

Big Data

Data Science Bootcamp

The Bridge



¿Qué es el Big Data?

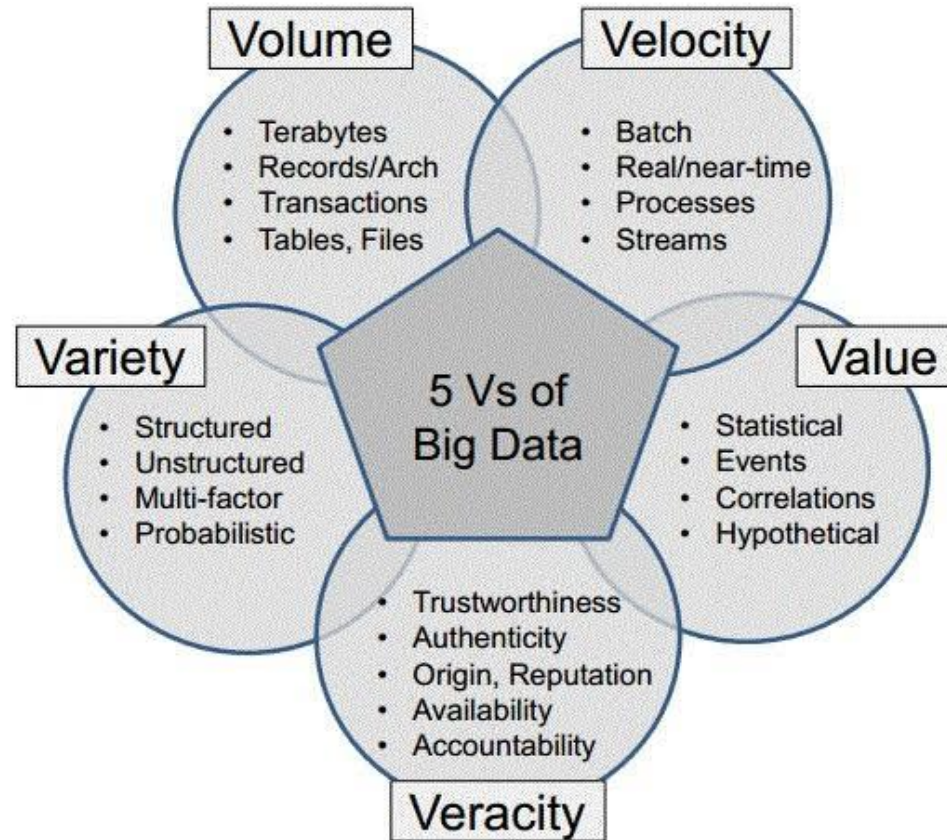
Big Data

Definición

*"Conjuntos de datos o combinaciones de conjuntos de datos cuyo **tamaño (volumen), complejidad (variabilidad) y velocidad de crecimiento (velocidad)** dificultan su captura, gestión, procesamiento o análisis mediante tecnologías y herramientas convencionales, tales como bases de datos relacionales y estadísticas convencionales o paquetes de visualización, dentro del tiempo necesario para que sean útiles."*

Las famosas 3 + 2 V's

Se suman dos más al carro



| ¿Por qué ha surgido esto ahora?



Internet



Generación de datos



Capacidad de cómputo

Más fuentes de datos

Visión 360 del cliente con mayor cantidad de fuentes de datos

Web

Gran cantidad de info a través de las cookies, logs, navegación de los usuarios

Terceros

Posibilidad de compra de información anonimizada o no, a terceros

Redes Sociales

Redes como Twitter, LinkedIn, Facebook

Internet of Things

Uso masivo de sensores sincronizados con la nube y generando datos en real time

Info no estructurada

Datos no estructurados como imágenes, HTML, XML, voz.

Data Lakes



***"Repositorio centralizado** que permite almacenar todos los datos estructurados y no estructurados a cualquier escala. Puede almacenar los datos tal cual, sin tener que estructurarlos primero, y ejecutar diferentes tipos de análisis, desde cuadros de mando y visualizaciones hasta grandes procesamiento de datos, análisis en tiempo real y aprendizaje automático para tomar mejores decisiones."*

Permite centralizar todos los datos en un mismo lugar, sea cual sea su origen.

Es posible que la fuente original del dato esté obsoleta o se haya desactivado. Con este sistema se puede acceder a dicha información.

Todos los datos que llegan al sistema pueden ser normalizados y enriquecidos.

Los datos se preparan de acuerdo a las necesidades del momento

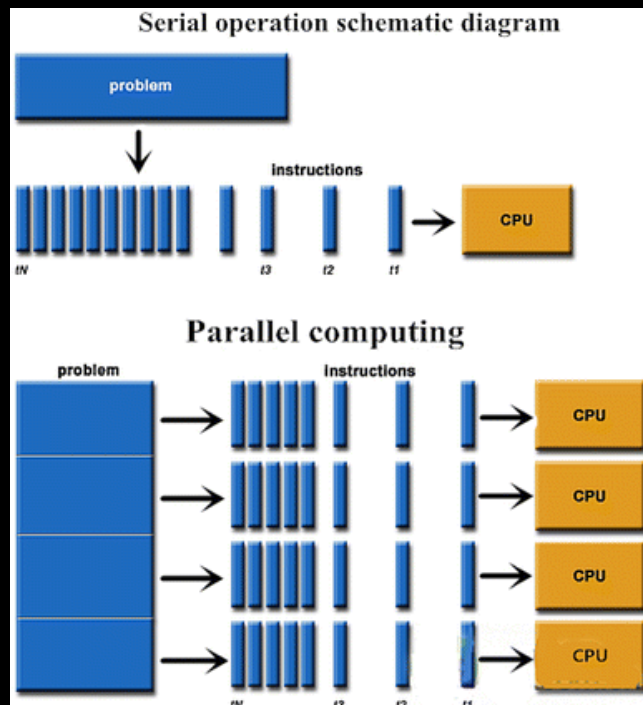
Cualquier usuario autorizado puede acceder a la información y enriquecerla desde cualquier lugar

Tecnologías

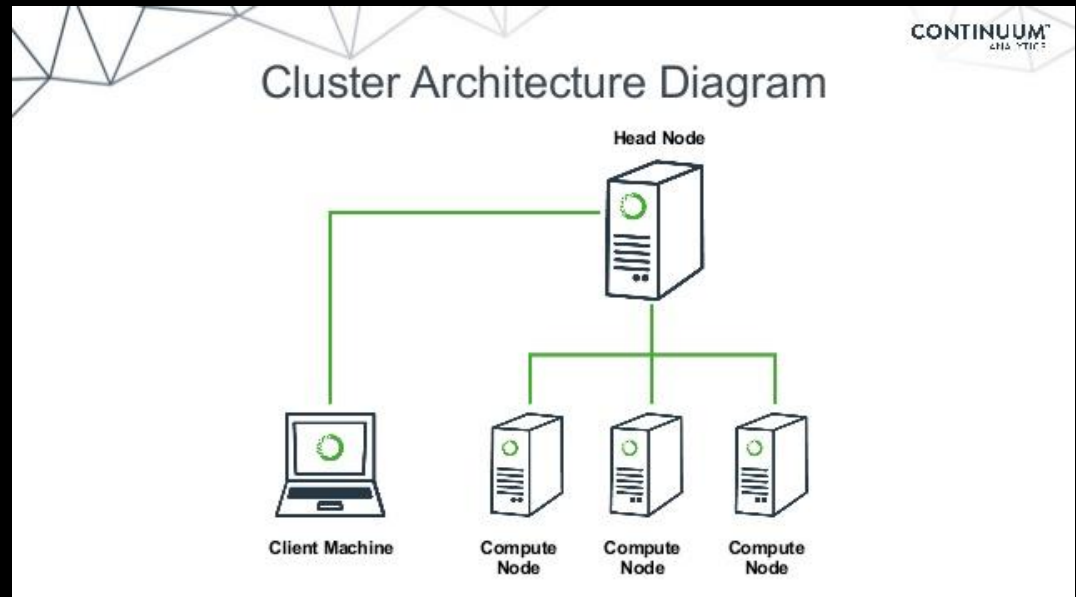
Sistema distribuido

Cuando la CPU no da más de sí

Computación distribuida en un ordenador



Computación distribuida en varios ordenadores



Hadoop

Definición



*“**Apache Hadoop** es un framework de código abierto que permite el almacenamiento distribuido y el procesamiento de grandes conjuntos de datos en base a un hardware comercial”*

Volumen

Sirve para almacenar grandes volúmenes de información

Backups

Guarda copias de la información en diferentes nodos

Tolerancia a fallos

En caso de que se caiga un nodo, cuenta con otros para mantener el servicio

YARN

Gestor de recursos de Hadoop

Escalabilidad

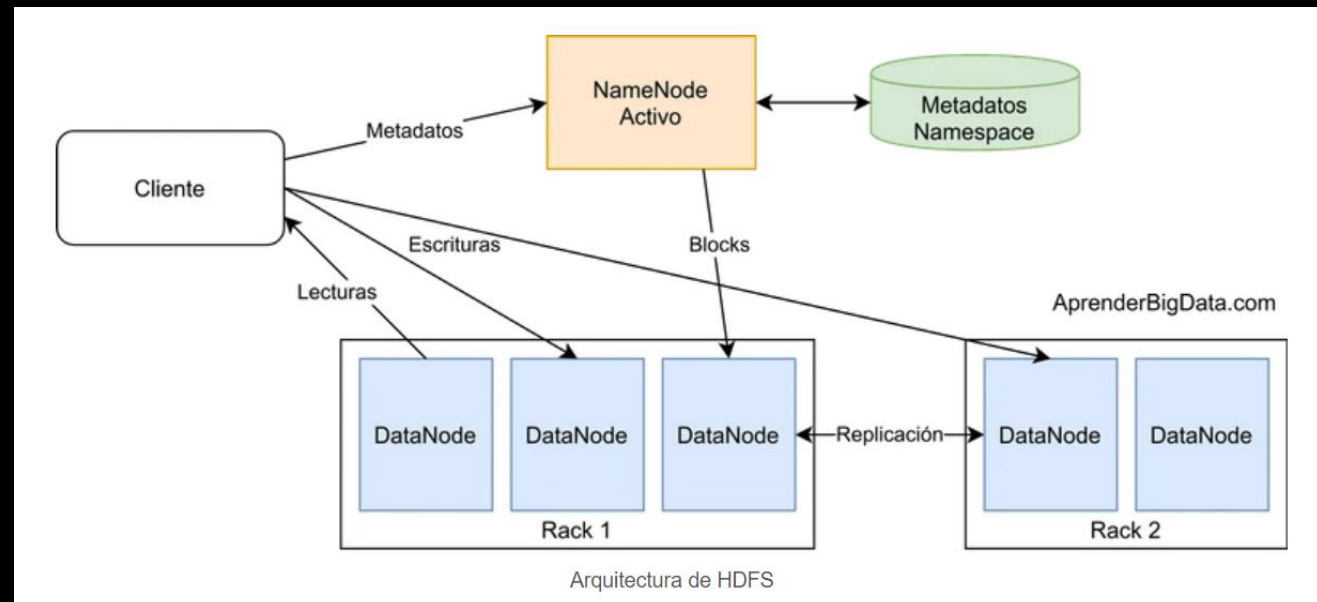
Es cuestión de añadir nuevos nodos, de hardware económico

HDFS

High Distributed File System



“Sistema de ficheros distribuidos de Hadoop. Sirve para el almacenamiento masivo de información, tanto para datos estructurados, semi-estructurados y no estructurados.”

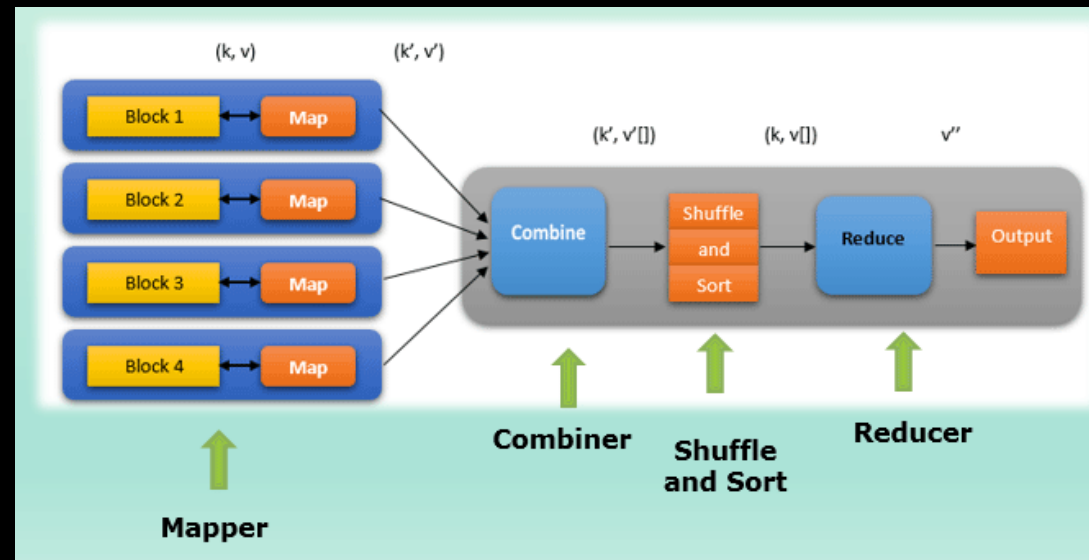


Map Reduce

Paradigma de programación



“MapReduce es una técnica de procesamiento y un programa modelo de computación distribuida basada en java. Mediante el Map se generan pares clave-valor y en el Reduce se produce la agregación.”



Spark

High Distributed File System

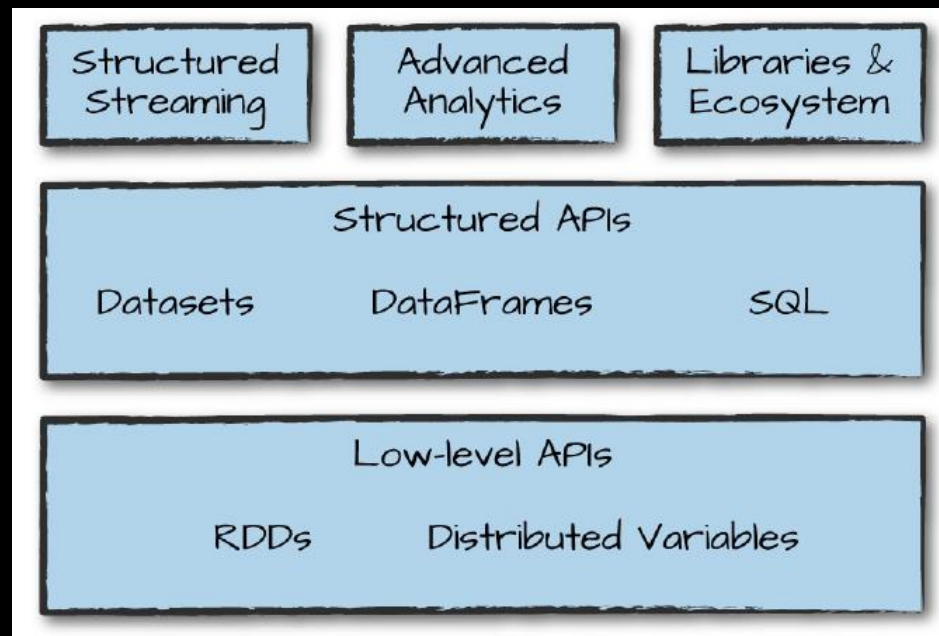


“Motor de computación, que utiliza una serie de librerías o APIs en lenguajes bien conocidos como R, Python, Scala o Java y sirve para procesar datos de forma paralela en un clúster”

Escrito en Scala
Corre sobre Java (JVM)

Se combina con Hadoop

Librerías y APIs de terceros



Spark

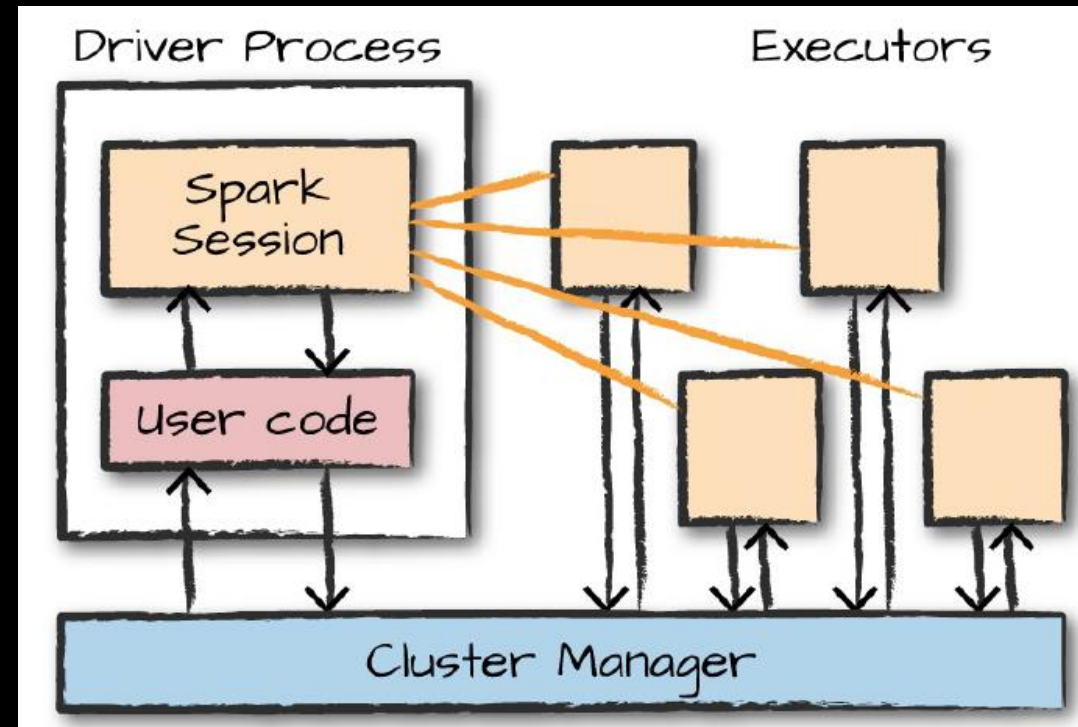
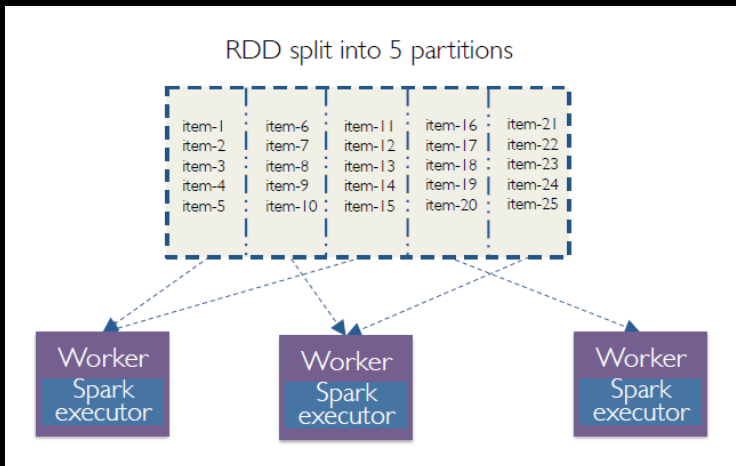
Funcionamiento



Partición

Una partición es una serie de filas de un DF que se almacenan en una maquina fisica, dentro de un cluster, por ejemplo, partición por fecha.

Trabajamos a alto nivel, no con las particiones



Spark

Funcionamiento

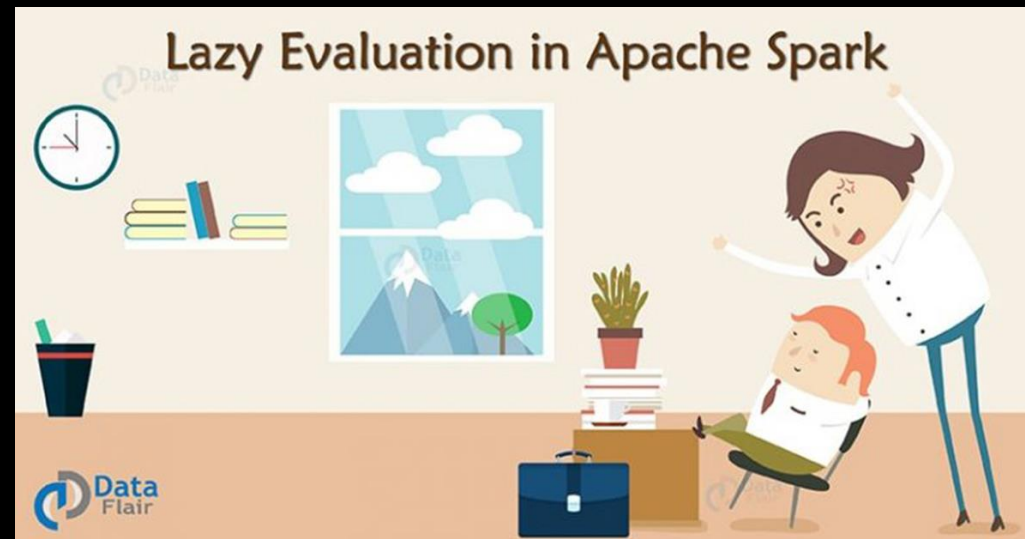


Transformaciones y acciones

Una transformación es cualquier modificación que hagamos sobre los datos, como por ejemplo, un filtrado. Mientras que en una acción necesitamos ejecutar todas las transformaciones ya que estamos pidiendo un resultado, como un count o un show. **Es el trigger de la ejecución.**




Lazy evaluation

Spark no ejecuta todas las operaciones hasta que no se ve en la necesidad de mostrar datos con una acción. Evalúa todas las operaciones de la ejecución y las ordena como crea más conveniente para que la ejecución sea óptima



Spark vs Hadoop



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 & enhances the system performance	Doesn't support caching of data

Process information

In-memory

In-disk

RDD

Resilient Distributed Datasets

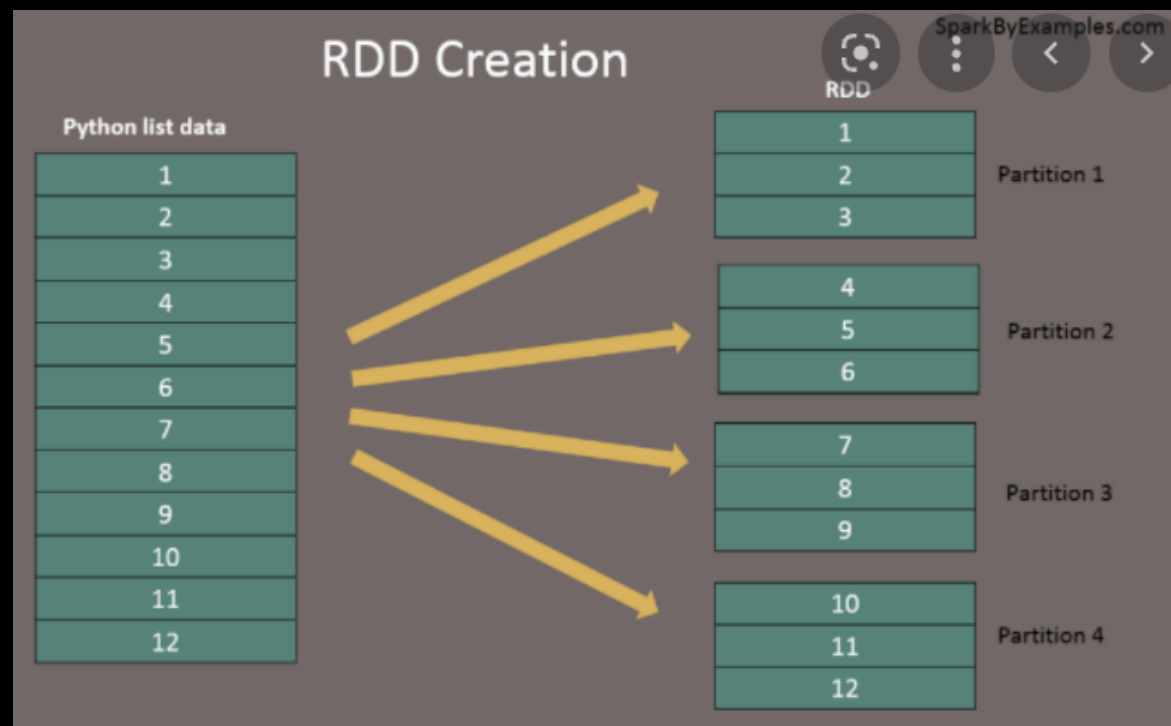


Definición

Un RDD, según Spark, se define como una **colección de elementos** que es tolerante a fallos y que es capaz de **operar en paralelo**.

Tolerante a fallos

Esto es debido a que, al ser capaces de **operar en paralelo**, la información está guardada en diferentes nodos del clúster, por lo que si se pierde en uno, podrá recuperarlo desde otro.



No son mutables

Si deseamos realizar una operación sobre un RDD, tendremos obligatoriamente que generar uno nuevo

Se crean desde archivos de Hadoop (HDFS)

aunque esto no impide que, por ejemplo, creamos un RDD a través de archivos con otra extensión como un .json o .csv, previamente subidos a HDFS

| Parquet files



*Formato de **almacenamiento de datos** orientado a **columnas**, lo que facilita el uso de una codificación y compresión eficientes para reducir tu tamaño. Es gratuito y de código abierto, característico del ecosistema Hadoop.*

Dataset	Columns	Size on Amazon S3	Data Scanned	Cost
Data stored as CSV file	4	4TB	4TB	\$20 (4TB x \$5/TB)
Data stored as GZIP CSV file	4	1TB	1TB	\$5 (1TB x \$5/TB)
Data stored as Parquet file	4	1TB	.25TB	\$1.25 (.25TB x \$5/TB)

Delta Lakes (no Data Lakes)



“Delta Lake es un proyecto de código abierto que permite crear una arquitectura de Lakehouse sobre Data Lakes. Delta Lake proporciona transacciones ACID y control escalable de metadatos, y unifica el procesamiento de datos de streaming y por lotes en los Data Lakes ya existentes.”

Transacciones ACID en Spark: garantizan que los lectores nunca vean datos incoherentes.

Controla automáticamente las variaciones de esquema para evitar la inserción de registros incorrectos durante la ingesta.

Control escalable de los metadatos.

Viajes en el tiempo: el control de versiones de datos permite reversiones, seguimientos de históricos completos y experimentos reproducibles de aprendizaje automático.

Unificación de streaming y lotes: una tabla en Delta Lake es una tabla de lotes, así como un origen y un receptor de streaming. La ingesta de datos de streaming, la reposición histórica de lotes y las consultas interactivas funcionan de manera integral.