Aplicaciones del aprendizaje profundo en el análisis de grandes volúmenes de datos

Especialización en Ciencia de Datos

Mg. Diego Encinas – Ing. Román Bond

Agenda-Clase 3

Spark

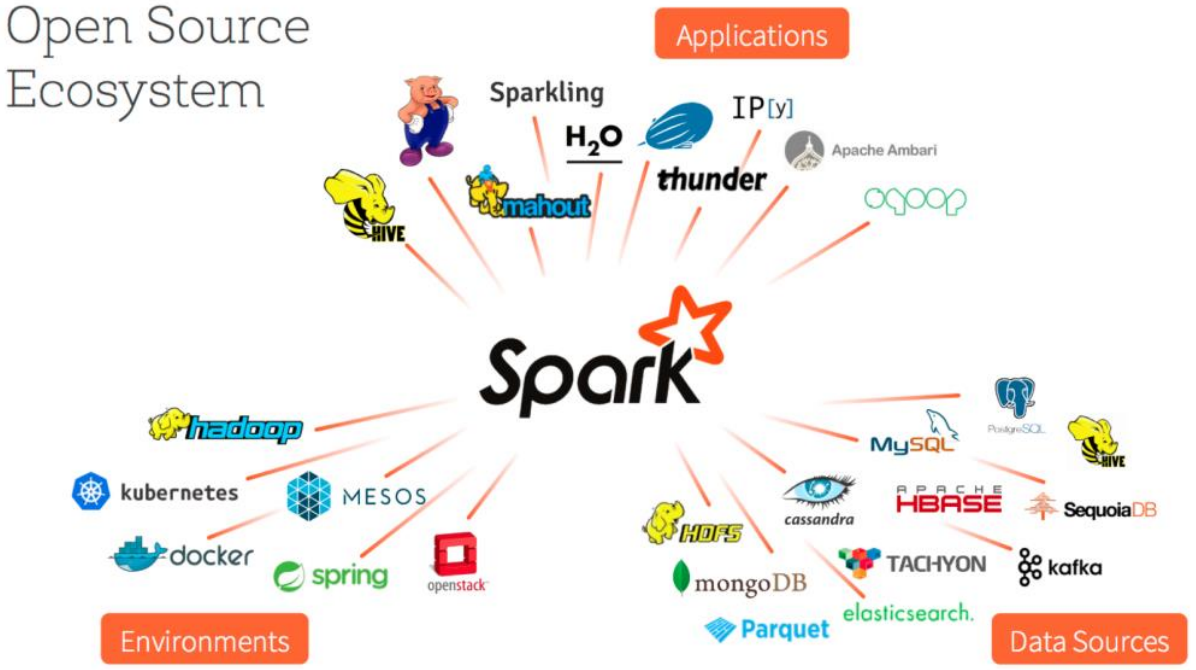
- Introducción
- Arquitectura
- RDDs
- Spark SQL
- Práctica

Spark

- Spark nace en 2009 en los laboratorios de la Universidad de California. Desde 2013 pertenece a la fundación Apache.
- Spark soporta los dos modos de trabajo en Big Data:
  - Consultas en grandes volúmenes de datos (batch processing)
  - Procesamiento de streaming
- Incluye MLlib (librería de machine learning)
- Soporta distintos lenguajes
  - Java, Python, R y Scala

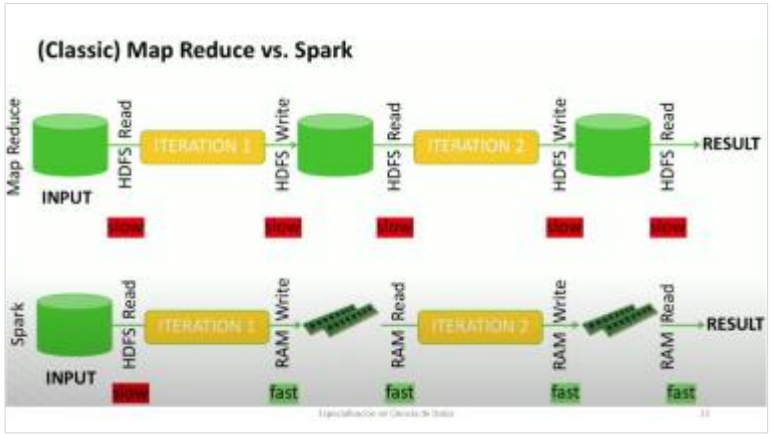
Spark

- Spark fue desarrollado para que trabaje con el HDFS de hadoop, pero también permite la integración con otros medios de almacenamiento:
  - HBase
  - Cassandra
  - MapR-DB
  - MongoDB
  - Amazon's S3.

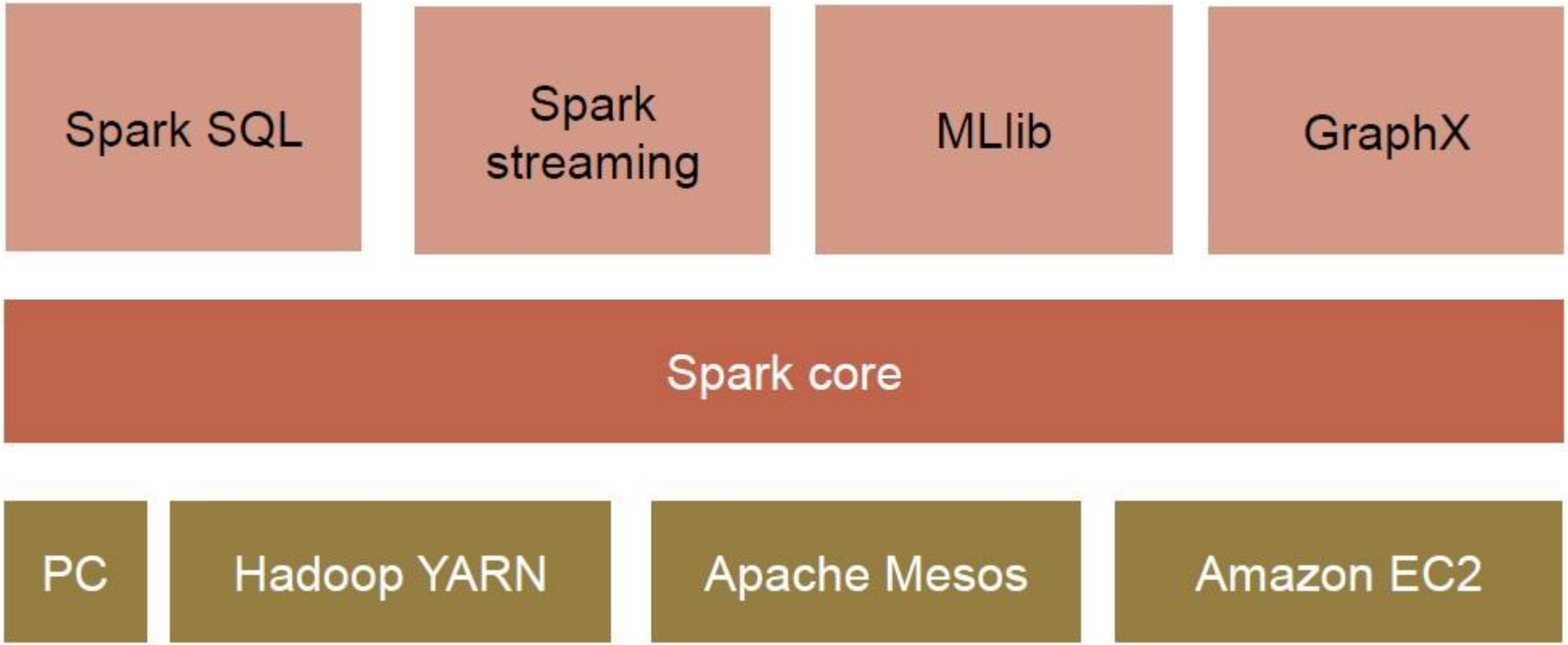


Características de Spark

- Simple
  - Una gran cantidad funciones (API) desarrolladas para trabajar con grande volúmenes de datos
- Velocidad
  - Optimizado para trabajar en RAM
  - Según benchmarks, spark es 100 veces más rápido que Hadoop MapReduce
- Soporte
  - Permite trabajar en varios lenguajes
  - Permite la integración con varios medios de almacenamiento.



Arquitectura de Spark



Arquitectura de Spark

- Spark core
  - Manejo de funciones, tareas de scheduling
- Spark SQL
  - Módulo de trabajo con datos estructurados, soporta Hive, conexiones ODBC, JDBC
- Spark streaming
  - Optimizado para trabajar con Flume (gestor de logs), Kafka (mensajería distribuida)
- MLlib
  - Libería de machine learning. Tiene algoritmos de clasificación, regresión, clustering
- GraphX
  - Librería para el análisis sobre grafos de datos

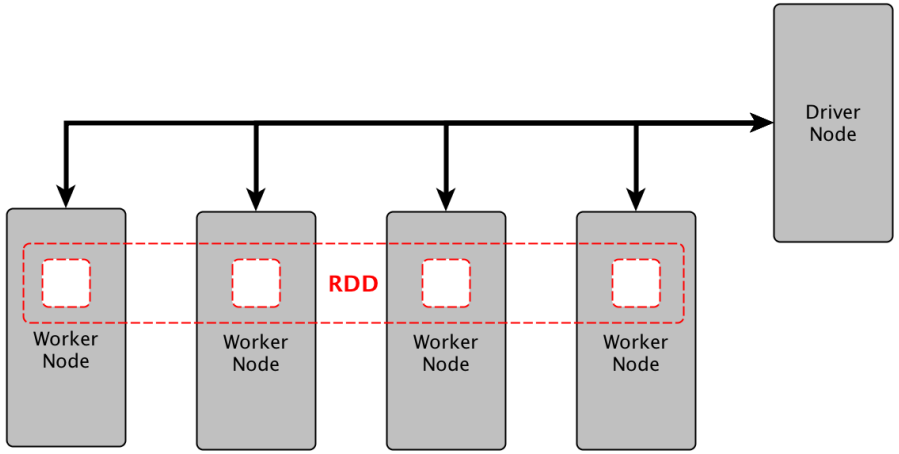
Por lo general una aplicación Spark solo requiere, además de Spark core, una única librería, pero se pueden desarrollar aplicaciones más potentes usando más de una librería.

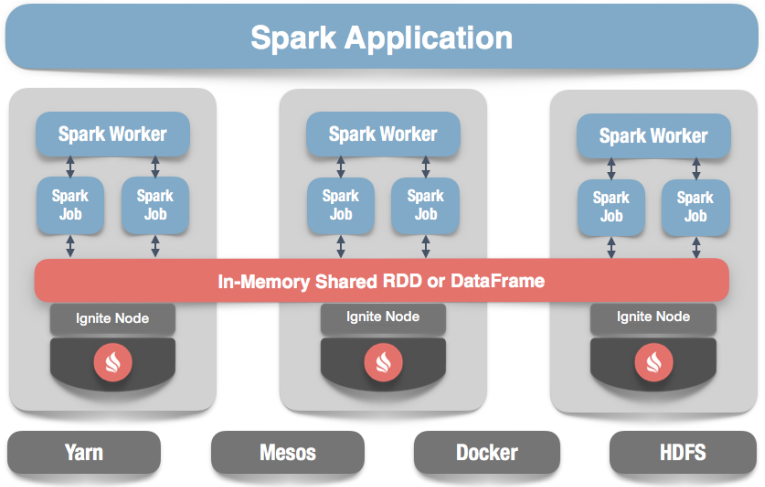
Resilient Distributed Datasets (RDDs)

- Las bases de datos distribuidas y elásticas (RDDs) son parte del núcleo de Spark.
- Están diseñadas para el almacenamiento de los datos en la RAM de manera distribuida en un cluster
  - Eficiencia: paralelización usando varios nodos; permite el replicado de datos
  - Tolerante a fallas: se realiza el tracking de las diferentes operaciones realizadas.

Resilient Distributed Datasets (RDDs)

- Son inmutables
- Son divididas en particiones las cuales se distribuyen entre los nodos de un cluster.
- Se construyen a partir de la lectura del contenido de un archivo o de una base de datos, como así también mediante la paralelización de colecciones de datos.

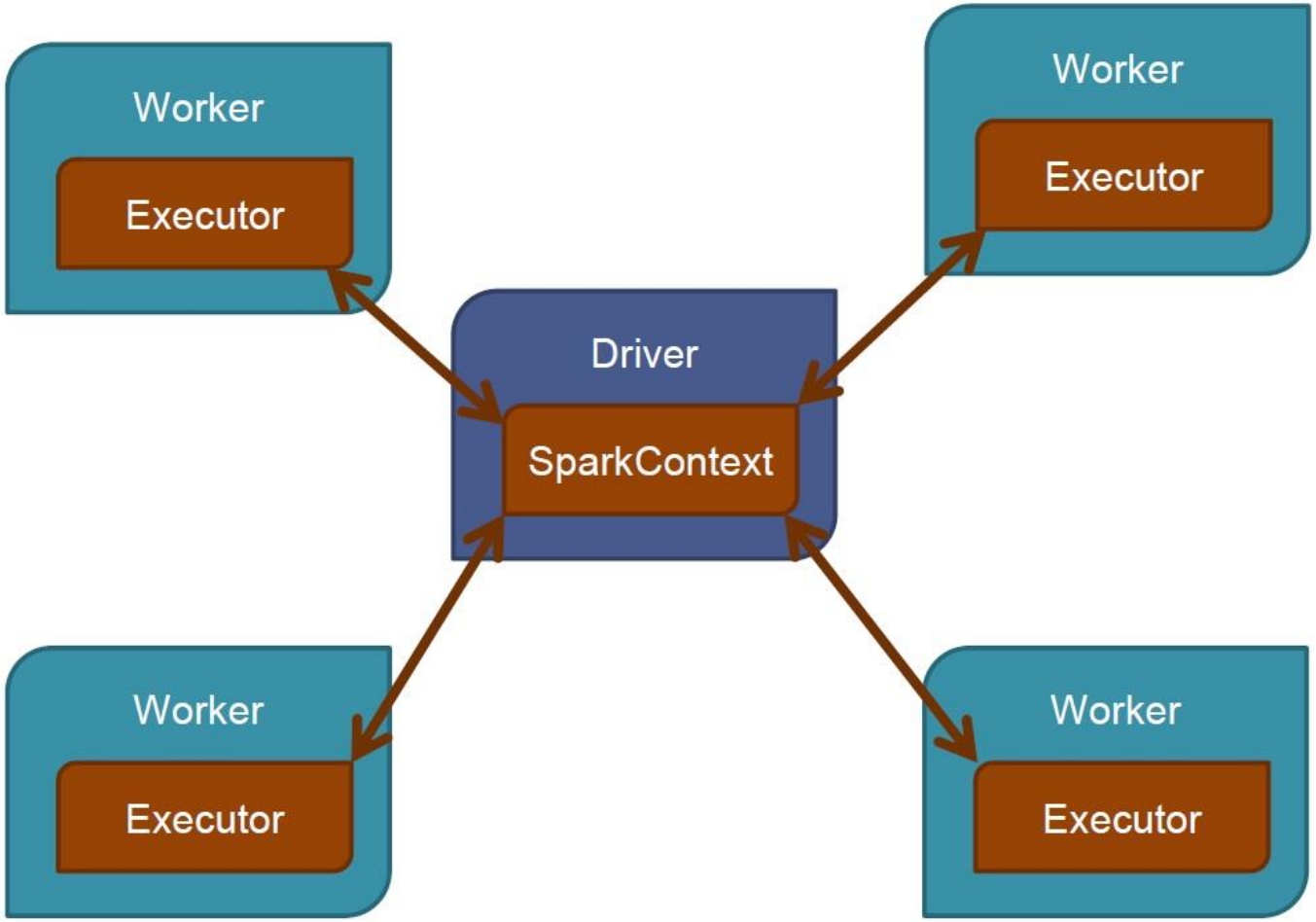




Spark context

- Spark trabaja con el modelo master-slave.
- En el master se ejecuta el módulo principal denominado driver que es el encargado de enviar las operaciones a realizar sobre las RDDs a cada nodo del cluster.
- La conexión con el cluster se realiza mediante un objeto denominado SparkContext.
- El SparkContext crea y maneja las RDDs.

Spark context



Spark shell

- De manera nativa las aplicaciones en Spark se programan en Java.
- Spark además posee dos shells para trabajar con Python y con Scala.
  - Python:
    - Pyspark: El shell crea automáticamente un spark context y lo deja disponible en una variable llamada sc.
  - Scala: spark-shell

Creando el spark shell

- Si se ejecuta un script de python fuera del shell entonces se debe incluir la creación del Spark Context de manera explícita:

```
from pyspark.sql import SparkSession
spark = SparkSession.builder.master("local[*]").getOrCreate()
sc = spark.sparkContext
```

- Se ejecuta con el comando spark-submit:

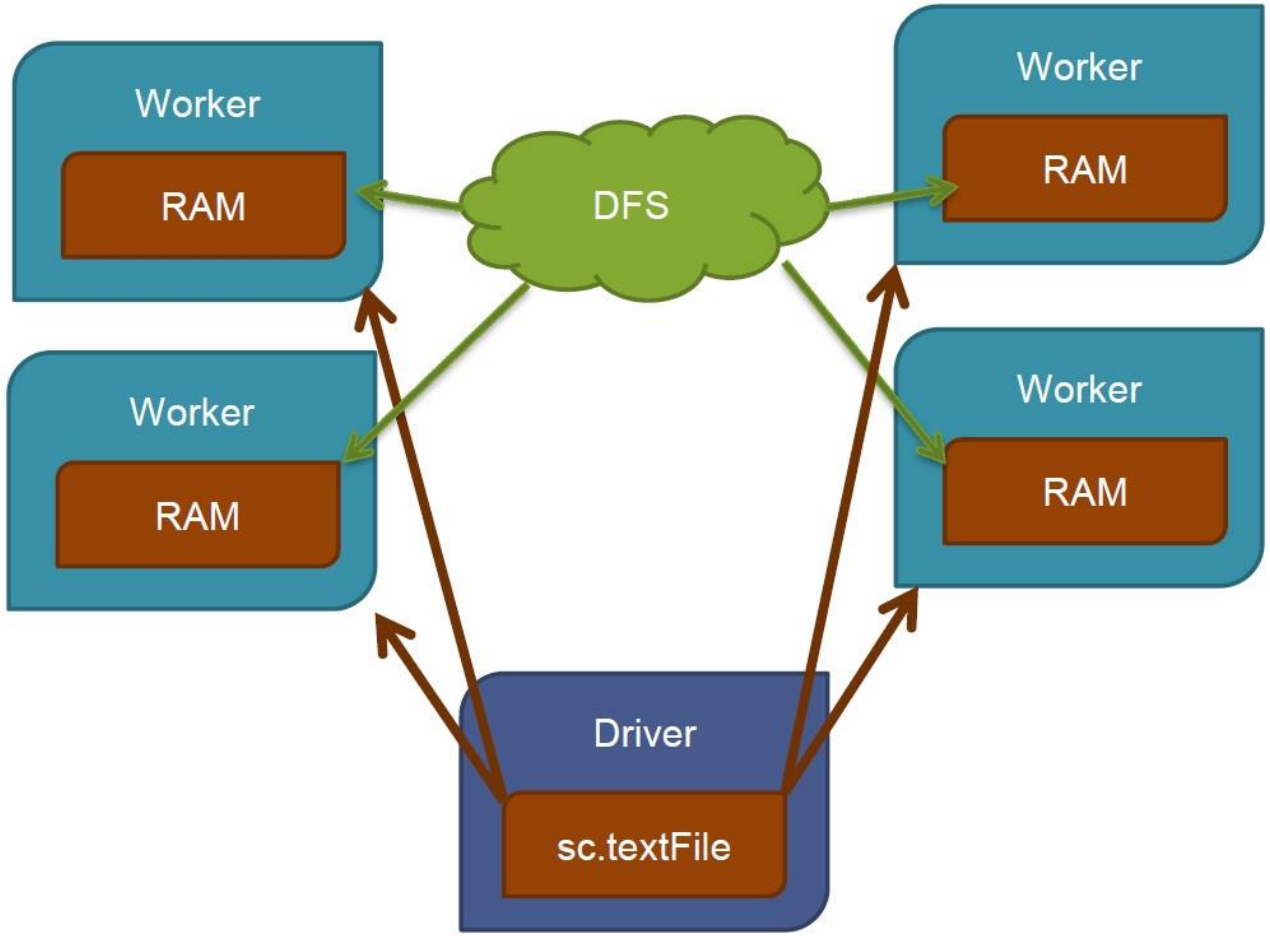
```
spark-submit MyProgram.py
```

- Se escribe un script que se ejecuta de manera secuencial
  - Las operaciones que llamaremos dentro de ese script se ejecutan en paralelo (transparente para el usuario)

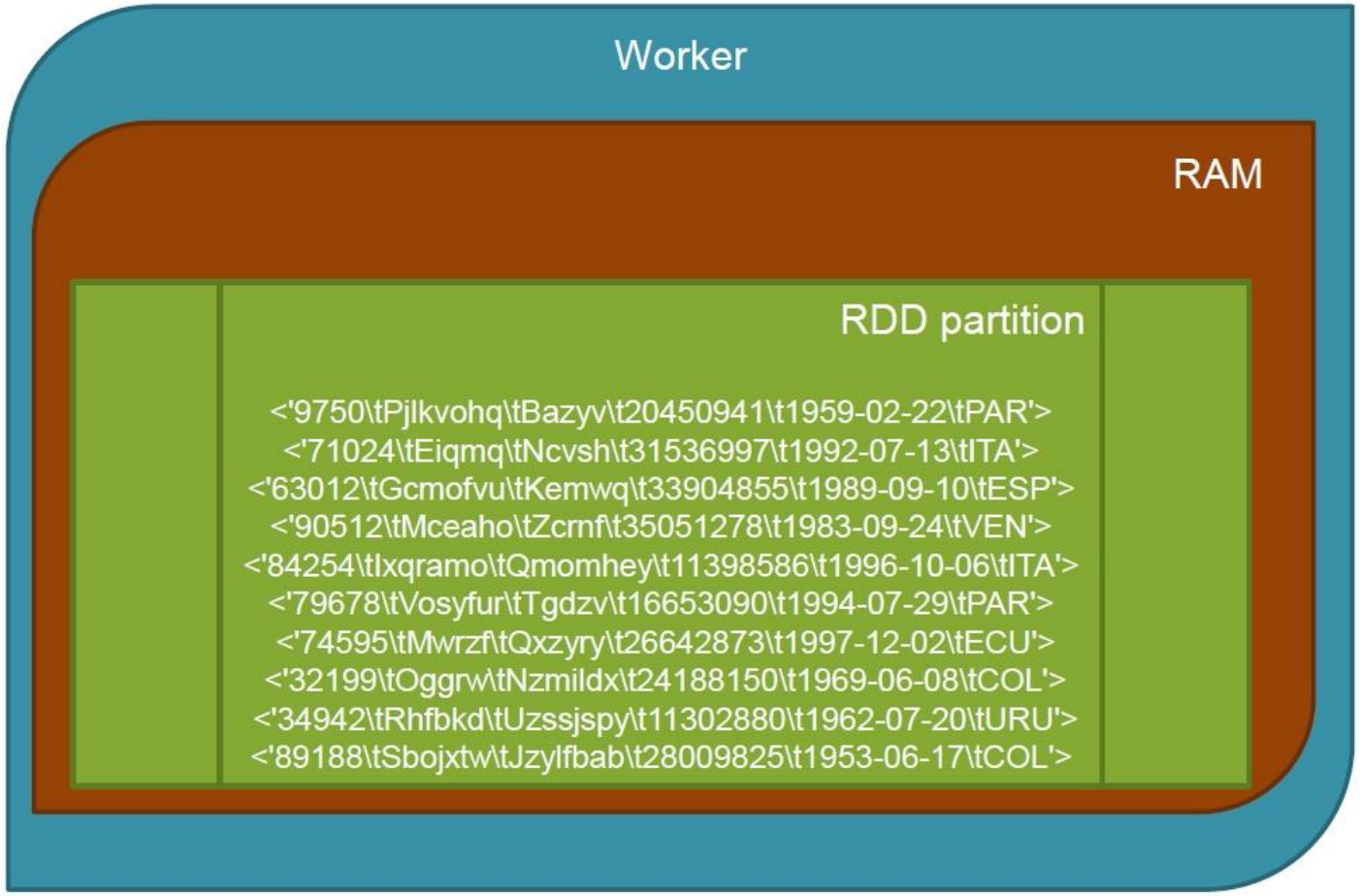
Lectura de archivos

```
clientes = sc.textFile("Clientes")

cajaDeAhorro = sc.textFile("CajasDeAhorro")

prestamos = sc.textFile("Prestamos")
```

Lectura de archivos



Lectura de archivos



Resilient Distributed Datasets (RDDs)

- Cargado los datos en una RDD se pueden realizar dos operaciones básicas:
  - Transformaciones: cambia el RDD original a través de un proceso de mapeo, filtrado, etc.
  - Acciones: tales como el conteo, sumas, los cuales se calculan sobre una RDD sin modificarla
- Las transformaciones y acciones son parte de la API de Spark

Spark SQL

- Spark SQL es una interface que permite usar Spark con datos estructurados, es decir con datos que tienen un esquema.
- Permite el uso del lenguaje SQL para realizar tareas de selección, filtrado, agrupación y joins.
- Es posible cargar datos de diferentes formatos: JSON, Hive, Parquet, ODBC, JDBC, etc.
- Permite una fácil integración entre SQL y Python/Scala/Java
- Spark SQL puede ser integrado con HiveQL.

Spark SQL - DataFrames

- Spark SQL trabaja con una abstracción de las RDD, los DataFrames.
- Un DataFrame es una RDD con esquema, es decir tuplas de datos con nombres y tipo de datos de cada campo.
  - Estos DataFrames son distribuidos
- Se crean mediante el sqlContext.

Preguntas? O …

(Classic) Map Reduce vs. Spark