



Módulo - 5



Vicent P. Tortosa Lorenzo

Ismael Torres Boigues

Índice

1. Introducción general
2. Modelo publicación/suscripción
3. Apache Kafka
4. Conceptos Básicos
5. Conclusiones

¿Qué es spark?

- Spark es un sistema de computación de datos basado en Hadoop Map reduce
- Es la herramienta de big data más utilizada para procesar datos (rápido y de propósito general, aunque también dispone de componentes especializados)
- Su principal característica es su característica para procesar datos masivos en paralelo (cluster miles de nodos)
- Apache Spark es una mejora de Hadoop MapReduce. (procesos iterativos en memoria, batch/streaming,...). Procesamiento en memoria
- Proyecto open source




¿Qué es spark?



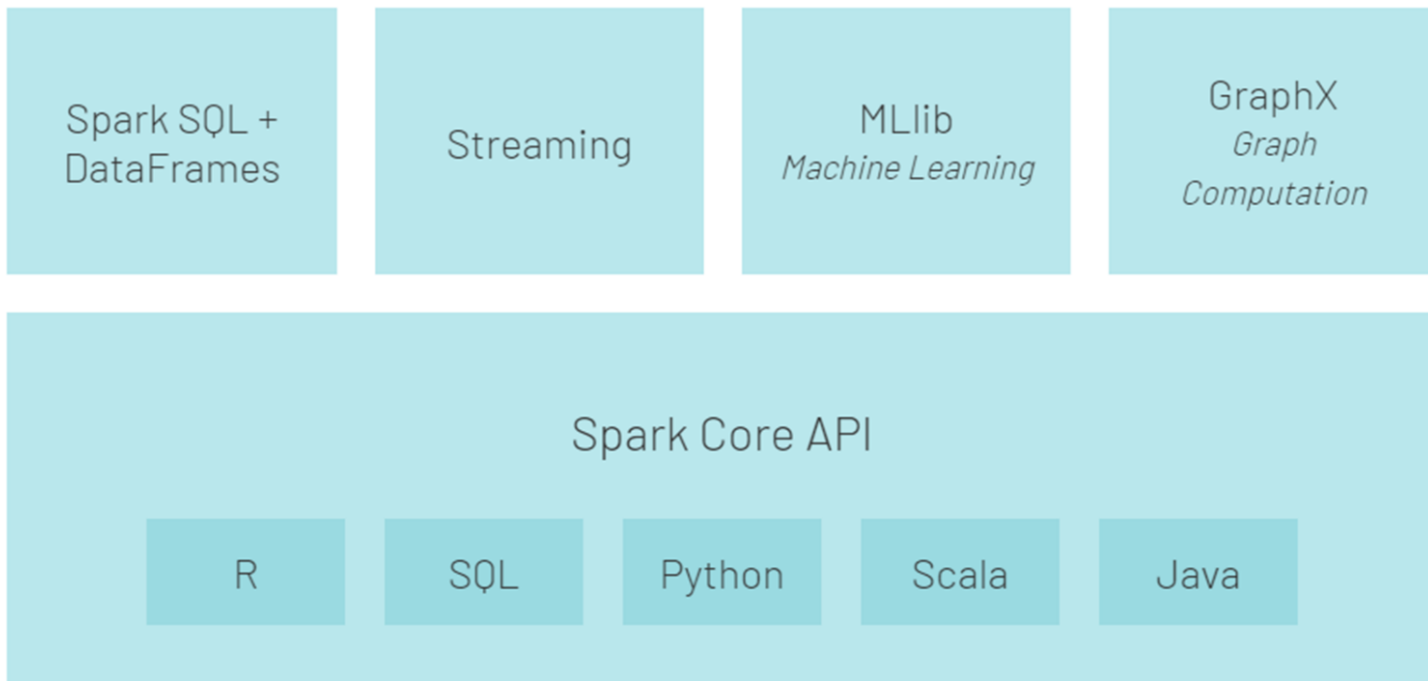
Comparativa Spark y hadoop

Spark permite trabajar en memoria y por tanto su rendimiento es mayor en ciertos casos (hasta 100 más rápido)

permite procesamiento casi en tiempo real (ya que se trabaja con datos en memoria)

Data Flair			Spark  vs  Hadoop MapReduce		
Factors			Spark 		Hadoop MapReduce
Speed			100x times than MapReduce		Faster than traditional system
Written In			Scala		Java
Data Processing			Batch / real-time / iterative / interactive /graph		Batch processing
Ease of Use			Compact & easier than Hadoop		Complex & lengthy
Caching			Caches the data in-memory &		Does not cache data in-memory

Spark- Ecosistema



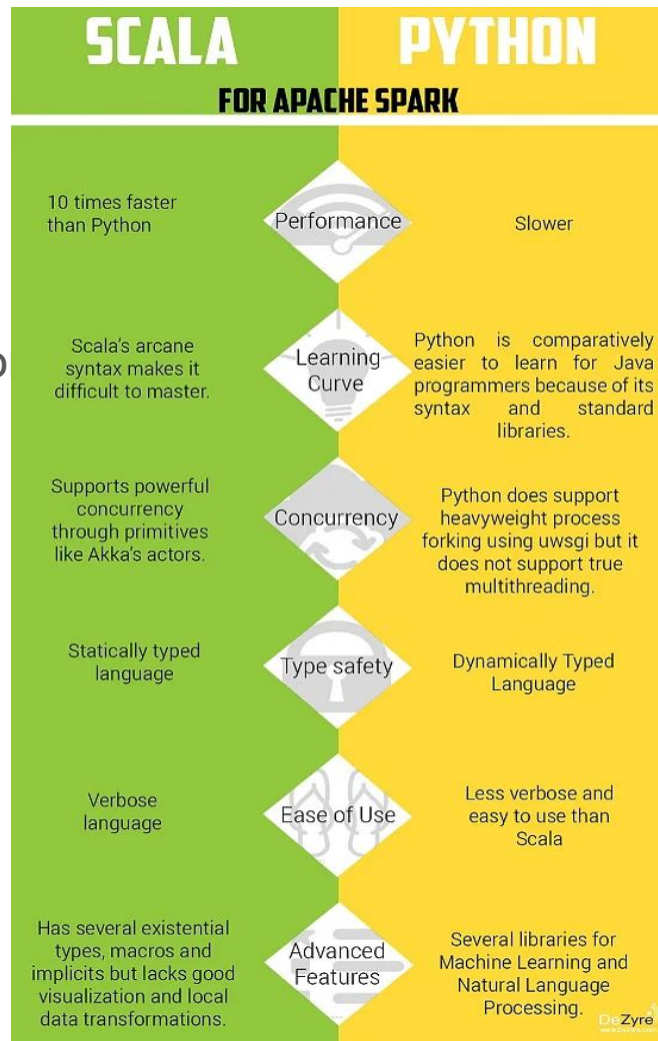
spark streaming



Comparativa python y scala

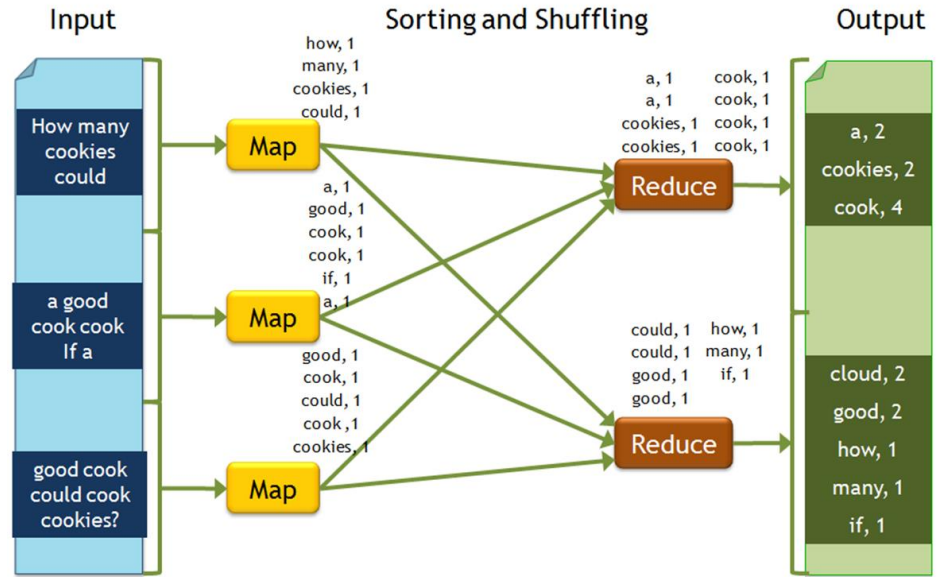
- Scala es hasta 10 veces más rápido

<https://www.projectpro.io/article/scala-vs-python-fo>



Mapreduce

- MapReduce es un modelo de programación para dar soporte a la computación paralela sobre grandes colecciones.
- Su nombre se debe a dos importantes métodos Map y Reduce



By Manaranjan Pradhan

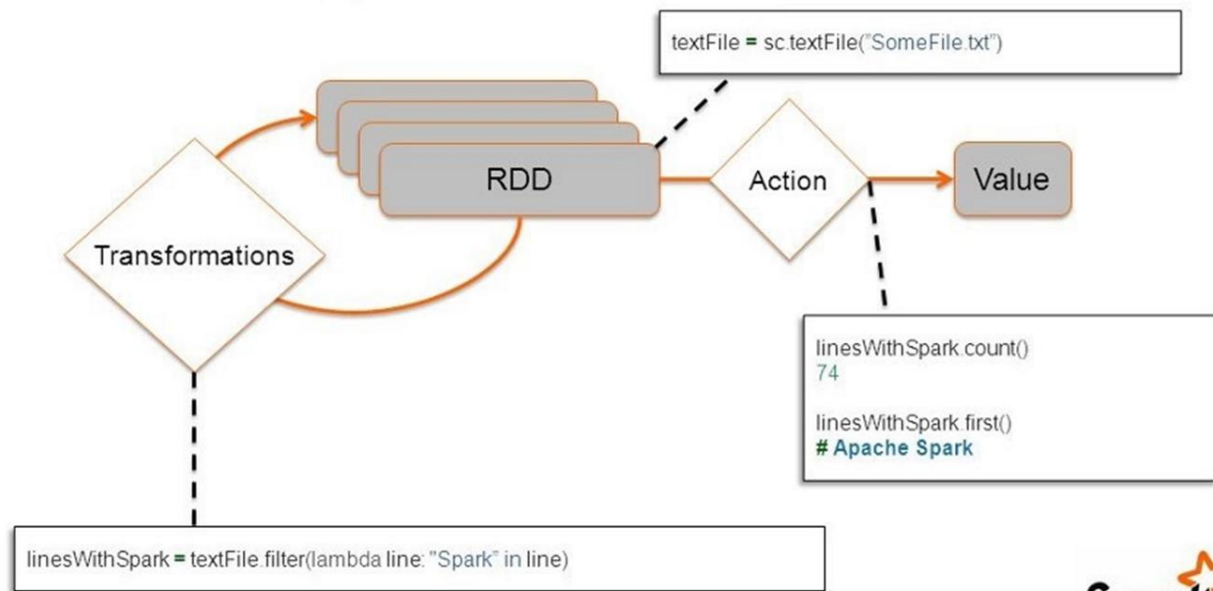


RDD - Resilient Distributed Dataset

- Dispone de dos abstracciones para trabajar con datos
- Data Frames
- RDD
 - Es una **colección distribuida en memoria**. Esto quiere decir que está particionada entre los distintos workers de Spark. “procesamiento en paralelo.
 - Son **inmutables**: cuando transformamos un nuevo RDD realmente estamos creando uno nuevo.
 - **Tolerante a fallos**
 - **Su evaluación es perezosa**. Con los RDD's estamos definiendo un flujo de información, pero no se ejecuta en el momento de definición, sino en el momento en el que se evalúe aplicando una acción sobre el RDD.

RDD- Esquema

Un RDD no solo se encarga de brindar acceso a datos si no también dispone de conjunto de funciones para manipular los datos



RDD. Transformaciones & Acciones

- Transformaciones: definen como cambiará el flujo de información generando un nuevo RDD
 - map, flatmap ,Filter
 - union, intersection, distinct
 - groupByKey
- Acciones: permitirán evaluar un RDD y devolver un resultado. De esta forma se ejecuta todo el flujo de datos definido
 - first
 - count
 - collect
 - saveasertextfile
- <https://spark.apache.org/docs/latest/rdd-programming-guide.html#transformations>

Funciones lambda

- Lambda es una manera de escribir/definir funciones Inline (anónimas) de manera rápida

lambda parámetros: expresión

lambda x, y : x ** y

def cuadrado(x):

return x ** 2

cuad = cuadrado (x)

cuad = lambda x: x ** 2

funcion Lambda Ejemplo

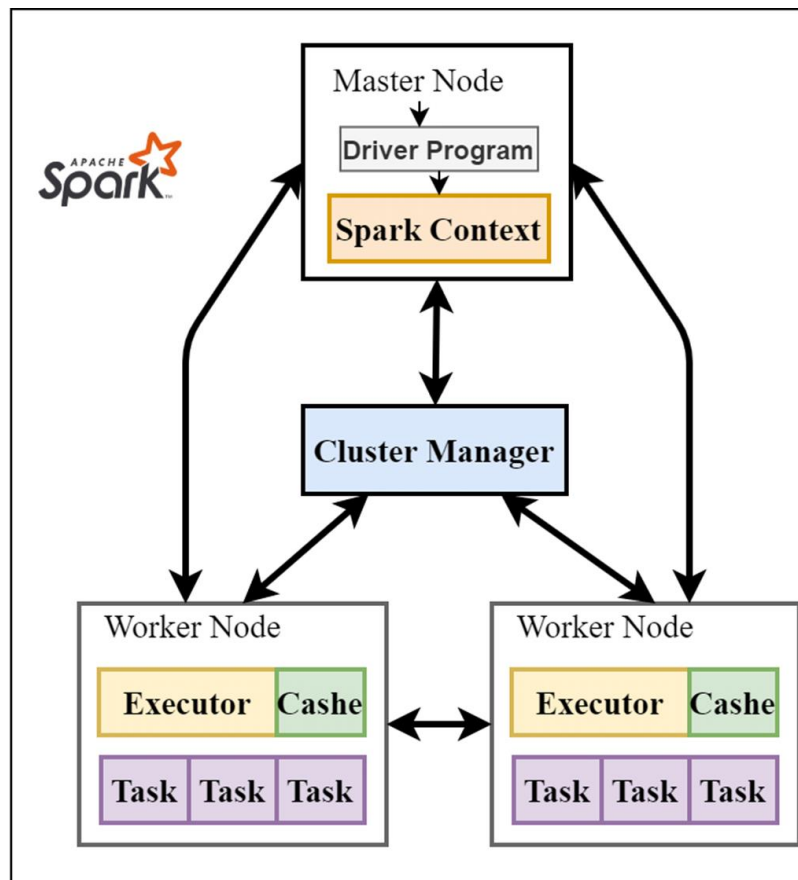
```
words = lines.flatMap(lambda line: line.split(" "))
```

```
pairs = words.map(lambda word: (word, 1))
```

```
wordCounts = pairs.reduceByKey(lambda x, y: x + y)
```

Spark cluster

- El nodo master contiene el “driver program” que dirige la aplicación creando el objeto “Spark context”.
- El Spark context object trabaja con el cluster manager para gestionar los diferentes trabajos. Los nodos Worker se encargan de ejecutar los trabajos y devuelven en resultado al nodo master.
- Cada aplicación Spark consiste en un “driver program” que lanza varias operaciones en paralelo sobre un cluster.
- El driver program contiene la función main de la aplicación y define la distribución de dataset sobre el cluster para que se apliquen operaciones
- El driver program accede a Spark a través de Spark contex.



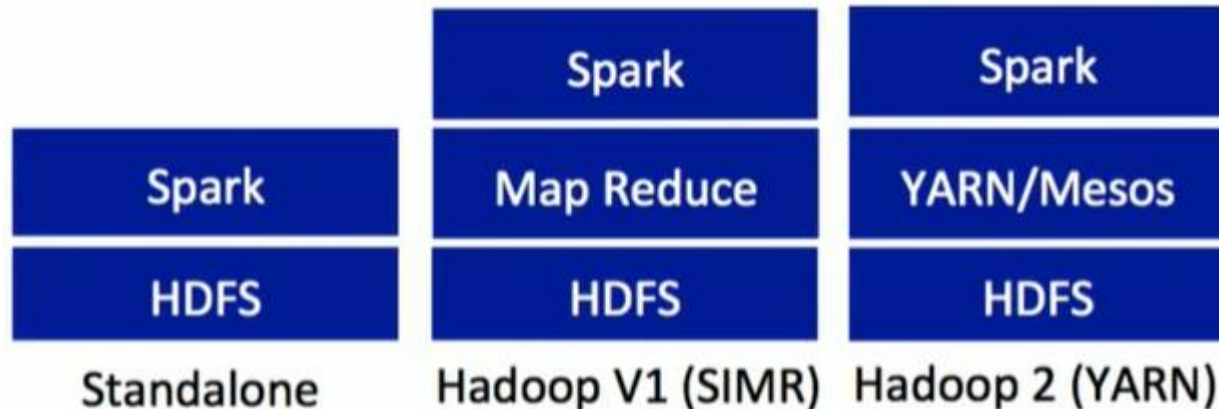
Instalacion

<https://spark.apache.org/downloads.html>

<https://spark.apache.org/docs/latest/spark-standalone.html>

<https://spark.apache.org/docs/latest/running-on-mesos.html>

<https://spark.apache.org/docs/latest/running-on-yarn.html>



Instalacion

Seguir los pasos del documento Instalación spark

Arrancar spark

arrancamos el master

```
start-master.sh
```

Arrancamos un worker

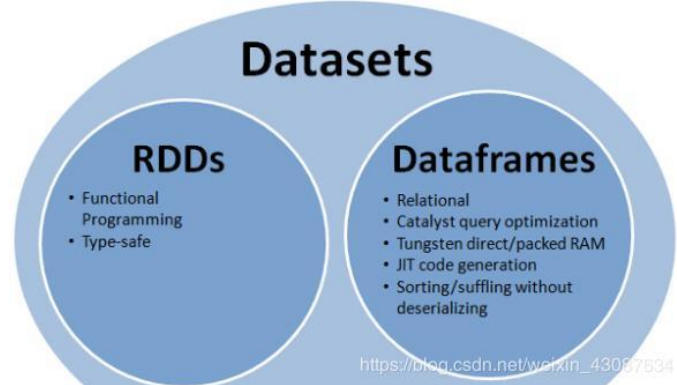
```
start-worker.sh spark://localhost:7077
```

```
start-worker.sh spark://itorres-Virtual-Machine:7077
```

acceso web a spark:

- localhost:8080
- localhost:4040/jobs

Datasets - RDD - Data Frame



- En SparkSQL, Spark nos proporciona dos nuevas abstracciones, DataFrame y DataSet. ¿Cuál es la diferencia entre ellos y RDD? En primer lugar, desde la generación de la versión:
- RDD (Spark1.0) → Dataframe(Spark1.3) → Dataset(Spark1.6)
- Si se dan los mismos datos a estas tres estructuras de datos, darán el mismo resultado después de cada cálculo. La diferencia es su eficiencia de ejecución y método de ejecución.
- En versiones posteriores de Spark, DataSet reemplazará gradualmente RDD y DataFrame como la única interfaz API.

<https://programmerclick.com/article/2504300331/>

