Fundamentos de

Big Data

Sesión 3

Configuración, Conexión y Contexto de Spark

Configuración de Spark

- Standalone
- Cluster Manager (YARN, Mesos, Kubernetes)
- Local Mode

Conexión y Contexto

```python
from pyspark import SparkContext

sc = SparkContext("local", "MiApp")  # "local" indica ejecución en un solo nodo
```

SparkContext

```python
from pyspark.sql import SparkSession

spark = SparkSession.builder.appName("MiApp").getOrCreate()
sc = spark.sparkContext  # Obtener el SparkContext desde SparkSession
```

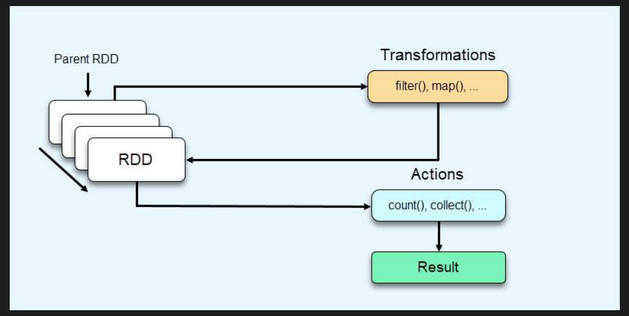
SparkSession

RDD: Qué es RDD

Un **RDD (*Resilient Distributed Dataset*)** es la estructura fundamental de datos en Spark. Es una colección distribuida de elementos que se pueden procesar en paralelo.

Características principales de un RDD:

1. **Inmutable:** Una vez creado, no puede modificarse, solo transformarse en un nuevo RDD.
2. **Distribuido:** Los datos se dividen en particiones y se distribuyen entre los nodos del clúster.
3. **Resiliente:** Soporta fallos debido a su capacidad de reconstrucción a partir de su linaje.



PAIR RDD

Un Pair RDD es un tipo especial de RDD donde cada elemento es una tupla (clave, valor).

Operaciones comunes:

groupByKey()

reduceByKey()

Transformaciones

```python
# 1. filter
rdd.filter(lambda x: x % 2 == 0).collect()  # Solo pares

# 2. map
rdd.map(lambda x: x * 2).collect()

# 3. flatMap
rdd = sc.parallelize(["hola mundo"])
rdd.flatMap(lambda x: x.split(" ")).collect()
# ['hola', 'mundo']

# 4. sample
rdd.sample(False, 0.5).collect()

# 5. union
rdd.union(rdd2).collect()

# 6. distinct
rdd.distinct().collect()

# 7. sortBy
rdd.sortBy(lambda x: x).collect()
```

1. filter(f)
2. map(f)
3. flatmap(f)
4. sample(fraction, replacement)
5. union(otherrdd)
6. distinct()
7. sortby(f,ascending=true)

Acción

1. collect()
2. take(n)
3. top(n)
4. takesample()
5. sum()
6. mean()
7. stdey()

```python
# 1. collect
rdd.collect()

# 2. taken
rdd.take(3)

# 3. top
rdd.top(2)

# 4. takeSample
rdd.takeSample(False, 2)

# 5. sum
rdd.sum()

# 6. mean
rdd.mean()

# 7. stdev
rdd.stdev()
```

Job Spark

Un **Job** en Spark es una unidad de trabajo que se ejecuta en el clúster. Se activa cuando se llama a una acción en un RDD o DataFrame.

Ejemplo:

```
rdd.count()  # Acción que dispara un Job
```

Cada job se divide en:

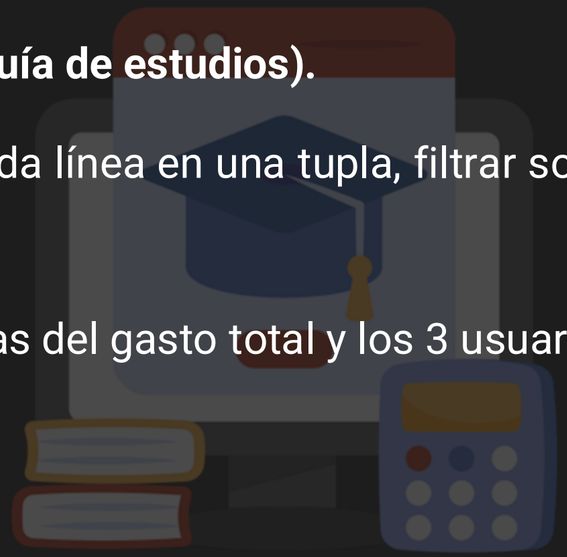
1. **Stages (etapas):** Secciones del job que pueden ejecutarse en paralelo.
2. **Tasks (tareas):** Unidades de ejecución en cada nodo.

Actividad Práctica Guiada

Objetivo: Aplicar conceptos de configuración, conexión, RDDs, transformaciones y acciones en Apache Spark.

Requisitos:

1. **Configuración y conexión con Spark.**
2. **Cargar los datos en un RDD (Datos en la guía de estudios).**
3. **Transformaciones en el RDD:** convertir cada línea en una tupla, filtrar solo las transacciones mayores a $1000, obtener los montos totales gastados por usuario.
4. **Acciones y resultados:** estadísticas básicas del gasto total y los 3 usuarios con mayores gastos.
5. **Ejecución en un Job Spark.**



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