**RDD (Conjunto de Datos Distribuidos Resilientes)**

Los RDD son la estructura de datos fundamental en Spark. Son colecciones distribuidas de objetos que pueden ser procesados en paralelo. Los RDD son inmutables y se pueden crear a partir de datos en almacenamiento o a partir de otros RDD.

**Transformaciones**

Las transformaciones son operaciones que crean un nuevo RDD a partir de uno existente. Son perezosas, lo que significa que no se ejecutan hasta que se llama a una acción.

1. **Map:** Esta transformación aplica una función a cada elemento del RDD, generando un nuevo RDD. Es útil para transformar datos, como convertir tipos de datos o aplicar cálculos.
2. **Filter:** Permite seleccionar elementos que cumplen con una condición específica. Esto es esencial para limpiar datos o extraer subconjuntos relevantes.
3. **FlatMap:** Similar a map, pero permite que cada entrada produzca múltiples salidas. Esto es útil para descomponer estructuras de datos complejas en elementos individuales.
4. **Unión (Union):** Combina dos RDDs en uno solo, manteniendo todos los elementos de ambos. Es útil para consolidar datos de diferentes fuentes.
5. **Intersección (Intersection)**: Devuelve un nuevo RDD que contiene solo los elementos comunes entre dos RDDs. Esto es útil para encontrar coincidencias o elementos compartidos.
6. **Distinct**: Elimina duplicados en el RDD, devolviendo solo elementos únicos. Es esencial para asegurar la calidad de los datos.
7. **groupByKey**: Agrupa los valores de un RDD de pares clave-valor por clave. Esto permite realizar análisis agregados sobre los datos.
8. **reduceByKey**: Combina los valores de cada clave utilizando una función de reducción. Es más eficiente que groupByKey porque reduce los datos antes de agruparlos.
9. **sortByKey**: Ordena los elementos de un RDD de pares clave-valor según las claves. Esto es útil para organizar datos para análisis posteriores.
10. **Join**: Combina dos RDDs de pares clave-valor basándose en las claves. Permite relacionar datos de diferentes conjuntos.
11. **Cogroup**: Agrupa los valores de dos RDDs por clave, devolviendo un RDD donde cada clave tiene una lista de valores de ambos RDDs. Esto es útil para comparaciones y análisis conjuntos.
12. **Coalesce**: Reduce el número de particiones en un RDD. Esto puede mejorar el rendimiento al disminuir la sobrecarga de procesamiento en operaciones posteriores.

### Acciones

1. **Reduce**: Combina todos los elementos del RDD utilizando una función de reducción. Esto es útil para obtener un resultado agregado, como una suma o un promedio.
2. **Collect**: Devuelve todos los elementos del RDD al controlador como una lista. Es útil para inspeccionar resultados pequeños, pero debe usarse con precaución en grandes conjuntos de datos.
3. **Count**: Devuelve el número total de elementos en el RDD. Es una operación básica para entender el tamaño de los datos.
4. **First**: Devuelve el primer elemento del RDD. Es útil para obtener una muestra rápida de los datos.
5. **Take**: Similar a first, pero permite especificar cuántos elementos devolver. Esto es útil para obtener una vista rápida de un subconjunto de datos.
6. **saveAsTextFile**: Guarda el contenido del RDD en un archivo de texto en el sistema de archivos. Es esencial para la persistencia de datos procesados.
7. **Max**: Devuelve el valor máximo en el RDD. Esto es útil para análisis estadísticos.
8. **Min**: Devuelve el valor mínimo en el RDD. Similar a max, es útil para obtener estadísticas descriptivas.
9. **CountByKey**: Devuelve un diccionario con la cantidad de elementos por clave en un RDD de pares clave-valor. Esto es útil para análisis de frecuencia.
10. **Foreach**: Aplica una función a cada elemento del RDD, pero no devuelve nada. Es útil para realizar efectos secundarios, como escribir en una base de datos o imprimir resultados.