Fundamentos de

Big Data

Sesión 2

¿Qué es Spark y por qué se necesita?

Apache Spark es un motor de procesamiento de datos optimizado para manejar grandes volúmenes de información mediante procesamiento distribuido en memoria. Fue desarrollado originalmente por el **AMPLab** de la Universidad de California, Berkeley, y ahora es mantenido por la **Apache Software Foundation**.

Se necesita Spark principalmente porque las arquitecturas tradicionales de procesamiento de datos, como Hadoop MapReduce, presentan limitaciones en velocidad y flexibilidad cuando se trabaja con grandes volúmenes de datos. Spark aborda estos problemas con un modelo basado en **RDDs** (*Resilient Distributed Datasets*), procesamiento en memoria y una estructura optimizada para cargas de trabajo iterativas y en tiempo real.

Conceptos básicos y problemas que resuelve

CONCEPTOS BÁSICOS

1. Procesamiento distribuido.
2. RDD (Resilient Distributed Dataset).
3. Spark Core.
4. APIs de alto nivel.
5. Componentes adiciones: Spark SQL, Spark Streaming, Mllib y GraphX

PROBLEMAS QUE RESUELVE

1. Lentitud en el procesamiento de datos grandes.
2. Dificultad en el manejo de datos en tiempo real.
3. Complejidad en la integración con múltiples fuentes de datos.
4. Limitaciones en el aprendizaje automático.
5. Escalabilidad

API de Spark

Python

Ampliamente utilizado por su facilidad de uso.

Scala
Lenguaje nativo de Spark. Flexibilidad y eficiencia.

Java

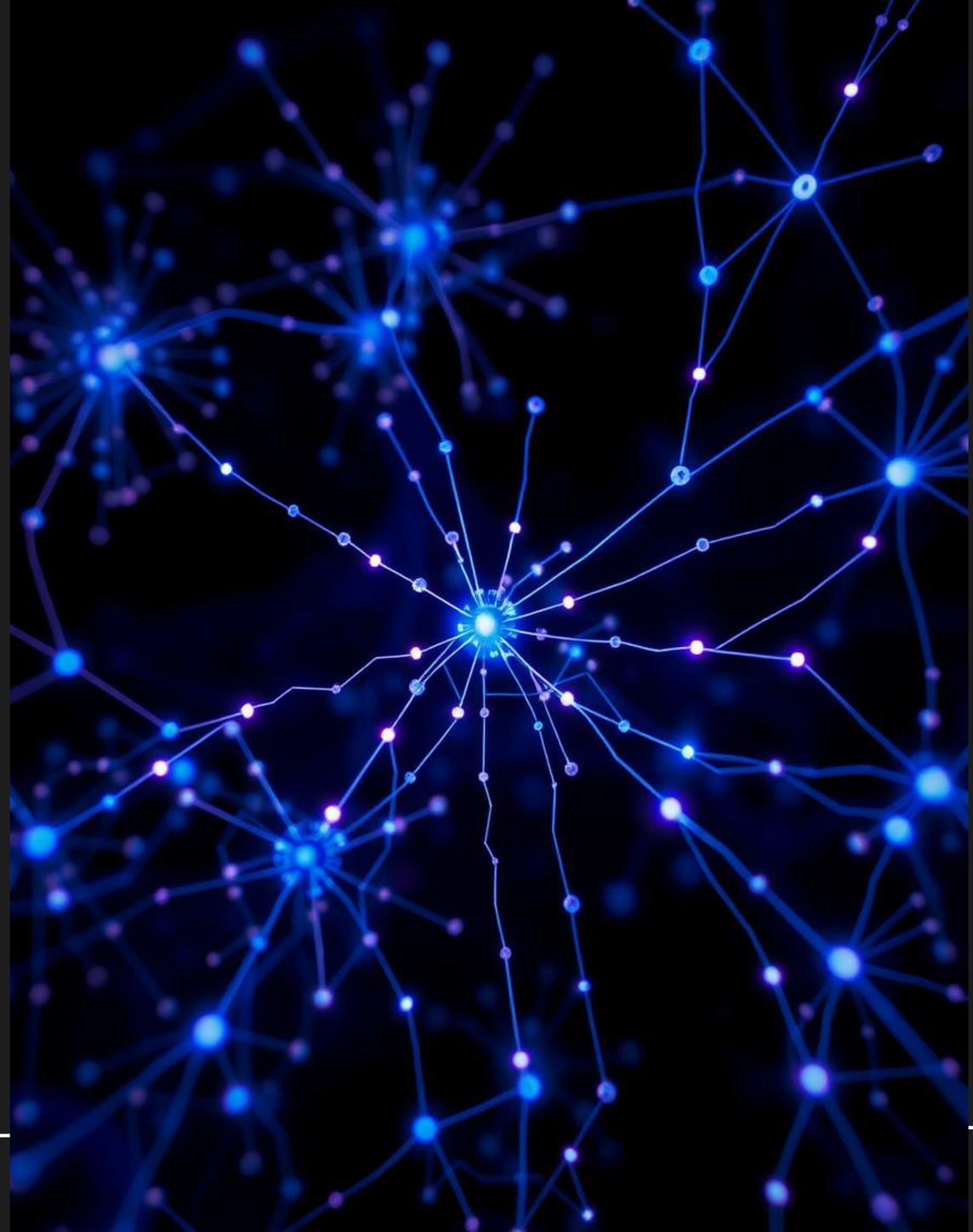
Sintaxis más verbosa, pero completamente funcional.

R

Ideal para trabajar con R en el ámbito estadístico.

Arquitectura General de Spark

1. **Spark Driver**
2. **Spark Cluster Manager**
3. **Executors**
4. **Tasks**
5. RDDs (Resilient Distributed Datasets)
6. • DAG (Directed Acyclic Graph)



Interacción entre Driver, Executor y Cluster Manager

- **El Driver** crea y coordina el plan de ejecución del trabajo, generando el DAG de tareas.
- **El Cluster Manager** asigna recursos para ejecutar estas tareas, ya sea en modo Standalone, sobre YARN o Mesos.
- **Los Executors** son los que realmente ejecutan las tareas en paralelo en el clúster, procesando las particiones de los datos y devolviendo los resultados al Driver.

Resumen de la Arquitectura

Driver	Executor	Cluster Manager
Coordina la ejecución, mantiene el SparkContext, crea el DAG, y supervisa las tareas.	Ejecuta las tareas distribuidas y almacena datos en memoria (o en disco si es necesario).	Gestiona los recursos del clúster y asigna tareas a los ejecutores.

Implementaciones de Spark

En la Nube: AWS, GCP, Microsoft Azure

En Windows: Java, Hadoop, Scala, Spark

PySpark

PySpark es la interfaz de Python para Apache Spark, que permite a los desarrolladores escribir trabajos de Spark utilizando el lenguaje de programación Python. Esta es una opción popular para científicos de datos y desarrolladores que prefieren trabajar en Python en lugar de Java o Scala.

Características clave:

Actividad Práctica Guiada

Objetivo: Procesar un conjunto de datos de ventas de productos para obtener información relevante sobre el rendimiento de cada producto.

Requisitos:

1. **Instalar PySpark.**
2. **Crear un nuevo archivo Python y configurar PySpark.**
3. **Cargar los datos desde un archivo CSV.**
4. **Realizar operaciones de filtrado y agrupamiento.**
5. **Realizar una agregación de precios totales por producto.**
6. **Guardar los resultados en un nuevo archivo CSV.**

El detalle de la actividad se encuentra en la guía de estudio de la sesión.

Preguntas

Sección de preguntas

Fundamentos de

Big Data

Continúe con las actividades