coalesce( ) Transformation

In [1]:
```
from pyspark.sql import SparkSession
from pyspark.sql.functions import col

spark = SparkSession \
        .builder \
        .master("local[*]") \
        .appName("pipe Transformation") \
        .enableHiveSupport() \
        .getOrCreate()
```

```
22/10/14 10:54:53 WARN Utils: Your hostname, Vaishalis-MacBook-Pro.local resolves to a loopback address: 127.0.
0.1; using 192.168.0.105 instead (on interface en0)
22/10/14 10:54:53 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
22/10/14 10:54:54 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builti
n-java classes where applicable
22/10/14 10:54:55 WARN Utils: Service 'SparkUI' could not bind on port 4040. Attempting port 4041.
22/10/14 10:54:55 WARN Utils: Service 'SparkUI' could not bind on port 4041. Attempting port 4042.
```

In [2]:
```
#Reading data from a file on the local machine
data_file_path = "/Users/vaishaliyasala/Desktop/Github/Spark/Exercise_Dependencies/SalesJan2009.csv"

df = spark.read.csv(data_file_path, header = True )

df1 = df.select(df["Name"],df["Country"]).repartition(4)

print("Count in the original data=", df1.count())


#Filter the names only from country United States
names_rdd = df1.rdd.filter(lambda x: (x[1] == "United States"))

filtered_data = spark.createDataFrame (data = names_rdd, schema = df1.printSchema())
filtered_data.show(5)

print("Filtered data Count =", names_rdd.count())
print("Number of Partitions =", names_rdd.getNumPartitions())
```

```
Count in the original data= 998
root
 |-- Name: string (nullable = true)
 |-- Country: string (nullable = true)

+---------+-------------+
|     Name|      Country|
+---------+-------------+
|   Abikay|United States|
|Christian|United States|
|  Alicja |United States|
|  Debora |United States|
| Sandrine|United States|
+---------+-------------+
only showing top 5 rows

Filtered data Count = 463
Number of Partitions = 4
```

In [3]:
```
#As the filtered data count increased significantly, we can reduce the number of partitions from 4 to 2
#This doesn't change the results as seen from the outputs of both before and after coalesce transformation

names_coalesce_rdd = names_rdd.coalesce(2)

coalesced_data = spark.createDataFrame (data = names_coalesce_rdd, schema = df1.printSchema())
coalesced_data.show(5)

print("Number of Partitions =", names_coalesce_rdd.getNumPartitions())
```

```
root
 |-- Name: string (nullable = true)
 |-- Country: string (nullable = true)

+---------+-------------+
|     Name|      Country|
+---------+-------------+
|   Abikay|United States|
|Christian|United States|
|  Alicja |United States|
|  Debora |United States|
| Sandrine|United States|
+---------+-------------+
only showing top 5 rows

Number of Partitions = 2
```