Curso Hackea Tu Futuro – Big Data Madrid Granada(Módulo 6)

Nombre: Pedro Jesús

Apellidos: Román Jiménez

Fecha: 20-11-2023

Responde a las siguientes preguntas. Justifica la respuesta.

1. ¿Qué es Apache Spark?

   Apache Spark es un motor unificado de analíticas para procesar datos a gran escala que integra módulos para SQL, streaming, aprendizaje automático y procesamiento de grafos. Spark se puede ejecutar de forma independiente o en Apache Hadoop, Apache Mesos, Kubernetes, la nube y distintas fuentes de datos.

2. ¿Qué son las funciones lambda?

   En el ámbito de la programación, una función lambda o función anónima, es una subrutina definida que no está enlazada a un identificador. una expresión lambda puede ser sintácticamente más simple que una función nombrada. Además, son muy comunes en la programación funcional y otros lenguajes con funciones de primera clase.

3. ¿Para qué sirven los map()? ¿Cómo se usan con RDDs y cómo se usan con listas? Muestra la sintaxis con un ejemplo.

   map() es una función que se utiliza en la programación funcional para aplicar una función a cada elemento de una colección, como una lista o un RDD (Resilient Distributed Dataset) en el contexto de Apache Spark. La función *map()* crea una nueva colección (lista o RDD) que contiene los resultados de aplicar la función a cada elemento de la colección original, sin modificar la colección original. A continuación se muestra un ejemplo de su uso con listas y RDD:

   ```
   datos = [1,2,3]

   list(map(lambda x : x+1, datos))
   ```

   ```
   rdd = sc.parallelize(datos)

   rdd.map(lambda x : x+1)

   rdd.collect()
   ```

4. ¿Para qué sirven los filter()? ¿Cómo se usan con RDDs y cómo se usan con listas? Muestra la sintaxis con un ejemplo.

   filter() es una función que se utiliza en la programación funcional para filtrar elementos de una colección (como una lista o un RDD) según una condición dada. Permite crear una nueva colección que contiene solo los elementos que cumplen con la condición especificada, sin modificar la colección original. A continuación se muestra un ejemplo de su uso con listas y RDD:

```
# Definimos una lista de números
numeros = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]


# Definimos una función que verifica si un número es par
def es_par(numero):
    return numero % 2 == 0


# Usamos filter() para obtener una lista de números pares
numeros_pares = filter(es_par, numeros)


# Convertimos el resultado a una lista (filter() devuelve un objeto iterable)
numeros_pares_lista = list(numeros_pares)


# Imprimimos el resultado
print(numeros_pares_lista)
```

```
# Importamos las bibliotecas necesarias
from pyspark import SparkContext


# Creamos un contexto Spark
sc = SparkContext("local", "EjemploFilter")


# Creamos un RDD a partir de una lista de números
rdd = sc.parallelize([1, 2, 3, 4, 5, 6, 7, 8, 9, 10])
```

```
# Definimos una función que verifica si un número es par
def es_par(numero):
    return numero % 2 == 0

# Usamos filter() para obtener un nuevo RDD con los números pares
numeros_pares = rdd.filter(es_par)

# Recopilamos los resultados en una lista (esto es opcional)
numeros_pares_lista = numeros_pares.collect()

# Imprimimos el resultado
print(numeros_pares_lista)

# Detenemos el contexto Spark
sc.stop()
```

5. ¿Cuál es la diferencia entre transformaciones y acciones?

   La principal diferencia es que una acción devuelve un resultado y una transformación modifica a un RDD.

6. ¿Cuál es el tamaño del rdd de salida al aplicarle un filter?

   Siempre menor o igual que el original.

7. ¿Qué acción realiza count() en una instrucción de Spark? ¿Y el collect()?

   La acción *count()* devuelve el número de elementos del RDD, mientras que la ación *collect()* devuelve una lista con todos los elementos del RDD.

8. ¿Qué es un cluster?

   Un cluster, en el contexto de la informática y la tecnología, se refiere a un conjunto o grupo de computadoras interconectadas que trabajan juntas de manera coordinada para realizar tareas específicas. Estas tareas pueden

incluir procesamiento de datos, cálculos complejos, almacenamiento y distribución de recursos, administración de servicios, entre otros.

9. Menciona algunas transformaciones junto con una breve explicación para cada una.

   Entre las principales transformaciones, podemos destacar las siguientes:

   - *distinct()* → crea un nuevo RDD eliminando duplicados.
   - *sample()* → remuestrea el RDD de entrada con o sin reemplazamiento.
   - *union()* → une dos RDD en uno.

10. Menciona algunas acciones junto con una breve explicación para cada una.

    Entre las principales acciones, podemos destacar las siguientes:

    - *count()* → devuelve el número de elementos del RDD.
    - *take()* → devuelve una lista con los primeros n elementos del RDD. (similar a la función *head.()*).
    - *collect()* → devuelve una lista con todos los elementos del RDD.
    - *takeOrdered()* → devuelve una lista con los primeros n elementos en forma ascendente.

11. ¿Podemos crear un RDD a partir de distintos tipos de archivos como un dataset local, Cassandra, HBase, Amazon S3...?

    En el contexto de Apache Spark, puedes crear RDDs (Resilient Distributed Datasets) a partir de diversas fuentes de datos, incluyendo datos locales, bases de datos NoSQL como Cassandra y HBase, así como sistemas de almacenamiento en la nube como Amazon S3. Apache Spark proporciona API y conectores para trabajar con una amplia variedad de fuentes de datos.

12. ¿Qué es la evaluación perezosa?

    Una evaluación perezosa es un proceso de Spark que no será ejecutado hasta que no sea necesario en una determinada acción.