFO mapred.JobClient:     FILE: Number of large read operatio
ns=0
24/03/20 15:01:21 INFO mapred.JobClient:     FILE: Number of write operations=0
24/03/20 15:01:21 INFO mapred.JobClient:     HDFS: Number of bytes read=50497113
2
24/03/20 15:01:21 INFO mapred.JobClient:     HDFS: Number of bytes written=83266
99
24/03/20 15:01:21 INFO mapred.JobClient:     HDFS: Number of read operations=16
24/03/20 15:01:21 INFO mapred.JobClient:     HDFS: Number of large read operatio
ns=0
24/03/20 15:01:21 INFO mapred.JobClient:     HDFS: Number of write operations=1
24/03/20 15:01:21 INFO mapred.JobClient:   Job Counters
24/03/20 15:01:21 INFO mapred.JobClient:     Launched map tasks=8
24/03/20 15:01:21 INFO mapred.JobClient:     Launched reduce tasks=1
24/03/20 15:01:21 INFO mapred.JobClient:     Data-local map tasks=8
24/03/20 15:01:21 INFO mapred.JobClient:     Total time spent by all maps in occ
upied slots (ms)=52191
24/03/20 15:01:21 INFO mapred.JobClient:     Total time spent by all reduces in
occupied slots (ms)=23244
24/03/20 15:01:21 INFO mapred.JobClient:     Total time spent by all maps waitin
g after reserving slots (ms)=0
24/03/20 15:01:21 INFO mapred.JobClient:     Total time spent by all reduces wai
ting after reserving slots (ms)=0
24/03/20 15:01:21 INFO mapred.JobClient:   Map-Reduce Framework
24/03/20 15:01:21 INFO mapred.JobClient:     Map input records=4477843
24/03/20 15:01:21 INFO mapred.JobClient:     Map output records=4477843
24/03/20 15:01:21 INFO mapred.JobClient:     Map output bytes=82281001
24/03/20 15:01:21 INFO mapred.JobClient:     Input split bytes=928
24/03/20 15:01:21 INFO mapred.JobClient:     Combine input records=0
24/03/20 15:01:21 INFO mapred.JobClient:     Combine output records=0
24/03/20 15:01:21 INFO mapred.JobClient:     Reduce input groups=358349
24/03/20 15:01:21 INFO mapred.JobClient:     Reduce shuffle bytes=91240451
24/03/20 15:01:21 INFO mapred.JobClient:     Reduce input records=4477843
24/03/20 15:01:21 INFO mapred.JobClient:     Reduce output records=358349
24/03/20 15:01:21 INFO mapred.JobClient:     Spilled Records=13433529
24/03/20 15:01:21 INFO mapred.JobClient:     CPU time spent (ms)=13410
24/03/20 15:01:21 INFO mapred.JobClient:     Physical memory (bytes) snapshot=17
32235264
24/03/20 15:01:21 INFO mapred.JobClient:     Virtual memory (bytes) snapshot=652
6558208
24/03/20 15:01:21 INFO mapred.JobClient:     Total committed heap usage (bytes)=
1255047168
[training@localhost src]$
```

• Si observamos el resultado de la salida vemos que ha contado correctamente las Ip. El comando abre el archivo "part-r-00000" que está dentro de la carpeta "IPCOUNT" dentro de "OUTPUTS_LAB4" que son las carpetas que se han generado. Al abrir, se muestra las últimas ("Tail") "20" líneas.

```
10.173.161.1[training@localhost src]$ hadoop fs -cat OUTPUTS_LAB4/IPCOUNT/part-r-00000 | tail -n 20
10.99.96.220    1
10.99.96.84     1
10.99.97.112    1
10.99.97.130    1
10.99.97.14     6
10.99.97.193    5
10.99.97.211    19
10.99.97.217    1
10.99.97.32     1
10.99.97.34     65
10.99.97.41     1
10.99.97.54     1
10.99.97.8      1
10.99.98.150    2
10.99.98.229    2
10.99.98.77     1
10.99.99.127    1
10.99.99.186    6
10.99.99.247    1
10.99.99.58     21
[training@localhost src]$
```

4. Average Length

Desarrollar y ejecutar el siguiente MapReduce:

Calcular el promedio de la longitud de las palabras que empiecen por una letra determinada en todas las obras de Shakespeare importadas anteriormente

Ejemplo de salida :

```
N 2.0
n 3.0
d 10.0
i 2.0
t 3.5
```

4.1. Código del Driver

```
package LAB4;

import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.FloatWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.mapreduce.Job;
//user/training/weblog/access_log
public class IPDRIVER2 {
```

```
public static void main(String[] args) throws Exception {
if (args.length != 2) {
System. out.printf ("Usage: WordCount <input dir> <output dir>n");
System.exit(-1);
}
Job job = new Job();
job.setJarByClass (IPDRIVER2.class);
job.setJobName ("Ip Driver");

FileInputFormat.setInputPaths (job, new Path(args[0]));
FileOutputFormat.setOutputPath (job, new Path(args[1]));

job.setMapperClass (IPMAPER2.class) ;
job.setReducerClass (IPREDUCE2.class) ;

job.setMapOutputKeyClass (Text.class);
job.setMapOutputValueClass (FloatWritable.class);

job.setOutputKeyClass (Text.class) ;
job.setOutputValueClass (FloatWritable.class);

Boolean success = job.waitForCompletion(true);
System.exit(success ? 0 : 1);
}
}
```

4.2. Código del Maper

```
package LAB4;

import java.io.IOException;
import org.apache.hadoop.io.FloatWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce .Mapper;

public class IPMAPER2 extends Mapper<LongWritable, Text, Text,
FloatWritable> {
@Override
public void map(LongWritable key, Text value, Context context)
throws IOException, InterruptedException {

        String line = value.toString();

        for (String word: line.split("\\W+")){  //se separa la linia por palabras “word”

                if (word.length() > 0){ //Si la palabra tiene longitude mayor que 0
                        String letra = word.substring(0,1); //letra = string que va del elemento 0 al 1 de la palabra
                        float longitud = word.length(); //longitud = longitude total de la palabra

                        context.write(new Text(letra), new FloatWritable(longitud));
//Devolvemos la letra como key y la longitude de la palabra como valor float. De hecho de todo el código cambiamos los valores a
float.

        }
}
}
}
```

4.3. Código del Reducer

```
package LAB4;

import java.io.IOException;
import org.apache.hadoop.io.FloatWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce .Reducer;

public class IPREDUCE2 extends Reducer<Text, FloatWritable, Text, FloatWritable>
{
@Override
public void reduce (Text key, Iterable<FloatWritable> values, Context context)
throws IOException, InterruptedException {
float wordCount = 0; //Contador de valores (longitudes de palabras)
float i=0; //Contador de letras (palabras con misma key)

for (FloatWritable value : values) { //Para cada valor que tenemos

        wordCount += value.get(); //Sumamos acumulativamente el valor
        i+=1; //Sumamos cuantas veces hemos tenido que sumar (cuantas key así había)
        }
float promedio = wordCount / i; //Se calcula el promedio como la suma del número de letras en una palabra / cuantas palabras
había con esa key
context.write (key, new FloatWritable (promedio)); //Se usa Float en todo el código.
}
}
```

4.4. Ejecución del Map-Reduce por consola

• Ir al directorio de LAB donde están los .java

```
[training@localhost ~]$ cd workspace
[training@localhost workspace]$ ls
hive  IPDRIVER2  LABORATORIO4_JOSE  oozie-labs
[training@localhost workspace]$ cd LABORATORIO4_JOSE
[training@localhost LABORATORIO4_JOSE]$ ls
bin  src
[training@localhost LABORATORIO4_JOSE]$ cd src
[training@localhost src]$ ls
LAB4
```

• dentro ejecutar los comandos:

javac -classpath `hadoop classpath` LAB4/IPMAPER2.java LAB4/IPREDUCE2.java LAB4/IPDRIVER2.java

jar cf /home/training/training/Jars/MYJAR2.jar LAB4/IPMAPER2.class LAB4/IPREDUCE2.class LAB4/IPDRIVER2.class

hadoop jar /home/training/training/Jars/MYJAR2.jar LAB4.IPDRIVER2 /user/training/Shakespeare /home/training/OUTPUTS_LAB4/AVERAGE3

• Aquí la ejecución por consola:

```
[training@localhost src]$ ls
LAB4
[training@localhost src]$ javac -classpath `hadoop classpath` LAB4/IPMAPER2.java LAB4/IPREDUC
E2.java LAB4/IPDRIVER2.java
[training@localhost src]$ jar cf /home/training/training/Jars/MYJAR2.jar LAB4/IPMAPER2.class
LAB4/IPREDUCE2.class LAB4/IPDRIVER2.class
[training@localhost src]$ hadoop jar /home/training/training/Jars/MYJAR2.jar LAB4.IPDRIVER2 /
user/training/Shakespeare /home/training/OUTPUTS_LAB4/AVERAGE3
24/03/21 13:30:05 WARN mapred.JobClient: Use GenericOptionsParser for parsing the arguments.
Applications should implement Tool for the same.
24/03/21 13:30:05 INFO input.FileInputFormat: Total input paths to process : 4
24/03/21 13:30:05 INFO mapred.JobClient: Running job: job_202403201646_0002
24/03/21 13:30:06 INFO mapred.JobClient:  map 0% reduce 0%
24/03/21 13:30:12 INFO mapred.JobClient:  map 50% reduce 0%
24/03/21 13:30:17 INFO mapred.JobClient:  map 100% reduce 0%
24/03/21 13:30:20 INFO mapred.JobClient:  map 100% reduce 100%
24/03/21 13:30:20 INFO mapred.JobClient: Job complete: job_202403201646_0002
24/03/21 13:30:20 INFO mapred.JobClient: Counters: 32
24/03/21 13:30:20 INFO mapred.JobClient:   File System Counters
24/03/21 13:30:20 INFO mapred.JobClient:     FILE: Number of bytes read=15029594
24/03/21 13:30:20 INFO mapred.JobClient:     FILE: Number of bytes written=23705553
24/03/21 13:30:20 INFO mapred.JobClient:     FILE: Number of read operations=0
24/03/21 13:30:20 INFO mapred.JobClient:     FILE: Number of large read operations=0
24/03/21 13:30:20 INFO mapred.JobClient:     FILE: Number of write operations=0
24/03/21 13:30:20 INFO mapred.JobClient:     HDFS: Number of bytes read=5284706
24/03/21 13:30:20 INFO mapred.JobClient:     HDFS: Number of bytes written=647
24/03/21 13:30:20 INFO mapred.JobClient:     HDFS: Number of read operations=8
24/03/21 13:30:20 INFO mapred.JobClient:     HDFS: Number of large read operations=0
24/03/21 13:30:20 INFO mapred.JobClient:     HDFS: Number of write operations=1
24/03/21 13:30:20 INFO mapred.JobClient:   Job Counters
24/03/21 13:30:20 INFO mapred.JobClient:     Launched map tasks=4
24/03/21 13:30:20 INFO mapred.JobClient:     Launched reduce tasks=1
24/03/21 13:30:20 INFO mapred.JobClient:     Data-local map tasks=4
24/03/21 13:30:20 INFO mapred.JobClient:     Total time spent by all maps in occupied slots (
ms)=19833
24/03/21 13:30:20 INFO mapred.JobClient:     Total time spent by all reduces in occupied slot
s (ms)=7304
24/03/21 13:30:20 INFO mapred.JobClient:     Total time spent by all maps waiting after reser
ving slots (ms)=0
24/03/21 13:30:20 INFO mapred.JobClient:     Total time spent by all reduces waiting after re
serving slots (ms)=0
24/03/21 13:30:20 INFO mapred.JobClient:   Map-Reduce Framework
24/03/21 13:30:20 INFO mapred.JobClient:     Map input records=173126
24/03/21 13:30:20 INFO mapred.JobClient:     Map output records=964453
24/03/21 13:30:20 INFO mapred.JobClient:     Map output bytes=5786718
24/03/21 13:30:20 INFO mapred.JobClient:     Input split bytes=475
24/03/21 13:30:20 INFO mapred.JobClient:     Combine input records=0
24/03/21 13:30:20 INFO mapred.JobClient:     Combine output records=0
24/03/21 13:30:20 INFO mapred.JobClient:     Reduce input groups=60
24/03/21 13:30:20 INFO mapred.JobClient:     Reduce shuffle bytes=7715648
24/03/21 13:30:20 INFO mapred.JobClient:     Reduce input records=964453


24/03/21 13:30:20 INFO mapred.JobClient:     Reduce output records=60
24/03/21 13:30:20 INFO mapred.JobClient:     Spilled Records=2843147
24/03/21 13:30:20 INFO mapred.JobClient:     CPU time spent (ms)=3640
24/03/21 13:30:20 INFO mapred.JobClient:     Physical memory (bytes) snapshot=884535296
24/03/21 13:30:20 INFO mapred.JobClient:     Virtual memory (bytes) snapshot=3618029568
24/03/21 13:30:20 INFO mapred.JobClient:     Total committed heap usage (bytes)=557989888
```

• Leemos el fichero donde se han guardado las palabras y su conteo:

hadoop fs -cat /home/training/OUTPUTS_LAB4/AVERAGE3/part-r-00000

```
[training@localhost src]$ hadoop fs -ls /home/training/OUTPUTS_LAB4/AVERAGE3
Found 3 items
-rw-r--r--   1 training supergroup          0 2024-03-21 13:30 /home/training/OUTPUTS_LAB4/AV
ERAGE3/_SUCCESS
drwxr-xr-x   - training supergroup          0 2024-03-21 13:30 /home/training/OUTPUTS_LAB4/AV
ERAGE3/_logs
-rw-r--r--   1 training supergroup        647 2024-03-21 13:30 /home/training/OUTPUTS_LAB4/AV
ERAGE3/part-r-00000
[training@localhost src]$ hadoop fs -cat /home/training/OUTPUTS_LAB4/AVERAGE3/part-r-00000
1       1.02
2       1.0588236
3       1.0
4       1.5
5       1.5
6       1.5
7       1.0
8       1.5
9       1.0
A       3.8913946
B       5.1393027
C       6.6296945
D       5.2018347
E       5.5142636
F       5.2555285
G       5.809792
H       4.4210725
I       1.4526861
J       4.9840083
K       4.657107
L       5.1158814
M       5.4464655
N       3.9848387
O       2.8794768
P       6.5057406
Q       5.5216427
R       5.929275
S       5.293126
T       3.9591436
```