Máster en Big Data

Tecnologías de Almacenamiento

3. Hands-On: Ejecución MapReduce

Índice

1. Introducción

El objetivo de este Hands-On es aprender a ejecutar jobs de MapReduce por línea de comandos y entender su comportamiento. El entorno de ejecución será la máquina desplegada en el Hands-On anterior.

2. Ejecutando un Job

Ejecución un MapReduce

- Una vez las clases necesarias estén implementadas, se deben compilar

```
$ javac -classpath `hadoop classpath` MyMapper.java
  MyReducer.java MyDriver.java
```

- Empaquetarlas en un jar

```
$ jar cf MyMR.jar MyMapper.class MyReducer.class
  MyDriver.class
```

- Y lanzar el job al clúster utilizando el comando siguiente

```
$ hadoop jar MyMR.jar MyDriver in_file out_dir
```

Todas las instrucciones deben introducirse mediante Shell de Linux.

a) Ejecutar el job del proyecto WordCount ya compilado en la carpeta */home/training/training/Jars* (la clase de entrada y su package es *nextret.WordCount*)

• Usamos *cd* para salir de las carpetas. Usamos *ls* para listar lo que tenemos allí. Usmos *cd + nombre carpeta* para entrar en la carpeta.

```
[training@localhost ~]$ cd
[training@localhost ~]$ cd
[training@localhost ~]$ ls
Desktop  Documents  Downloads  eclipse  kiji-bento-albacore-1.0.5-release.tar.gz  Lab

lib  Music  Pictures  Public  ruta  scripts  spark-1.3.1  src  Templates
```

```
 training  training_materials  Videos  workspace  workspace.save.dev

[training@localhost ~]$ cd training
[training@localhost training]$ ls
Jars
[training@localhost training]$ cd Jars
[training@localhost Jars]$ ls
wordcount.jar
```

De este modo en el siguiente paso no hace falta poner la ruta.

• Ejecutamos el proyecto wordcount.jar. Para hacerlo usamos el siguiente esquema

```
$ hadoop jar MyMR.jar MyDriver in_file out_dir
```

Donde:

- MyMR es el jar con su path. En nuestro caso como ya estamos en el directorio no hace falta poner el path.
- MyDriver es nextret.WordCount
- In_file son los datos a leer. En este caso leemos el texto de Shakespeare
- Out_dir es el directorio de salida. Tiene que ser uno que no exista, como por ejemplo una carpeta llamada “Output” dentro de traning.

Vemos:

```
[training@localhost Jars]$ hadoop jar wordcount.jar nextret.WordCount /user/training/Shakespeare /user/training/Outpu
t
```

Como respuesta obtenemos:

```
24/03/13 14:59:56 WARN mapred.JobClient: Use GenericOptionsParser for parsing the arguments. Applications should impl
ement Tool for the same.
24/03/13 14:59:56 INFO input.FileInputFormat: Total input paths to process : 5
24/03/13 14:59:56 INFO mapred.JobClient: Running job: job_202403111456_0003
24/03/13 14:59:57 INFO mapred.JobClient:  map 0% reduce 0%
24/03/13 15:00:04 INFO mapred.JobClient:  map 40% reduce 0%
24/03/13 15:00:09 INFO mapred.JobClient:  map 80% reduce 0%
24/03/13 15:00:10 INFO mapred.JobClient:  map 100% reduce 0%
24/03/13 15:00:13 INFO mapred.JobClient:  map 100% reduce 100%
24/03/13 15:00:13 INFO mapred.JobClient: Job complete: job_202403111456_0003
24/03/13 15:00:13 INFO mapred.JobClient: Counters: 32
24/03/13 15:00:13 INFO mapred.JobClient:   File System Counters
24/03/13 15:00:13 INFO mapred.JobClient:     FILE: Number of bytes read=20983046
24/03/13 15:00:13 INFO mapred.JobClient:     FILE: Number of bytes written=32957704
24/03/13 15:00:13 INFO mapred.JobClient:     FILE: Number of read operations=0
24/03/13 15:00:13 INFO mapred.JobClient:     FILE: Number of large read operations=0
24/03/13 15:00:13 INFO mapred.JobClient:     FILE: Number of write operations=0
24/03/13 15:00:13 INFO mapred.JobClient:     HDFS: Number of bytes read=5343801
24/03/13 15:00:13 INFO mapred.JobClient:     HDFS: Number of bytes written=324841
24/03/13 15:00:13 INFO mapred.JobClient:     HDFS: Number of read operations=10
24/03/13 15:00:13 INFO mapred.JobClient:     HDFS: Number of large read operations=0
24/03/13 15:00:13 INFO mapred.JobClient:     HDFS: Number of write operations=1
24/03/13 15:00:13 INFO mapred.JobClient:   Job Counters
24/03/13 15:00:13 INFO mapred.JobClient:     Launched map tasks=5
24/03/13 15:00:13 INFO mapred.JobClient:     Launched reduce tasks=1
24/03/13 15:00:13 INFO mapred.JobClient:     Data-local map tasks=5
24/03/13 15:00:13 INFO mapred.JobClient:     Total time spent by all maps in occupied slots (ms)=22382
24/03/13 15:00:13 INFO mapred.JobClient:     Total time spent by all reduces in occupied slots (ms)=9222
24/03/13 15:00:13 INFO mapred.JobClient:     Total time spent by all maps waiting after reserving slots (ms)=0
24/03/13 15:00:13 INFO mapred.JobClient:     Total time spent by all reduces waiting after reserving slots (ms)=0
24/03/13 15:00:13 INFO mapred.JobClient:   Map-Reduce Framework
24/03/13 15:00:13 INFO mapred.JobClient:     Map input records=175558
24/03/13 15:00:13 INFO mapred.JobClient:     Map output records=974078
24/03/13 15:00:13 INFO mapred.JobClient:     Map output bytes=8880434
24/03/13 15:00:13 INFO mapred.JobClient:     Input split bytes=594
24/03/13 15:00:13 INFO mapred.JobClient:     Combine input records=0
24/03/13 15:00:13 INFO mapred.JobClient:     Combine output records=0
24/03/13 15:00:13 INFO mapred.JobClient:     Reduce input groups=31809
24/03/13 15:00:13 INFO mapred.JobClient:     Reduce shuffle bytes=10828620
24/03/13 15:00:13 INFO mapred.JobClient:     Reduce input records=974078
24/03/13 15:00:13 INFO mapred.JobClient:     Reduce output records=31809
24/03/13 15:00:13 INFO mapred.JobClient:     Spilled Records=2862397
24/03/13 15:00:13 INFO mapred.JobClient:     CPU time spent (ms)=4770
24/03/13 15:00:13 INFO mapred.JobClient:     Physical memory (bytes) snapshot=1092792320
24/03/13 15:00:13 INFO mapred.JobClient:     Virtual memory (bytes) snapshot=4343140352
24/03/13 15:00:13 INFO mapred.JobClient:     Total committed heap usage (bytes)=689524736
[training@localhost Jars]$
```

b) Revisa los archivos generados y visualiza los resultados. Explica los resultados.

- Visitamos la nube para ver si se creó la carpeta *Output*.

```
[training@localhost training]$ hadoop fs -ls
Found 4 items
drwxr-xr-x   - training supergroup          0 2024-03-13 15:00 Output
drwxr-xr-x   - training supergroup          0 2024-03-06 15:33 Shakespeare
drwxr-xr-x   - training supergroup          0 2024-03-06 15:54 Shakespeare1
drwxr-xr-x   - training supergroup          0 2024-03-06 14:56 weblog
[training@localhost training]$
```

- Vemos lo que hay dentro de la carpeta.

```
[training@localhost training]$ hadoop fs -ls /user/training/Output
Found 3 items
-rw-r--r--   1 training supergroup          0 2024-03-13 15:00 /user/training/Output/_SUCCESS
drwxr-xr-x   - training supergroup          0 2024-03-13 14:59 /user/training/Output/_logs
-rw-r--r--   1 training supergroup     324841 2024-03-13 15:00 /user/training/Output/part-r-00000
```

- Dentro de la carpeta *_logs* hay otra carpeta *history*

```
[training@localhost training]$ hadoop fs -ls /user/training/Output/_logs
Found 1 items
drwxr-xr-x   - training supergroup          0 2024-03-13 14:59 /user/training/Output/_logs/history
```

- Dentro de la carpeta *history* hay 2 archivos.

```
[training@localhost training]$ hadoop fs -ls /user/training/Output/_logs/history
Found 2 items
-rw-r--r--   1 training supergroup      33124 2024-03-13 15:00 /user/training/Output/_logs/history/job_202403111456_0003_1710356396754_training_Word+Count
-rw-r--r--   1 training supergroup      82226 2024-03-13 14:59 /user/training/Output/_logs/history/job_202403111456_0003_conf.xml
```

- Entonces explicamos esquemáticamente lo que hay en Output

  • Archivo _SUCCESS que no tiene peso. Simplemente es un archivo que indica que se ha realizado correctamente la ejecución.

  • Carpeta _logs.

    - Carpeta history
      - Archivo job_202403111456_0003_1710356396754_training_Word+Count
      - Archivo job_202403111456_0003_conf.xml

  • Archivo part-r-00000.

- Leemos lo que hay en el archivo part-r-00000

```
[training@localhost training]$ hadoop fs -cat /user/training/Output/part-r-00000
```

- Y la respuesta es muy larga (1400 páginas de Word) así que reducimos a las últimas palabras.

```
[training@localhost training]$ hadoop fs -cat /user/training/Output/part-r-00000 | tail -n20
```

```
yourself        281
yourselves      74
youth   288
youthful        32
youths  5
zanies  1
zany    1
zeal    33
zealous 6
zeals   1
zed     1
zenith  1
zephyrs 1
zir     2
zo      1
zodiac  1
zodiacs 1
zone    1
zounds  3
zwaggered       1
```

- Tenemos el conteo de cada palabra. Key, tab, conteo.

c) Vuelve a ejecutar el mismo comando para ejecutar el Job, ¿qué ha pasado?

Volvemos a hacer los comandos anteriores y la respuesta es que “FileAlreadyEsistsExemption: Output already exists.” Porque ya existe la carpeta “Output” que acabamos de crear.

```
[training@localhost training]$ cd
[training@localhost ~]$ cd
[training@localhost ~]$ ls
Desktop  Documents  Downloads  eclipse  kiji-bento-albacore-1.0.5-release.tar.gz  lib  Music  Pictures  Public  ruta  scripts  src  Templates  training  training_materials  Videos  workspace  workspace.save.dev
[training@localhost ~]$ cd training
[training@localhost training]$ ls
Jars
[training@localhost training]$ cd Jars
[training@localhost Jars]$ ls
wordcount.jar
[training@localhost Jars]$ hadoop jar wordcount.jar nextret.WordCount /user/training/Shakespeare /user/training/Output
24/03/13 15:28:43 WARN mapred.JobClient: Use GenericOptionsParser for parsing the arguments. Applications should implement Tool for the same.
24/03/13 15:28:43 INFO mapred.JobClient: Cleaning up the staging area hdfs://0.0.0.0:8020/var/lib/hadoop-hdfs/cache/mapred/mapred/staging/training/.staging/job_202403111456_0004
24/03/13 15:28:43 ERROR security.UserGroupInformation: PriviledgedActionException as:training (auth:SIMPLE) cause:org.apache.hadoop.mapred.FileAlreadyExistsException: Output directory /user/training/Output already exists
Exception in thread "main" org.apache.hadoop.mapred.FileAlreadyExistsException: Output directory /user/training/Output already exists
        at org.apache.hadoop.mapreduce.lib.output.FileOutputFormat.checkOutputSpecs(FileOutputFormat.java:132)
        at org.apache.hadoop.mapred.JobClient$2.run(JobClient.java:985)
        at org.apache.hadoop.mapred.JobClient$2.run(JobClient.java:946)
        at java.security.AccessController.doPrivileged(Native Method)
        at javax.security.auth.Subject.doAs(Subject.java:396)
        at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1408)
        at org.apache.hadoop.mapred.JobClient.submitJobInternal(JobClient.java:946)
        at org.apache.hadoop.mapreduce.Job.submit(Job.java:566)
        at org.apache.hadoop.mapreduce.Job.waitForCompletion(Job.java:596)
        at nextret.WordCount.main(WordCount.java:98)
        at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
        at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39)
        at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:25)
        at java.lang.reflect.Method.invoke(Method.java:597)
        at org.apache.hadoop.util.RunJar.main(RunJar.java:208)
[training@localhost Jars]$
```

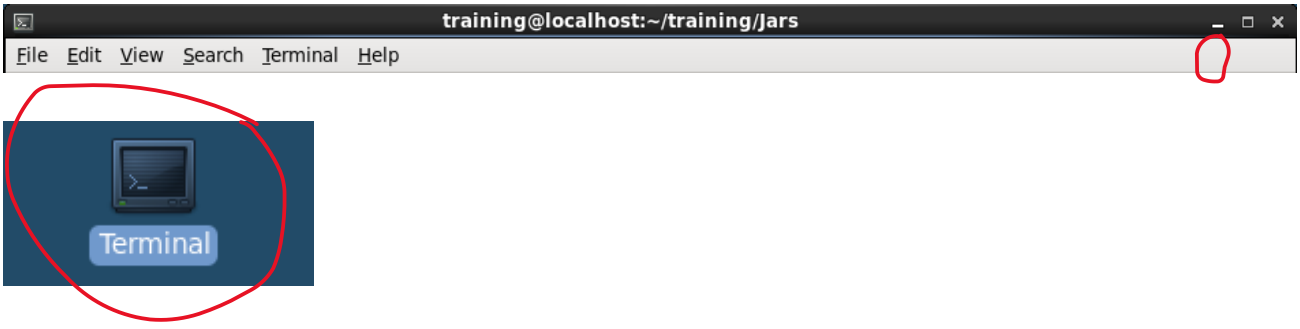
d) Elimina todos los archivos de esta última ejecución

Eliminamos y vemos que ya no aparece la carpeta “Output” en ese path.

```
[training@localhost Jars]$ hadoop fs -rm -r /user/training/Output/
Deleted /user/training/Output
[training@localhost Jars]$ hadoop fs -ls /user/training/
Found 3 items
drwxr-xr-x   - training supergroup          0 2024-03-06 15:33 /user/training/Shakespeare
drwxr-xr-x   - training supergroup          0 2024-03-06 15:54 /user/training/Shakespeare1
drwxr-xr-x   - training supergroup          0 2024-03-06 14:56 /user/training/weblog
```

3. Finalizando un Job

a) Abre dos terminales de Linux. En uno, vuelve a ejecutar el Job con el volcado del resultado en un nuevo directorio. En el otro, lista los Jobs en ejecución *(mapred job -list)* y mata el Job actual *(mapred job -kill <jobId>)*



En una consola preparamos el siguiente comando:

```
[training@localhost Jars]$ hadoop jar wordcount.jar nextret.WordCount /user/training/Shakespeare /user/training/Output
```

En otra consola de preparamos “mapred job”. Primero ejecutamos lo anterior y después este :

```
[training@localhost ~]$ mapred job -list
```

Que devuelve

```
1 jobs currently running
JobId   State   StartTime       UserName        Priority        SchedulingInfo
job_202403111456_0008   4       1710359015824   training        NORMAL  NA
```

Y de forma rápida (antes que termine) escribimos:

```
[training@localhost ~]$ mapred job -kill job_202403111456_0008
```

Donde el número de job es el que ha devuelto anteriormente. Para hacerlo exitosamente nos damos cuenta que solo cambia el valor final de 005 a 006 a 007 para cada jo basí que tenemos preparado previamente el comando y con solo ↑ ya lo tenemos escrito.

La respuesta en la consola donde lo matamos es:

```
Killed job job_202403111456_0008
```

La respuesta donde se está ejecutando el job es:

```
24/03/13 15:43:35 WARN mapred.JobClient: Use GenericOptionsParser for parsing the arguments. Applications should implement Tool for t
he same.
24/03/13 15:43:35 INFO input.FileInputFormat: Total input paths to process : 5
24/03/13 15:43:35 INFO mapred.JobClient: Running job: job_202403111456_0008
24/03/13 15:43:36 INFO mapred.JobClient:  map 0% reduce 0%
24/03/13 15:43:42 INFO mapred.JobClient: Job complete: job_202403111456_0008
24/03/13 15:43:42 INFO mapred.JobClient: Counters: 6
24/03/13 15:43:42 INFO mapred.JobClient:   Job Counters
24/03/13 15:43:42 INFO mapred.JobClient:     Launched map tasks=2
24/03/13 15:43:42 INFO mapred.JobClient:     Data-local map tasks=2
24/03/13 15:43:42 INFO mapred.JobClient:     Total time spent by all maps in occupied slots (ms)=7760
24/03/13 15:43:42 INFO mapred.JobClient:     Total time spent by all reduces in occupied slots (ms)=1192
24/03/13 15:43:42 INFO mapred.JobClient:     Total time spent by all maps waiting after reserving slots (ms)=0
```