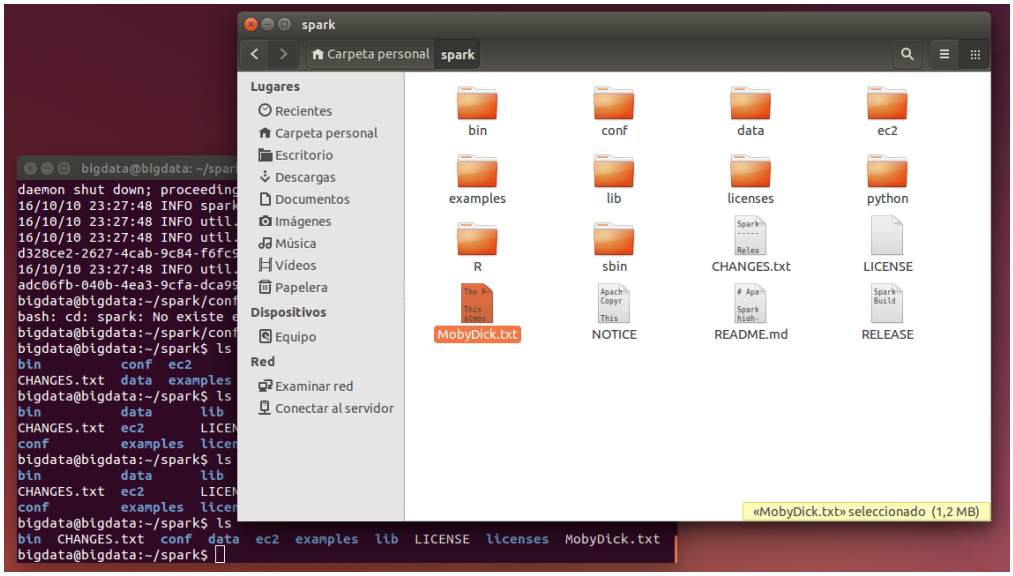
EJERCICIOS DE SPARK

1. Recuperar el libro de Moby Dick del proyecto gutenberg

http://www.gutenberg.org/cache/epub/2701/pg2701.txt

MobyDick.txt

```
men were tossed from out of the rocking boat, was dropped astern. So,
floating on the margin of the ensuing scene, and in full sight of it,
when the halfspent suction of the sunk ship reached me, I was then,
but slowly, drawn towards the closing vortex. When I reached it, it had
subsided to a creamy pool. Round and round, then, and ever contracting
towards the button-like black bubble at the axis of that slowly wheeling
circle, like another Ixion I did revolve. Till, gaining that vital
centre, the black bubble upward burst; and now, liberated by reason of
its cunning spring, and, owing to its great buoyancy, rising with great
force, the coffin life-buoy shot lengthwise from the sea, fell over, and
floated by my side. Buoyed up by that coffin, for almost one whole day
and night, I floated on a soft and dirgelike main. The unharming sharks,
they glided by as if with padlocks on their mouths; the savage sea-hawks
sailed with sheathed beaks. On the second day, a sail drew near, nearer,
and picked me up at last. It was the devious-cruising Rachel, that in
her retracing search after her missing children, only found another
orphan.




End of Project Gutenberg's Moby Dick; or The Whale, by Herman Melville

*** END OF THIS PROJECT GUTENBERG EBOOK MOBY DICK; OR THE WHALE ***

***** This file should be named 2701.txt or 2701.zip *****
This and all associated files of various formats will be found in:
        http://www.gutenberg.org/2/7/0/2701/

Produced by Daniel Lazarus and Jonesey

Updated editions will replace the previous one--the old editions
will be renamed.

Creating the works from public domain print editions means that no
one owns a United States copyright in these works, so the Foundation
(and you!) can copy and distribute it in the United States without
permission and without paying copyright royalties.  Special rules,
set forth in the General Terms of Use part of this license, apply to
copying and distributing Project Gutenberg-tm electronic works to
protect the PROJECT GUTENBERG-tm concept and trademark.  Project
```



2. Crear una carpeta ejercicioSpark en HDFS

Primero arranco los demonios:

start-dfs.sh

jps

```
bigdata@bigdata: ~/spark
conf         examples  licenses  NOTICE         README.md
bigdata@bigdata:~/spark$ ls
bin  CHANGES.txt  conf  data  ec2  examples  lib  LICENSE  licenses  MobyDick.txt
bigdata@bigdata:~/spark$ jps
3187 Jps
bigdata@bigdata:~/spark$ start-dfs.sh
16/10/10 23:49:16 WARN util.NativeCodeLoader: Unable to load native-hadoop library
for your platform... using builtin-java classes where applicable
Starting namenodes on [localhost]
localhost: starting namenode, logging to /home/bigdata/hadoop/logs/hadoop-bigdata-n
amenode-bigdata.out
localhost: starting datanode, logging to /home/bigdata/hadoop/logs/hadoop-bigdata-d
atanode-bigdata.out
Starting secondary namenodes [0.0.0.0]
0.0.0.0: starting secondarynamenode, logging to /home/bigdata/hadoop/logs/hadoop-bi
gdata-secondarynamenode-bigdata.out
16/10/10 23:49:36 WARN util.NativeCodeLoader: Unable to load native-hadoop library
for your platform... using builtin-java classes where applicable
bigdata@bigdata:~/spark$ jps
3810 Jps
3343 NameNode
3501 DataNode
3701 SecondaryNameNode
bigdata@bigdata:~/spark$
```

Ahora creo la carpeta en hdfs:

```
bigdata@bigdata:~/spark$ hdfs dfs -mkdir /ejercicioSpark
16/10/10 23:56:41 WARN util.NativeCodeLoader: Unable to load native-hadoop library
for your platform... using builtin-java classes where applicable
bigdata@bigdata:~/spark$
```

3. Subir el fichero a HDFS a la carpeta anterior
hdfs dfs -put MobyDick.txt /ejercicioSpark

```
bigdata@bigdata:~/spark$ hdfs dfs -put MobyDick.txt /ejercicioSpark
16/10/11 00:08:19 WARN util.NativeCodeLoader: Unable to load native-hadoop library
for your platform... using builtin-java classes where applicable
bigdata@bigdata:~/spark$
```

4. Comprobar que el libro está correctamente subido
cat MobyDick.txt

```
bigdata@bigdata: ~/spark
Section 5.  General Information About Project Gutenberg-tm electronic
works.

Professor Michael S. Hart is the originator of the Project Gutenberg-tm
concept of a library of electronic works that could be freely shared
with anyone.  For thirty years, he produced and distributed Project
Gutenberg-tm eBooks with only a loose network of volunteer support.


Project Gutenberg-tm eBooks are often created from several printed
editions, all of which are confirmed as Public Domain in the U.S.
unless a copyright notice is included.  Thus, we do not necessarily
keep eBooks in compliance with any particular paper edition.


Most people start at our Web site which has the main PG search facility:

     http://www.gutenberg.org

This Web site includes information about Project Gutenberg-tm,
including how to make donations to the Project Gutenberg Literary
Archive Foundation, how to help produce our new eBooks, and how to
subscribe to our email newsletter to hear about new eBooks.
bigdata@bigdata:~/spark$
```

5. Acceder a pyspark

pyspark

```
16/10/11 00:20:48 INFO server.AbstractConnector: Started SelectChannelConnector@0.0
.0.0:4040
16/10/11 00:20:48 INFO util.Utils: Successfully started service 'SparkUI' on port 4
040.
16/10/11 00:20:48 INFO ui.SparkUI: Started SparkUI at http://10.0.2.15:4040
16/10/11 00:20:48 INFO executor.Executor: Starting executor ID driver on host local
host
16/10/11 00:20:48 INFO util.Utils: Successfully started service 'org.apache.spark.n
etwork.netty.NettyBlockTransferService' on port 34210.
16/10/11 00:20:48 INFO netty.NettyBlockTransferService: Server created on 34210
16/10/11 00:20:48 INFO storage.BlockManagerMaster: Trying to register BlockManager
16/10/11 00:20:48 INFO storage.BlockManagerMasterEndpoint: Registering block manage
r localhost:34210 with 517.4 MB RAM, BlockManagerId(driver, localhost, 34210)
16/10/11 00:20:48 INFO storage.BlockManagerMaster: Registered BlockManager
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 1.6.2
      /_/

Using Python version 2.7.6 (default, Mar 22 2014 22:59:56)
SparkContext available as sc, SQLContext available as sqlContext.
>>>
```

6. Indicar los comandos y el resultado de contar el número de líneas que tiene el fichero

textFile = sc.textFile("/ejercicioSpark/MobyDick.txt")

```
>>> textFile = sc.textFile("/ejercicioSpark/MobyDick.txt")
16/10/11 00:25:28 INFO storage.MemoryStore: Block broadcast_0 stored as values in m
emory (estimated size 130.9 KB, free 130.9 KB)
16/10/11 00:25:28 INFO storage.MemoryStore: Block broadcast_0_piece0 stored as byte
s in memory (estimated size 16.0 KB, free 146.8 KB)
16/10/11 00:25:28 INFO storage.BlockManagerInfo: Added broadcast_0_piece0 in memory
 on localhost:34210 (size: 16.0 KB, free: 517.4 MB)
16/10/11 00:25:28 INFO spark.SparkContext: Created broadcast 0 from textFile at Nat
iveMethodAccessorImpl.java:-2
>>>
```

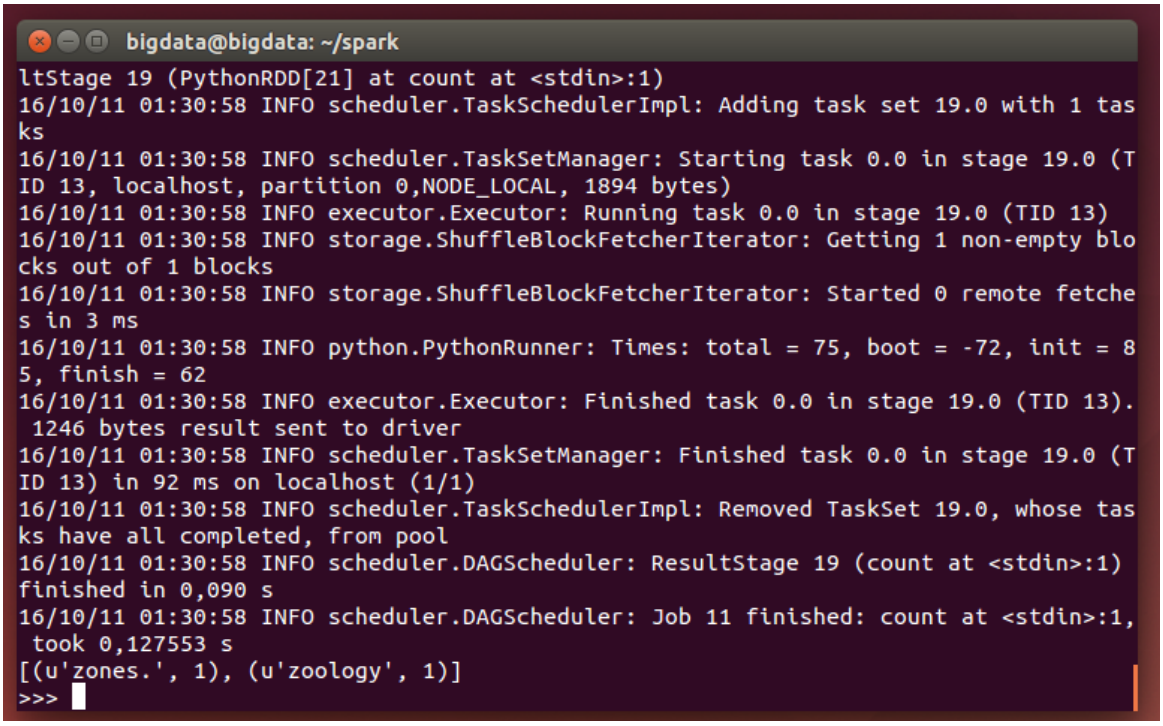
textFile.filter(lambda line: ""in line).count()

```
bigdata@bigdata: ~/spark
16/10/11 00:32:42 INFO Configuration.deprecation: mapred.tip.id is deprecated. Inst
ead, use mapreduce.task.id
16/10/11 00:32:42 INFO Configuration.deprecation: mapred.task.id is deprecated. Ins
tead, use mapreduce.task.attempt.id
16/10/11 00:32:42 INFO Configuration.deprecation: mapred.task.is.map is deprecated.
 Instead, use mapreduce.task.ismap
16/10/11 00:32:42 INFO Configuration.deprecation: mapred.task.partition is deprecat
ed. Instead, use mapreduce.task.partition
16/10/11 00:32:42 INFO Configuration.deprecation: mapred.job.id is deprecated. Inst
ead, use mapreduce.job.id
16/10/11 00:32:43 INFO python.PythonRunner: Times: total = 585, boot = 189, init =
77, finish = 319
16/10/11 00:32:43 INFO executor.Executor: Finished task 0.0 in stage 0.0 (TID 0). 2
125 bytes result sent to driver
16/10/11 00:32:43 INFO scheduler.DAGScheduler: ResultStage 0 (count at <stdin>:1) f
inished in 0,919 s
16/10/11 00:32:43 INFO scheduler.TaskSetManager: Finished task 0.0 in stage 0.0 (TI
D 0) in 880 ms on localhost (1/1)
16/10/11 00:32:43 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 0.0, whose task
s have all completed, from pool
16/10/11 00:32:43 INFO scheduler.DAGScheduler: Job 0 finished: count at <stdin>:1,
took 1,102321 s
22108
>>>
```

7. Ejecutar un word count e indicar ordenadas alfabéticamente las dos últimas palabras que aparecen y el número de repeticiones

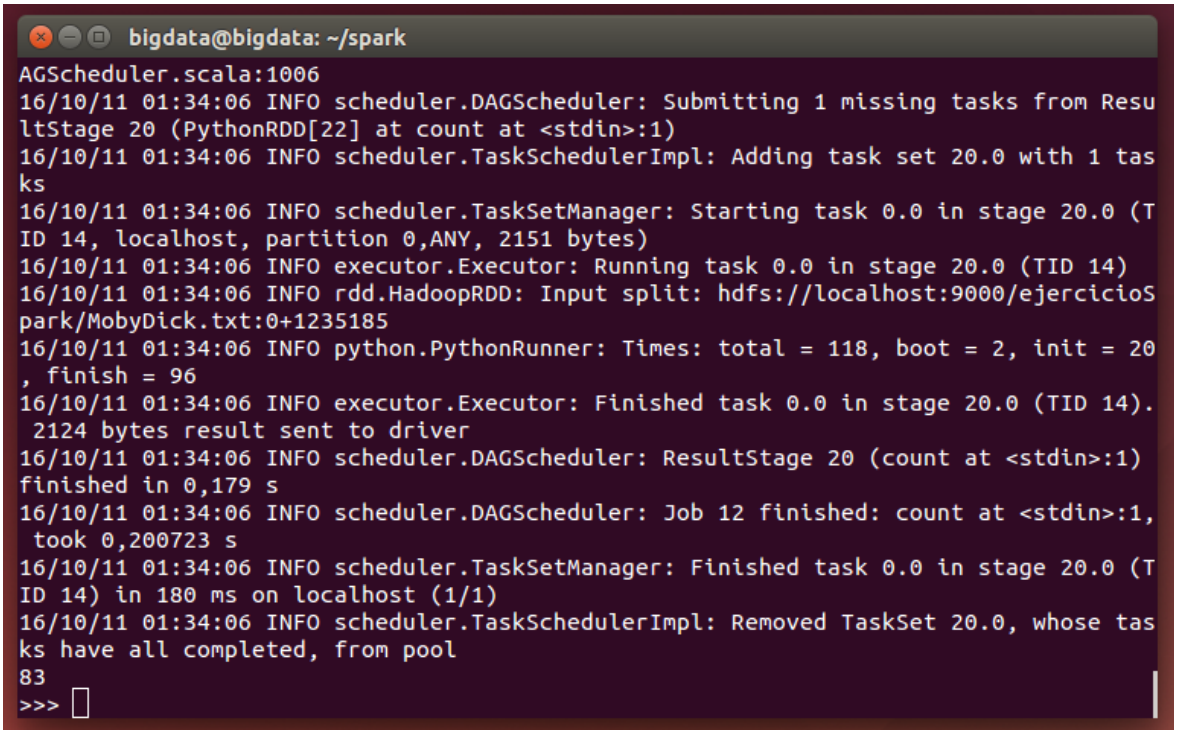
wordCounts = textFile.flatMap(lambda line: line.split()).map(lambda word: (word, 1)).reduceByKey(lambda a, b: a+b)

wordCounts.sortBy(lambda x: x[0]).collect()[wordCounts.count()-2:wordCounts.count()]



8. Indicar las instrucciones y el valor devuelto del número de líneas en las que aparece la palabra Moby en el libro

textFile.filter(lambda line: "Moby"in line).count()



9. Diferencias entre Map Reduce y Spark

La diferencia más importante entre Map Reduce y Spark es que Map Reduce trabaja en el disco y Spark utiliza la memoria y cuando los datos no caben en la memoria puede usar disco. Spark, al usar la memoria es más rápido que MapReduce.
Por como el es modo de funcionamiento, Spark requiere más capacidad de RAM y Map Reduce – más hardware.
Los dos motores de procesamiento también afrontan de forma distinta la tolerancia a fallos. Map reduce usa Task Trackers y Spark usa Resilent Distributes Datasets.