

# Prácticas BigData

## 1. Lanzar procesos con Python

- Con vi o cualquier otro editor, creamos el siguiente programa Python y lo llamamos “pymap.py”
- El programa va extrayendo las palabras del fichero y añadiendo un 1 a cada una de ellas, siguiendo el patrón map reduce

```
#!/usr/bin/env python

import sys

for line in sys.stdin:
    line = line.strip()
    words = line.split()
    for word in words:
        print '%s\t1' % word
```

- Ahora creamos el programa para el reduce, que permite realizar la suma total de palabras. Lo llamamos pyreduce.py

```
#!/usr/bin/env python

from operator import itemgetter
import sys

last_word = None
last_count = 0
cur_word = None

for line in sys.stdin:
    line = line.strip()

    cur_word, count = line.split('\t', 1)

    count = int(count)
```

---

```

if last_word == cur_word:
    last_count += count
else:
    if last_word:
        print '%s\t%s' % (last_word, last_count)
    last_count = count
    last_word = cur_word

if last_word == cur_word:
    print '%s\t%s' % (last_word, last_count)

```

- Lanzamos el proceso a través de hadoop streaming

```
hadoop jar /opt/hadoop/share/hadoop/tools/lib/hadoop-streaming-2.9.0.jar
-file pymap.py -mapper pymap.py -file pyreduce.py -reducer pyreduce.py -
input /practicas/quijote.txt -output /resultado4
```

18/01/07 10:08:12 WARN streaming.StreamJob: -file option is deprecated, please use generic option -files instead.

packageJobJar: [pymap.py, pyreduce.py, /tmp/hadoop-unjar2186090198010276252/] [] /tmp/streamjob8257554939186511413.jar tmpDir=null

18/01/07 10:08:15 INFO client.RMProxy: Connecting to ResourceManager at localhost/127.0.0.1:8032

18/01/07 10:08:15 INFO client.RMProxy: Connecting to ResourceManager at localhost/127.0.0.1:8032

18/01/07 10:08:19 INFO mapred.FileInputFormat: Total input files to process : 1

18/01/07 10:08:20 INFO mapreduce.JobSubmitter: number of splits:2

18/01/07 10:08:21 INFO Configuration.deprecation: yarn.resourcemanager.system-metrics-publisher.enabled is deprecated. Instead, use yarn.system-metrics-publisher.enabled

18/01/07 10:08:22 INFO mapreduce.JobSubmitter: Submitting tokens for job: job\_1515272962334\_0006

18/01/07 10:08:24 INFO impl.YarnClientImpl: Submitted application application\_1515272962334\_0006

18/01/07 10:08:24 INFO mapreduce.Job: The url to track the job: http://nodo1:8088/proxy/application\_1515272962334\_0006/

18/01/07 10:08:24 INFO mapreduce.Job: Running job: job\_1515272962334\_0006

.....

.....

- Vemos que genera una salida similar al programa hecho en Java
- También podemos ver en la página Web que el tipo de programa lanzado es Map Reduce

Cluster Metrics					
Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Containers Pending
4	0	0	4	0	0
Cluster Nodes Metrics					
Active Nodes	Decommissioning Nodes			Decommissioned Nodes	
1	0			0	
Scheduler Metrics					
Scheduler Type	Scheduling Resource Type			Minimum	
Capacity Scheduler	[MEMORY]			<memory:1024, vCores:1>	
Show 20 entries					
ID	User	Name	Application Type	Queue	Application Priority
application_1515272962334_0005	hadoop	streamjob9196474600754661752.jar	MAPREDUCE	default	0
application_1515272962334_0004	hadoop	WebLogMessageSizeAggregator	MAPREDUCE	default	0
application_1515272962334_0003	hadoop	word count	MAPREDUCE	default	0

- En el capítulo del cluster veremos algunos ejemplos más de Python y otros entornos de Streaming