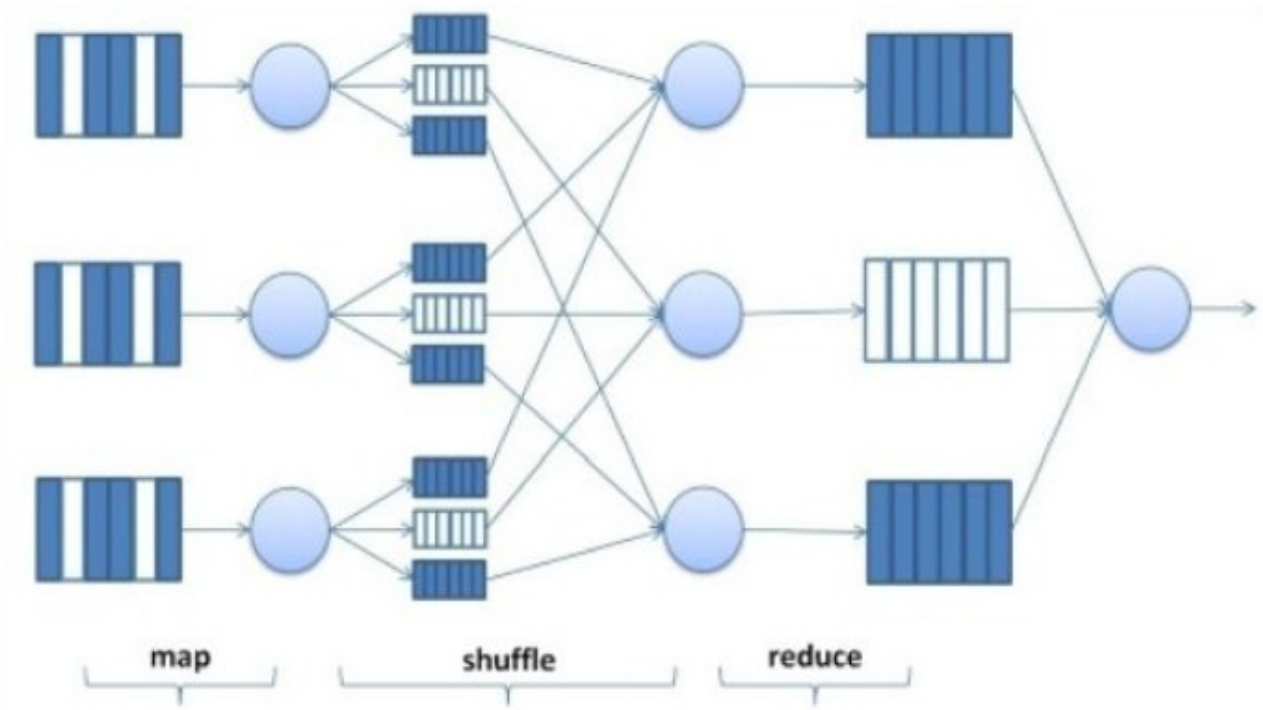
Map Reduce en pyspark

MapReduce es un paradigma de programación paralelo que abstrae las complejidades de computación y datos de paralelismo en un entorno de computación distribuida. Funciona sobre el concepto de llevar la función de cálculo a los datos en lugar de llevar los datos a la función de cálculo.

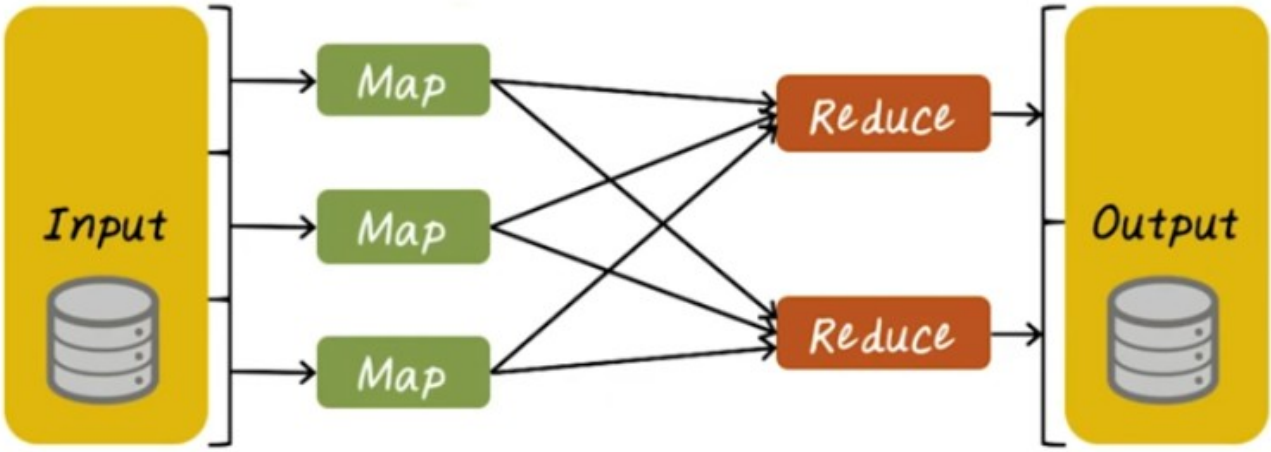
MapReduce es más un marco de programación que viene con muchas funciones incorporadas que el desarrollador no necesita preocuparse por construir y puede reducir muchas complejidades de implementación como el particionamiento de datos, la programación, la administración de excepciones y las comunicaciones entre sistemas. La figura siguiente muestra una composición típica de la función MapReduce:



Esto tiene, en el núcleo, las funciones Map () y Reduce () que son capaces de ejecutarse en paralelo a través de los nodos del clúster. La función Map () funciona en los datos distribuidos y ejecuta la funcionalidad requerida en paralelo, y la función Reduce () ejecuta una operación de resumen de los datos.

La **fase map** tiene como objetivo realizar un mapeo(mapping) de los datos de entrada,es decir,un cambio de dominio. Fundamentalmente esta rutina servirá para realizar una conversión de la entrada y si fuera necesario,un filtrado de datos.

El objetivo de la **fase reduce** es llevar a cabo una agregación de los datos que han sido mapeados con anterioridad.esto podría ser el equivalente a algunas funciones de agregación de sql como sum,avg,count



Map reduce es un paradigma,es decir,una forma de pensar a la hora de resolver un determinado problema.Este paradigma surge para dar respuesta a las necesidades de google de llevar a cabo un procesamiento masivo de los datos.

Este procedimiento de datos se lleva a cabo de forma distribuida,de tal forma que las tareas concretas a realizar se dividen en varias subtareas que se reparten entre diferentes nodos del cluster. Cada nodo ejecuta su parte de forma paralela a los demás y finalmente los resultados se agregan.

Además,una ventaja esencial de este paradigma es que el desarrollador no tiene que preocuparse por la forma en la que las tareas se dividen entre los diferentes nodos del cluster,ni tampoco debe prever qué ocurrirá si uno de estos nodo falla.El sistema gestiona automáticamente estos supuestos,de tal forma que el desarrollador puede centrarse en construir programas que se encargarán de procesar los datos.

En Spark las funciones que más se utilizan son las siguientes:

- **map(func):** Devuelve un nuevo RDD ,resultado de pasar cada uno de los elementos del RDD original como parámetro de la función func.Aplica una función a cada elemento de la colección.

- **flatMap(func):** similar a map(func),pero en caso de que func retorne una lista,está no será mapeada directamente en el RDD resultante,sino que se mapearán individualmente los elementos contenidos.

- **reduceByKey(func):** devuelve un RDD de pares clave/valor donde cada clave única se corresponde con las diferentes claves del RDD original y el valor es el resultado de aplicar una operación reduce sobre los valores correspondientes a una misma clave.En resumen,combina valores con la misma clave en cada procesado.