mapPartitions Transformation

mapPartitions( ) - This can be used when we need specific data from each partition of the RDD. In the example below, we are aiming to find the minimum and maximum values of all the values in the RDD. Using map( ) can create a lot of Intermediary key-value (K, V) pairs for the simple task of finding minimum and maximum value of numbers in the RDD.

In [3]:

```
import pyspark

from pyspark.sql import SparkSession

#Create the Spark Session with 4 partitions with master("local[4]")
spark = SparkSession.builder \
    .master("local[4]") \
    .appName('test') \
    .getOrCreate()

#Create an rdd with integers in the range of 1 to 1000
rdd = spark.sparkContext.range(1,1000)

#Printing the count of elements in RDD
print('data count =', rdd.count())

#Check the number of Partitions in the RDD
print("Number of Partitions = ", rdd.getNumPartitions())
```

```
data count = 999
Number of Partitions =  4
```

In [4]:

```
def minmax(iterator):
    a = True
    for x in iterator:
        if(a):
            local_min = x;
            local_max = x;
            a = False
        else:
            local_min = min(x, local_min)
            local_max = max(x, local_max)
    return (local_min, local_max)

minmax_rdd = rdd.mapPartitions(minmax)
print("List of Minimum and Maximum values of each partition = ", minmax_rdd.collect())

minmax_list = minmax_rdd.collect()
print("Minimum value of the list = ", min(minmax_list))
print("Maximum value of the list = ", max(minmax_list))
```

```
List of Minimum and Maximum values of each partition =  [1, 249, 250, 499, 500, 749, 750, 999]
Minimum value of the list =  1
Maximum value of the list =  999
```

In [ ]: