In [1]:

```python
from pyspark.sql import SparkSession
import pyspark.sql.functions as F
from pyspark.sql.types import *
```

In [2]:

```python
spark = SparkSession\
    .builder\
    .appName("chapter-15-cluster")\
    .getOrCreate()
```

In [3]:

```python
import os
SPARK_BOOK_DATA_PATH = os.getenv('SPARK_BOOK_DATA_PATH')

SPARK_BOOK_DATA_PATH
```

Out[3]:

```
'/home/wengong/spark_data/'
```

In [4]:

```python
spark
```

Out[4]:

SparkSession - in-memory

SparkContext

Spark UI (http://192.168.0.114:4043)

Version

`v3.0.1`

Master

`local[*]`

AppName

`chapter-15-cluster`

In [5]:

```python
df1 = spark.range(2, 10000000, 2)
df2 = spark.range(2, 10000000, 4)
step1 = df1.repartition(5)
step12 = df2.repartition(6)
step2 = step1.selectExpr("id * 5 as id")
step3 = step2.join(step12, ["id"])
step4 = step3.selectExpr("sum(id)")

step4.collect() # 2500000000000
```

Out[5]:

```
[Row(sum(id)=2500000000000)]
```

In [6]:

```
step4.explain()
```

```
== Physical Plan ==
*(7) HashAggregate(keys=[], functions=[sum(id#8L)])
+- Exchange SinglePartition, true, [id=#66]
   +- *(6) HashAggregate(keys=[], functions=[partial_sum(id#8L)])
      +- *(6) Project [id#8L]
         +- *(6) SortMergeJoin [id#8L], [id#2L], Inner
            :- *(3) Sort [id#8L ASC NULLS FIRST], false, 0
            :  +- Exchange hashpartitioning(id#8L, 200), true, [id=#5
0]
            :     +- *(2) Project [(id#0L * 5) AS id#8L]
            :        +- Exchange RoundRobinPartitioning(5), false, [id
=#46]
            :           +- *(1) Range (2, 10000000, step=2, splits=4)
            +- *(5) Sort [id#2L ASC NULLS FIRST], false, 0
               +- Exchange hashpartitioning(id#2L, 200), true, [id=#5
7]
                  +- Exchange RoundRobinPartitioning(6), false, [id=#5
6]
                     +- *(4) Range (2, 10000000, step=4, splits=4)
```

In [7]:

```
step2.show(3)
```

```
+--------+
|      id|
+--------+
|10582630|
| 2424040|
| 3263590|
+--------+
only showing top 3 rows
```

In [8]:

```
step3.show(3)
```

```
+----+
|  id|
+----+
|1950|
|2250|
|4590|
+----+
only showing top 3 rows
```

In [9]:

```
step4.show(3)
```

```
+-------------+
|      sum(id)|
+-------------+
|2500000000000|
+-------------+
```

In [17]:

```
spark.range(11).where("id %2 = 0").show()
```

```
+---+
| id|
+---+
|  0|
|  2|
|  4|
|  6|
|  8|
| 10|
+---+
```

In [18]:

```
spark.range(11).where("id %2 = 0").selectExpr("sum(id)").collect()
```

Out[18]:

```
[Row(sum(id)=30)]
```

Spark UI

In [10]:

```
file_path = SPARK_BOOK_DATA_PATH + "/data/retail-data/all/online-ret
```

```
spark.read\
  .option("header", "true")\
  .csv(file_path)\
  .repartition(2)\
  .selectExpr("instr(Description, 'GLASS') >= 1 as is_glass")\
  .groupBy("is_glass")\
  .count()\
  .collect()
```

In [11]:

```
df = (
    spark.read
    .option("header", "true")
    .csv(file_path)
    .repartition(2)
    .selectExpr("instr(Description, 'GLASS') >= 1 as is_glass")
    .groupBy("is_glass")
    .count()
)

```

In [12]:

```
df.show()
```

```
+--------+------+
|is_glass| count|
+--------+------+
|    null|  1454|
|    true| 12861|
|   false|527594|
+--------+------+
```

In [13]:

```
df.collect()
```

Out[13]:

```
[Row(is_glass=None, count=1454),
 Row(is_glass=True, count=12861),
 Row(is_glass=False, count=527594)]
```

In [ ]:

```

```