

# Prácticas BigData

## 1. Lanzar un proceso en streaming con comandos Shell de Linux

- En este caso, vamos a usar comandos de la Shell de Linux para hacer de Mapper y Reducer, simulando el programa que hemos hecho en el punto anterior.
- Lo hacemos con la librería de Streaming
- Le pasamos el fichero y en el mapper extraemos solo los datos del campo 2 con el comando “cut” y eliminamos duplicados con el reducer y el comando “uniq”.
- NOTA: recuerda adaptar el número de tu versión de hadoop al fichero streaming

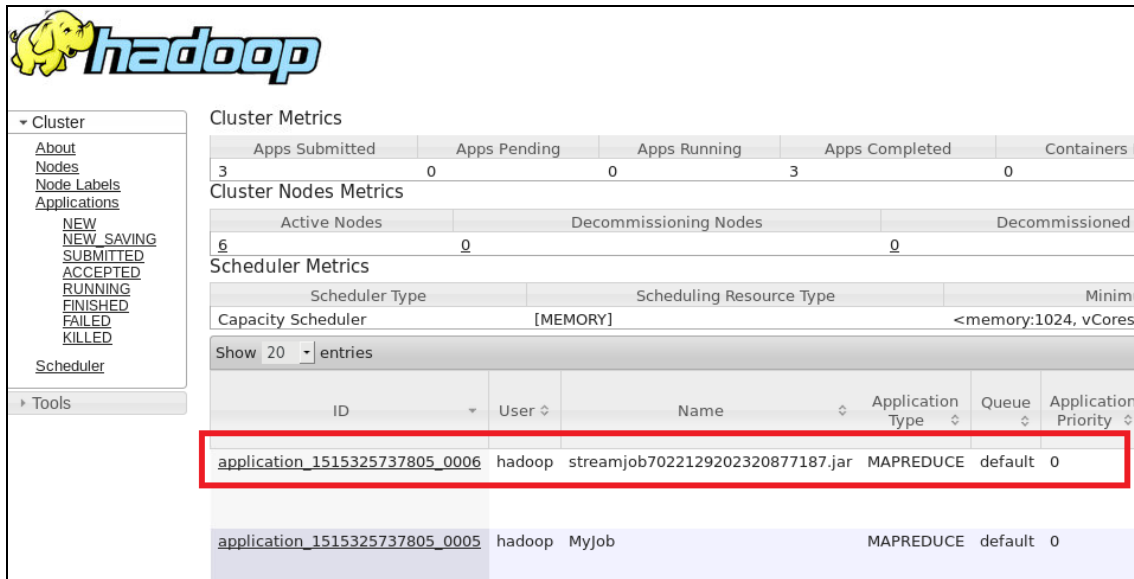
```
hadoop jar /opt/hadoop/share/hadoop/tools/lib/hadoop-streaming-2.9.0.jar -input /practicas/cite75_99.txt -output /resultado8 -mapper 'cut -f 2 -d ,' -reducer 'uniq'
```

```
packageJobJar:          [/tmp/hadoop-unjar2997682931744806206/] []
/tmp/streamjob7022129202320877187.jar tmpDir=null

18/01/07 15:08:26 INFO client.RMProxy: Connecting to ResourceManager at
nodo1/192.168.56.101:8032
18/01/07 15:08:26 INFO client.RMProxy: Connecting to ResourceManager at
nodo1/192.168.56.101:8032
18/01/07 15:08:27 INFO mapred.FileInputFormat: Total input files to process : 1
18/01/07 15:08:27 INFO net.NetworkTopology: Adding a new node: /default-
rack/192.168.56.132:50010
18/01/07 15:08:27 INFO net.NetworkTopology: Adding a new node: /default-
rack/192.168.56.123:50010
18/01/07 15:08:27 INFO net.NetworkTopology: Adding a new node: /default-
rack/192.168.56.103:50010
18/01/07 15:08:27 INFO net.NetworkTopology: Adding a new node: /default-
rack/192.168.56.125:50010
18/01/07 15:08:28 INFO mapreduce.JobSubmitter: number of splits:2
18/01/07 15:08:29 INFO Configuration.deprecation: yarn.resourcemanager.system-
metrics-publisher.enabled is deprecated. Instead, use yarn.system-metrics-
publisher.enabled
18/01/07 15:08:29 INFO mapreduce.JobSubmitter: Submitting tokens for job:
job_1515325737805_0006
18/01/07 15:08:30 INFO impl.YarnClientImpl: Submitted application
application_1515325737805_0006
18/01/07 15:08:30 INFO mapreduce.Job: The url to track the job:
http://nodo1:8088/proxy/application_1515325737805_0006/
```

18/01/07 15:08:30 INFO mapreduce.Job: Running job: job\_1515325737805\_0006

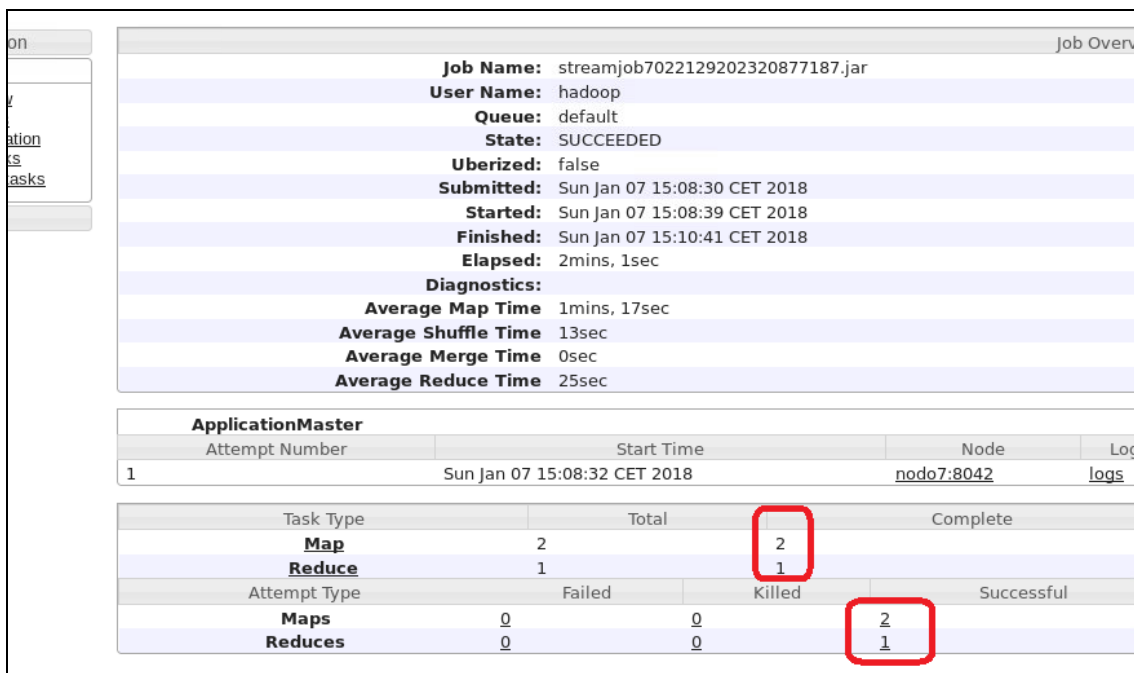
- Podemos verlo en la WEB de administración



The screenshot shows the Hadoop Administration Web Interface. On the left is a navigation menu with links like 'Cluster', 'About', 'Nodes', 'Node Labels', 'Applications', and 'Tools'. The main content area displays 'Cluster Metrics' and 'Cluster Nodes Metrics'. Below these, the 'Scheduler Metrics' section shows a table of running applications. One application, 'application\_1515325737805\_0006', is highlighted with a red box. It is a Hadoop job named 'streamjob7022129202320877187.jar' running on the 'default' queue.

ID	User	Name	Application Type	Queue	Application Priority
application_1515325737805_0006	hadoop	streamjob7022129202320877187.jar	MAPREDUCE	default	0
application_1515325737805_0005	hadoop	MyJob	MAPREDUCE	default	0

- Si seleccionamos History, podemos ver el número de mappers y reducers



The screenshot shows the 'Job History' page for the job 'streamjob7022129202320877187.jar'. It displays job details such as 'User Name: hadoop', 'Queue: default', and 'State: SUCCEEDED'. Below the job details, the 'ApplicationMaster' section shows a table of task attempts. The 'Map' task type has 2 attempts, and the 'Reduce' task type has 1 attempt. The 'Diagnosics' section shows 'Average Map Time: 1mins, 17sec' and 'Average Reduce Time: 25sec'. The 'Task Type' table has red boxes around the '2' for Map and '1' for Reduce, and the 'Attempt Type' table has red boxes around the '2' for Maps and '1' for Reduces.

Task Type	Total	Complete
Map	2	2
Reduce	1	1

Attempt Type	Failed	Killed	Successful
Maps	0	0	2
Reduces	0	0	1

- Otro ejemplo. Si solo queremos contar ocurrencias. No necesitamos Reducer y le decimos que solo hay mapper, y ejecutamos el mapper con el comando "wc -l"

```
hadoop jar /opt/hadoop/share/hadoop/tools/lib/hadoop-streaming-2.9.0.jar -D mapred.reduce.tasks=0 -input /practicass/cite75_99.txt -output /resultado9 -mapper 'wc -l'
```

```
packageJobJar: [/tmp/hadoop-unjar2795715086677832161/] []
/tmp/streamjob6020154308636157887.jar tmpDir=null
```

```
18/01/07 15:16:47 INFO client.RMPProxy: Connecting to ResourceManager at
nodo1/192.168.56.101:8032
18/01/07 15:16:48 INFO client.RMPProxy: Connecting to ResourceManager at
nodo1/192.168.56.101:8032
18/01/07 15:16:49 INFO mapred.FileInputFormat: Total input files to process : 1
18/01/07 15:16:49 INFO net.NetworkTopology: Adding a new node: /default-
rack/192.168.56.123:50010
18/01/07 15:16:49 INFO net.NetworkTopology: Adding a new node: /default-
rack/192.168.56.103:50010
18/01/07 15:16:49 INFO net.NetworkTopology: Adding a new node: /default-
rack/192.168.56.132:50010
18/01/07 15:16:49 INFO net.NetworkTopology: Adding a new node: /default-
rack/192.168.56.125:50010
18/01/07 15:16:50 INFO mapreduce.JobSubmitter: number of splits:2
18/01/07 15:16:50 INFO Configuration.deprecation: mapred.reduce.tasks is
deprecated. Instead, use mapreduce.job.reduces
```

- Podemos ver el resultado en el directorio /resultado9
- Como le hemos quitado el reducer y tenemos dos bloques en el fichero (recordemos que cada bloque es de 128Mb y nuestro fichero ocupa 200MB), nos aparecen dos ficheros con el número de líneas en cada bloque

**hdfs dfs -ls /resultado9**

Found 3 items

```
-rw-r--r--      3  hadoop  supergroup          0  2018-01-07  15:17
/resultado9/_SUCCESS
-rw-r--r--      3  hadoop  supergroup          9  2018-01-07  15:17 /resultado9/part-
00000
-rw-r--r--      3  hadoop  supergroup          9  2018-01-07  15:17 /resultado9/part-
00001
```

**hdfs dfs -cat /resultado9/part-00001**

8264198

**hdfs dfs -cat /resultado9/part-00000**

8258241

- Como siempre, podemos ver en la página Web el resultado