|  |
| --- |
|  |
| **LABORATORIO N°14** |
| Big Data |

|  |
| --- |
| Integrantes: Cristian Álvarez – Jorge González |

Introducción

En este informe se documentará todo lo realizado para el laboratorio 14, esto incluye el paso a paso que se solicita realizar usando las herramientas solicitadas como una MV y Jupyter para realizar el laboratorio, finalizando con la realización de un cuestionario.

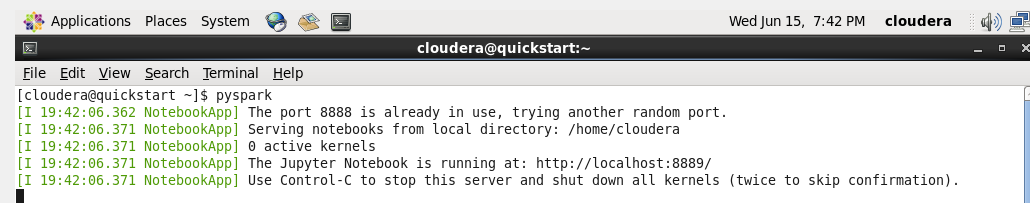
Parte 1. Actividad con Jupyter

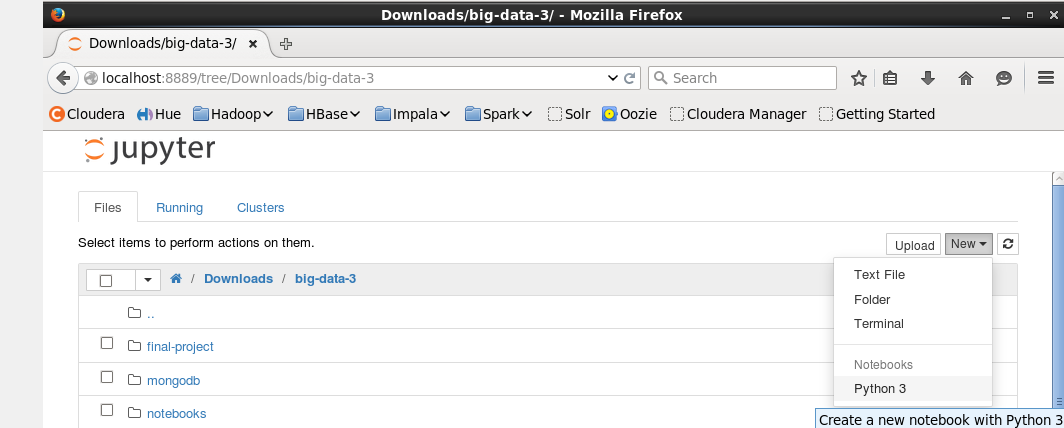
## Paso Previo

Iniciamos en Jupyter Notebook para realizar la actividad y seguido ejecutamos Pyspark con el siguiente comando:

pyspark

En este paso previo deberemos tener abierto el Jupyter en el navegador listo para continuar con los siguientes pasos de la actividad.



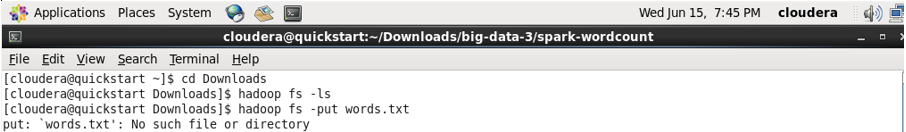


***Paso1.*** Abrimos un terminar en el sistema para ver el contenido de HDFS .

Ejecutamos el comando para ver el contenido:

cd Dowloads

hadoop fs -ls

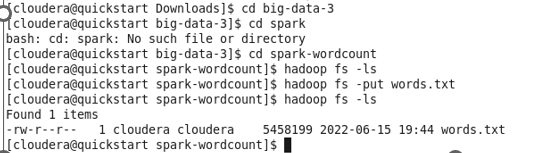


En caso de no poder, hay que ejecutar lo siguientes comandos:

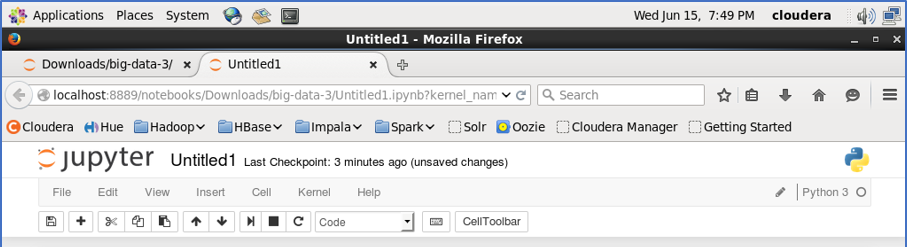
cd big-data-3/spark-wordcount

hadoop fs -put words.txt

hadoop fs -ls



***Paso2.*** En una nueva pestaña en Jupyter Python Notebook .



***Paso3.*** Procedemos a leer el archivo .

Ejecutamos:

lines = sc.textFile(“hdfs:/user/cloudera/words.txt”)

Verificamos que cargo el archivo con el comando:

lines.count()

***Paso4.*** Dividimos en palabras y almacenamos el resultado en un RDD.

Ejecutamos el comando:

words = lines.flatMap(lambda line : line.split(“ ”))

***Paso5.*** Asignamos a cada palabra un numero, iniciando con el 1.

Usamos le comando siguiente:

tuples = words.map(lambda word : (word, 1))

***Paso6.*** Hacemos la sumatorio del conteo de palabras.

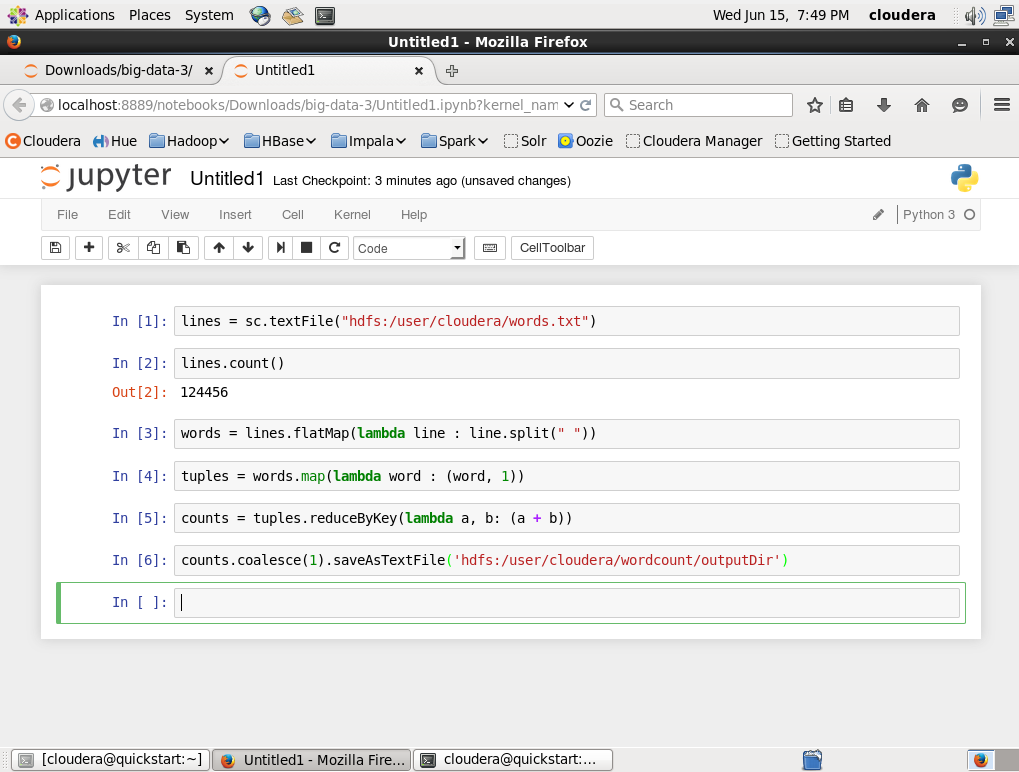
Ejecutamos:

count = tuples.reduceByKey(lambda a, b: (a + b))

***Paso7.*** Excribimos el recuento en un archivo de texto de RDD a HDFS.

Realizamos el siguiente codigo:

counts.coalesce(1).saveAsTextFile(‘hdfs:/user/cloudera/wordcount/outputDir’)



*(Imagen del paso 3,4,5,6,7)*

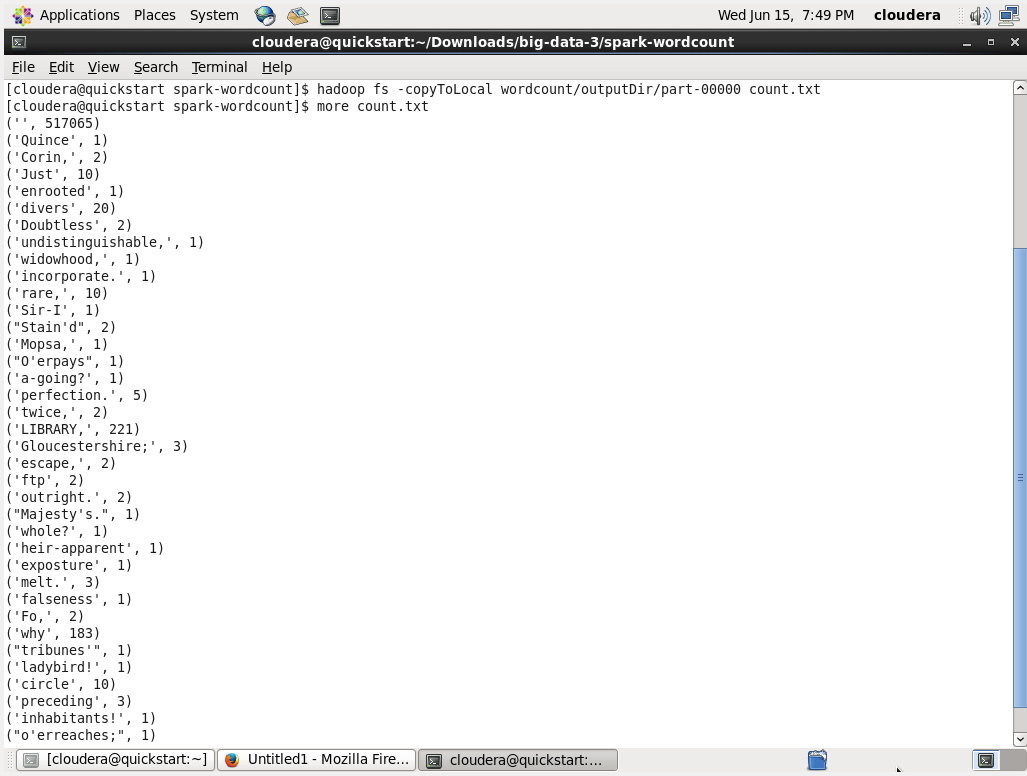
***Paso8.*** Finalmente visualizamos el resultado en el terminal del sistema ejecutando los comandos que se indican.

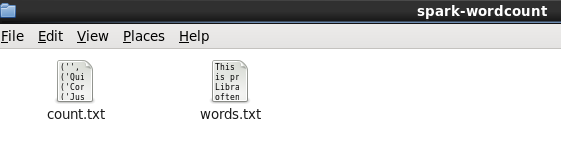
Ejecutamos el comando en el terminal:

hadoop fs -copyToLocal wordcount/outputDir/part-00000 count.txt

Luego para poder ver más:

more count.txt





Paso2. Cuestionario

## Respuestas:

**1.-** What technique does Spark employ for faster processing speed?

1. Excessive Programming Model

**2. In-Memory Processing**

3. HDFS

4. Interactive Shell

**2.-** What does RDD stand for?

1. Reach Distributed Design

2. Resilient Data Dynamics

3. Readily Distributed Data

**4. Resilient Distributed Datasets**

**3.-** What is data-parallelism?

1. Simultaneously processing input data from multiple cores.

2. Having multiple multiple data pipelines at the same time.

3. At each step of the data pipeline, process values simultaneously by using multiple cores.

**4. Running the same function simultaneously for the partitions of a data set on multiple cores.**

**4.-** Of the following, which procedure best generalizes big data procedures such as (but not limitedto) the map reduce process?

1. split->sort->merge

**2. split->do->merge**

3. split ->shuffle and sort->map->reduce

4. split->map->shuffle and sort->reduce

**5.-** What are the three layers for the Hadoop Ecosystem? (Choose 3)

**1. Data Management and Storage**

2. Data Manipulation and Integration

3. Data Creation and Storage

**4. Data Integration and Processing**

**5. Coordination and Workflow Management**

**6.-** What are the 5 key points in order to categorize big data systems?

1. Coordination, Latency, Productivity, Flexibility, Fault Tolerance

2. Execution model, Speed, Scalability, Flexibility, Fault Tolerance

**3. Execution model, Latency, Scalability, Programming Language, Fault Tolerance**

4. Coordination, Latency, Productivity, Speed, Fault Tolerance

**7.-** What is the lambda architecture as shown in ppt?

**1. A type of hybrid data processing architecture.**

2. An architecture that natively supports lambda calculus.

3. A type of swappable data processing layer.

4. A type of architecture that only contains part of the data processing method.

**8.-** Which of the following scenarios is NOT an aggregation operation?

**1. Removing undefined values.**

2. Counting the total number of data.

3. Averaging the total number of data per type.

4. Counting the total number of data per type.

**9.-** What usually happens to data when aggregated as mentioned in lecture?

**1. Data becomes smaller.**

2. Data become organized.

3. Data becomes faster to process.

4. Data becomes personalized.

**10.-** What is K-means clustering?

1. Divide samples using k lines.

2. Classify data by k decisions.

3. Classify data by k actions.

**4. Group samples into k clusters.**

**11.-** Why is Hadoop not a good platform for machine learning as mentioned in ppt? (Choose 4)

**1. Java support only.**

2. Unable to support machine learning.

3. Requires nodes and multiple machines.

**4. Map and Reduce Based Computation.**

5. Too massive.

**6. No interactive shell and streaming.**

**7. Bottleneck using HDFS.**

**12.-** What are the layers (parts) of Spark? (Choose 5)

**1. SparkSQL**

2. Worker Node

**3. MLlib**

**4. Spark Streaming**

5. Spark Graph

**6. Spark Core**

7. Spark RDD

**8. Graphx**

**13.-** What is in-memory processing?

1. Having the input completely in memory.

2. Having the pipeline completely in memory.

3. Having the pipeline completely in disk.

4. Having the input completely in disk.

5. Writing data to disk between pipeline steps.

**6. Writing data to memory between pipeline steps.**

**14.-** What does the following line of code do? words = lines.flatMap(lambda line: line.split(“ “))

1. Each word is merged into lines to be counted later.

2. Each word in each line is counted.

**3. Each line in the document is split up into words.**

4. Each line in the document is split into various Spark partitions.

**15.-** What does the following line of code imply about the state of partitions before the action is performed?

1. words = lines.flatMap(lambda line: line.split(“ “))

**2. Each Spark partition corresponds to a line in the document.**

3. There is only one single partition containing the full document.

4. Each Spark partition corresponds to a word in the document.

**16.-** When the following command is executed, where is the file written and how can it be

accessed? counts.coalesce(1).saveAsTextFile(‘hdfs:/user/cloudera/wordcount/outputDir’)

1. The local file system and through the “hadoop fs” command.

**2. HDFS and through the “hadoop fs” command.**

3. The local file system and through the directory with the “cd” terminal command.

4. HDFS and through the system directory with the “cd” terminal command.

**17.-** What does the number one (1) allow us to do in the following line of code?

tuples = words.map(lambda word: (word,1))

1. The number represents the number of partitions in charge of keeping track of each word.

2. None, completely arbitrary in order to apply an algorithm that requires a tuple.

3. The number represents the number of partitions in charge of counting each line.

**4. Treat each word with a weight of one during the counting process.**