cogroup( ) Transformation

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

spark = SparkSession \
        .builder \
        .master("local[2]") \
        .appName("cogroup Transformation") \
        .enableHiveSupport() \
        .getOrCreate()

#key-value pairs from dataset 1
kvPair = [(1, 2.5), (2, 4.0), (3, 5.5) ]

kvPairRdd = spark.sparkContext.parallelize(kvPair)
print(kvPairRdd.collect())

#key-value pairs from dataset 2
otherKvPair = [(1, 3.5), (1, 4.0), (2, 2.5), (2, 3.5), (4, 10.0), (3, 4.5) ]

otherKvPairRdd = spark.sparkContext.parallelize(otherKvPair)
print(otherKvPairRdd.collect())
```

```
22/10/14 01:01:26 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 01:01:26 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 01:01:27 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builti
n-java classes where applicable
22/10/14 01:01:28 WARN Utils: Service 'SparkUI' could not bind on port 4040. Attempting port 4041.
[Stage 0:>                                                          (0 + 2) / 2]
[(1, 2.5), (2, 4.0), (3, 5.5)]
[(1, 3.5), (1, 4.0), (2, 2.5), (2, 3.5), (4, 10.0), (3, 4.5)]
```

In [2]:
```
#kvPairRdd has (K,V)
#otherKvPairRdd has (K,W)
#cogroup returns a dataset of (K, (Iterable<V>, Iterable<W>)) tuples.
#That means for kvPairRDD, there are a list of values for each key in Iterable<V>
#That means for otherKvPairRDD, there are a list of values for each key in Iterable<W>

cogroupKvRdd = kvPairRdd.cogroup(otherKvPairRdd)
print(cogroupKvRdd.collect())
```

```
[Stage 2:=============================>                             (2 + 2) / 4]
[(4, (<pyspark.resultiterable.ResultIterable object at 0x106fce830>, <pyspark.resultiterable.ResultIterable obj
ect at 0x112e81750>)), (1, (<pyspark.resultiterable.ResultIterable object at 0x112e81780>, <pyspark.resultitera
ble.ResultIterable object at 0x112e817e0>)), (2, (<pyspark.resultiterable.ResultIterable object at 0x112e81810
>, <pyspark.resultiterable.ResultIterable object at 0x112e81870>)), (3, (<pyspark.resultiterable.ResultIterable
object at 0x112e818d0>, <pyspark.resultiterable.ResultIterable object at 0x112e81930>))]
```

In [3]:
```
#Let us look at the actual values

print([(x,tuple(map(list,y)))for x,y in sorted(list(cogroupKvRdd.collect()))])
```

```
[(1, ([2.5], [3.5, 4.0])), (2, ([4.0], [2.5, 3.5])), (3, ([5.5], [4.5])), (4, ([], [10.0]))]
```