Prácticas BigData

1. MapReduce

- Vamos a subir al directorio prácticas un fichero denominado "quijote.txt" que contiene el Quijote. Lo tienes disponible en los recursos de las prácticas. Lo más sencillo es que lo descargues desde la propia máquina virtual

```
hdfs dfs -put /home/hadoop/Descargas/quijote.txt /practicas
```

- **NOTA IMPORTANTE**: Aquellos que estáis usando **Hadoop 3**, es posible que el siguiente ejemplo no funcione correctamente. En ese caso tenemos que añadir al fichero yarn-site.xml el siguiente contenido. Por supuesto adaptarlo a vuestro HADOOP_PATH

```
<property>
<name>yarn.application.classpath</name>
   <value>
         /opt/hadoop3/hadoop/etc/hadoop,
         /opt/hadoop3/share/hadoop/common/*,
         /opt/hadoop3/share/hadoop/common/lib/*,
         /opt/hadoop3/share/hadoop/hdfs/*,
         /opt/hadoop3/share/hadoop/hdfs/lib/*,
         /opt/hadoop3/share/hadoop/mapreduce/*,
         /opt/hadoop3/share/hadoop/mapreduce/lib/*,
         /opt/hadoop3/share/hadoop/yarn/*,
         /opt/hadoop3/share/hadoop/yarn/lib/*
</value>
</property>
```

- Lanzamos el wordcount contra el fichero. Indicamos el directorio de salida donde dejar el resultado, en este caso en /practicas/resultado (siempre en HDFS)

```
hadoop jar /opt/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.9.0.jar wordcount /practicas/quijote.txt /practicas/resultado

8/01/06 19:29:24 INFO Configuration.deprecation: session.id is deprecated. Instead, use dfs.metrics.session-id

18/01/06 19:29:24 INFO jvm.JvmMetrics: Initializing JVM Metrics with processName=JobTracker, sessionId=

18/01/06 19:29:26 INFO input.FileInputFormat: Total input files to process : 1

18/01/06 19:29:27 INFO mapreduce.JobSubmitter: number of splits:1

18/01/06 19:29:28 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local382862986_0001

18/01/06 19:29:28 INFO mapreduce.Job: The url to track the job: http://localhost:8080/

18/01/06 19:29:28 INFO mapreduce.Job: Running job: job_local382862986_0001
```

18/01/06 19:29:28 INFO mapred.LocalJobRunner: OutputCommitter set in config null

18/01/06 19:29:28 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 1

18/01/06 19:29:28 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:false, ignore cleanup failures: false

18/01/06 19:29:28 INFO mapred.LocalJobRunner: OutputCommitter is org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter

…..

……

……

8/01/06 19:29:35 INFO mapreduce.Job: Job job_local382862986_0001 completed successfully

18/01/06 19:29:35 INFO mapreduce.Job: Counters: 35

File System Counters

FILE: Number of bytes read=1818006

FILE: Number of bytes written=3374967

FILE: Number of read operations=0

FILE: Number of large read operations=0

FILE: Number of write operations=0

HDFS: Number of bytes read=4397854

HDFS: Number of bytes written=448894

HDFS: Number of read operations=13

HDFS: Number of large read operations=0

HDFS: Number of write operations=4

Map-Reduce Framework

Map input records=37861

Map output records=384260

Map output bytes=3688599

Map output materialized bytes=605509

Input split bytes=108

Combine input records=384260

Combine output records=40059

Reduce input groups=40059

Reduce shuffle bytes=605509

Reduce input records=40059

```
Reduce output records=40059
Spilled Records=80118
Shuffled Maps =1
Failed Shuffles=0
Merged Map outputs=1
GC time elapsed (ms)=100
Total committed heap usage (bytes)=331489280
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=2198927
File Output Format Counters
Bytes Written=448894
```

- Vemos que nos hace un resumen del resultado
- Podemos ver el contenido del directorio

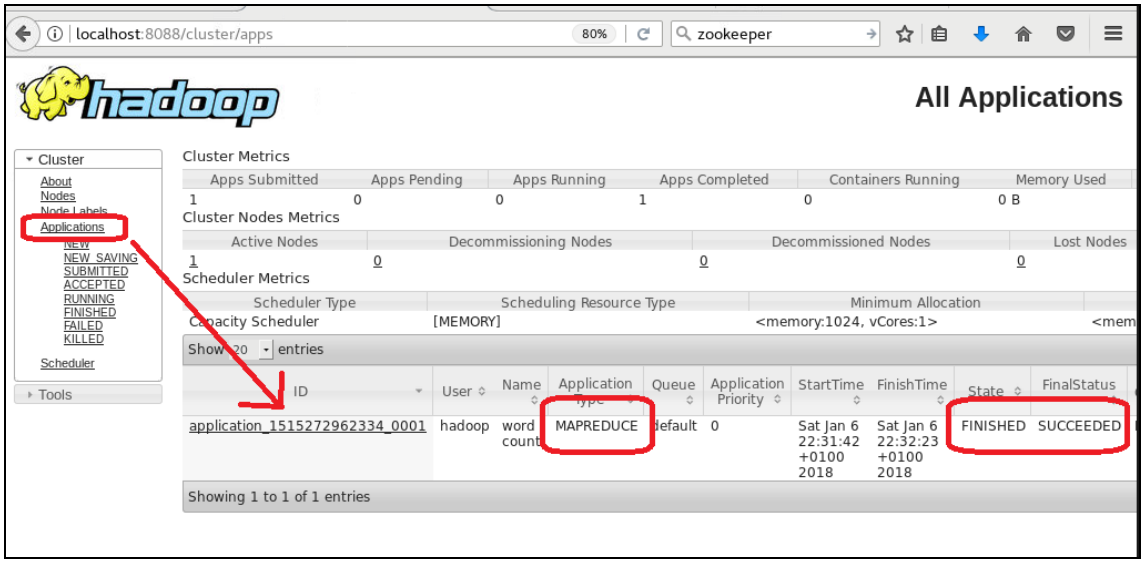
```
hdfs dfs -ls /practicas/resultado
Found 2 items
-rw-r--r--   1 hadoop supergroup          0 2018-01-06 19:29 /practicas/resultado/_SUCCESS
-rw-r--r--   1 hadoop supergroup     448894 2018-01-06 19:29 /practicas/resultado/part-r-00000
```

- Podemos traerlo desde HDFS al Linux con el comando “get” y lo dejamos en /tmp con otro nombre

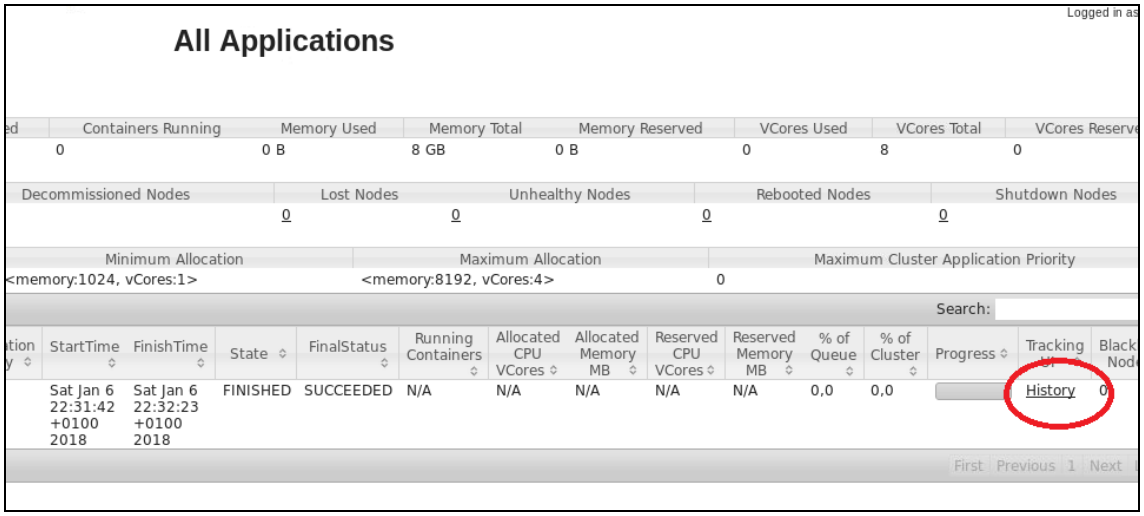
```
hdfs dfs -get /practicas/resultado/part-r-00000 /tmp/palabras_quijote.txt
Con “vi” podemos ver el contenido
Mal    1
"Al    1
"Cuando 2
"Cuidados      1
"De    2
```

```
"Defects,"      1
"Desnudo        1
"Dijo   1
"Dime   1
"Don    1
"Donde  1
"Dulcinea       1
"El     2
"Esta   1
"Harto  1
"Iglesia,       1
"Information    1
"Más    2
"No     5
"Nunca  1
"Plain  2
"Project        5
"Que    1
"Quien  1
"Right  1
"Salta  1
"Sancho 1
"Si     3
"Tened  1
"Toda   1
"Vengan 1
"Vete,  1
"/tmp/palabras_quijote.txt" 40059L, 448894C
```

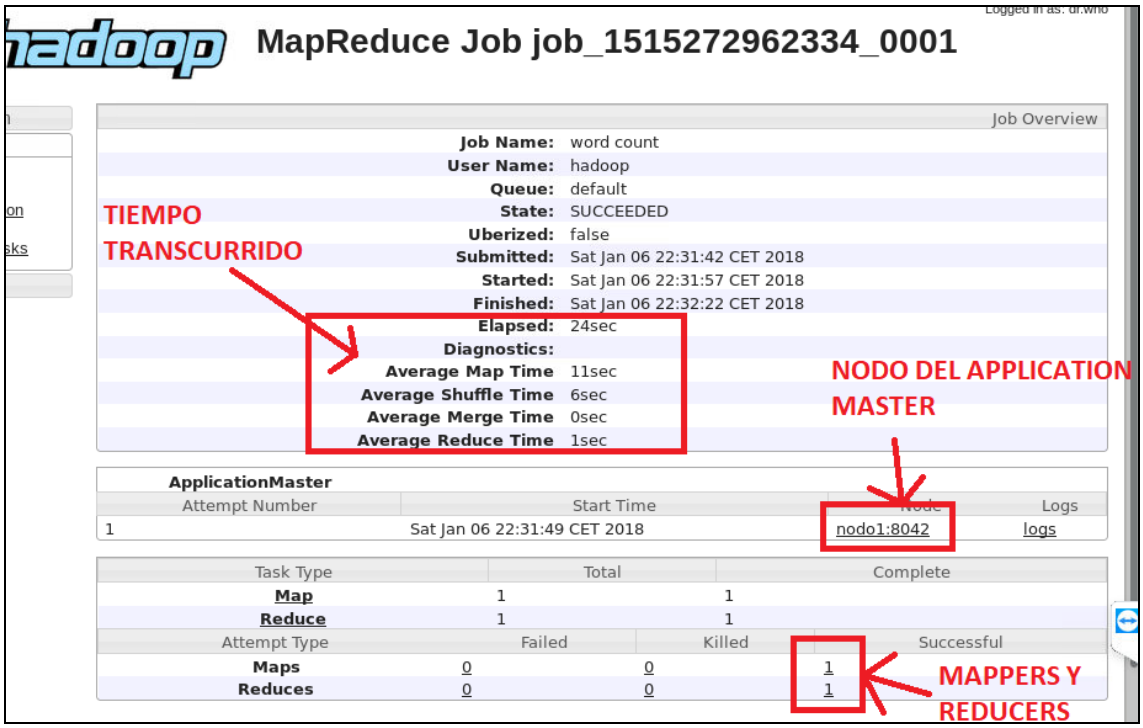
- Accedemos a la WEB de Administración de YARN.
- Si seleccionamos la opción “Applications” podemos ver la aplicación que acabamos de lanzar



- A la derecha de la aplicación, si pulsamos sobre “history”, podremos ver el detalle completo de la aplicación



- Podemos ver información muy valiosa
-



- Seleccionando un mapper o un reducer podemos acceder a su información: nodo en el que se ha ejecutado, etc…