Data-Intensive Distributed Computing

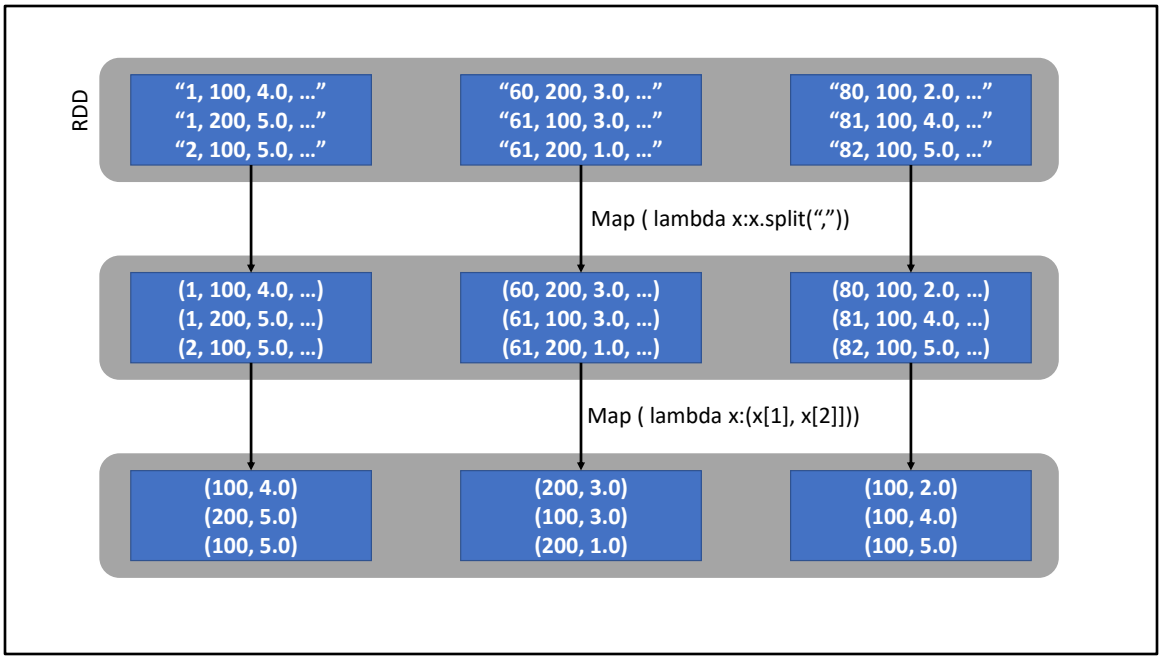
CS 431/631/451/651 (Fall 2021)

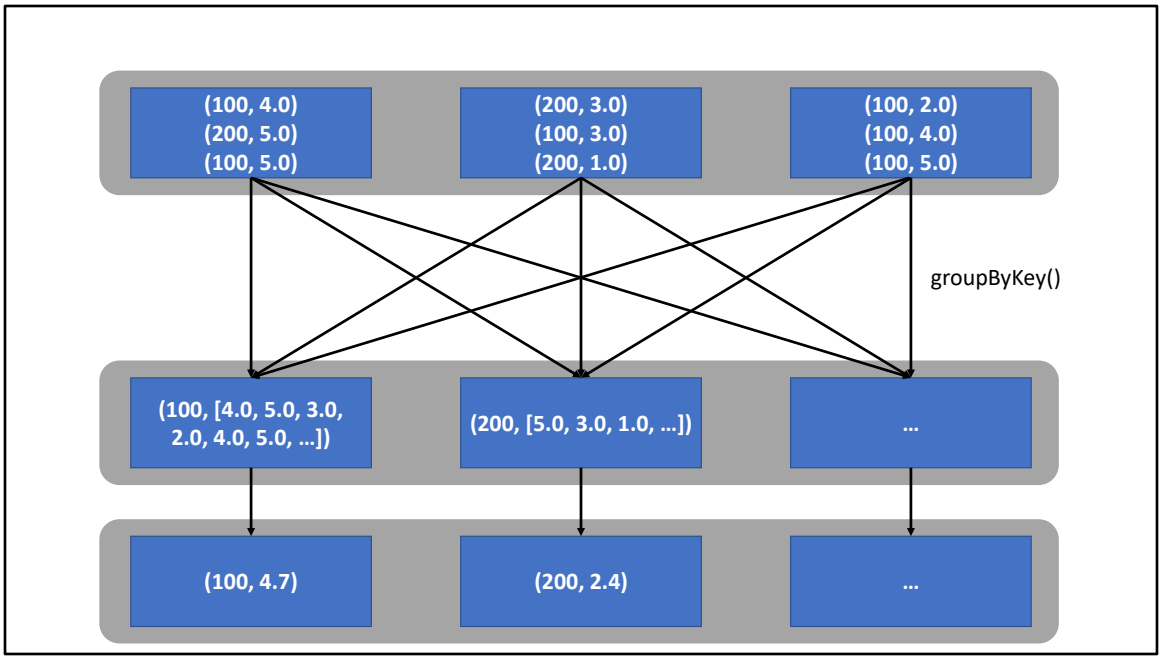
Part 3: From MapReduce to Spark (3/3)

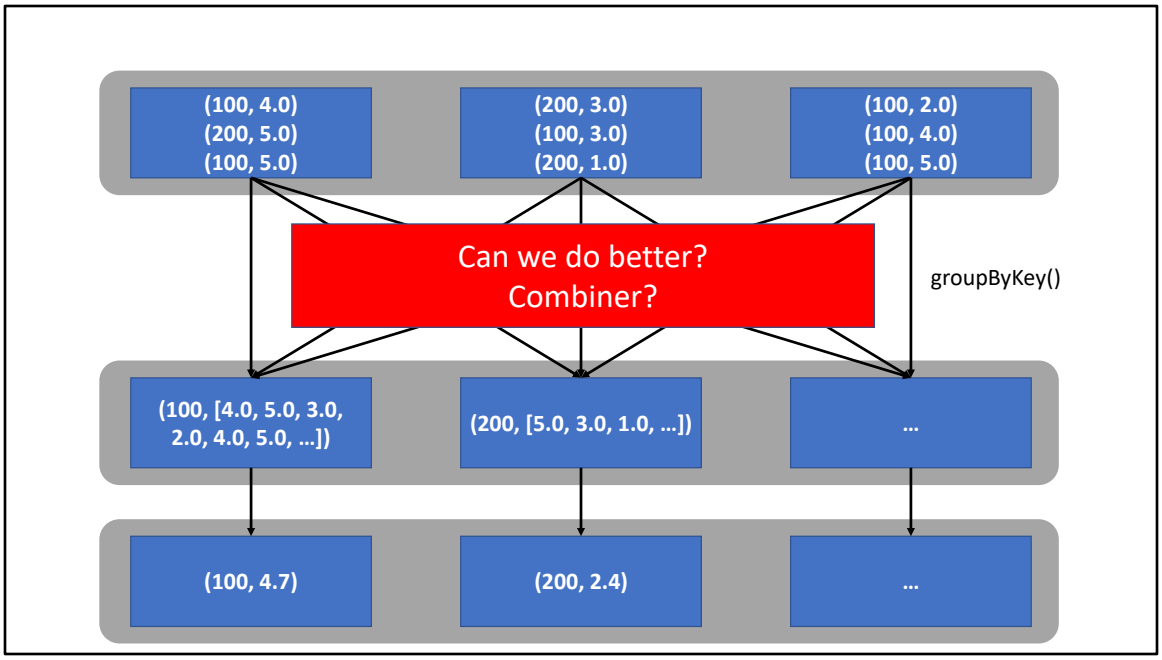
Ali Abedi

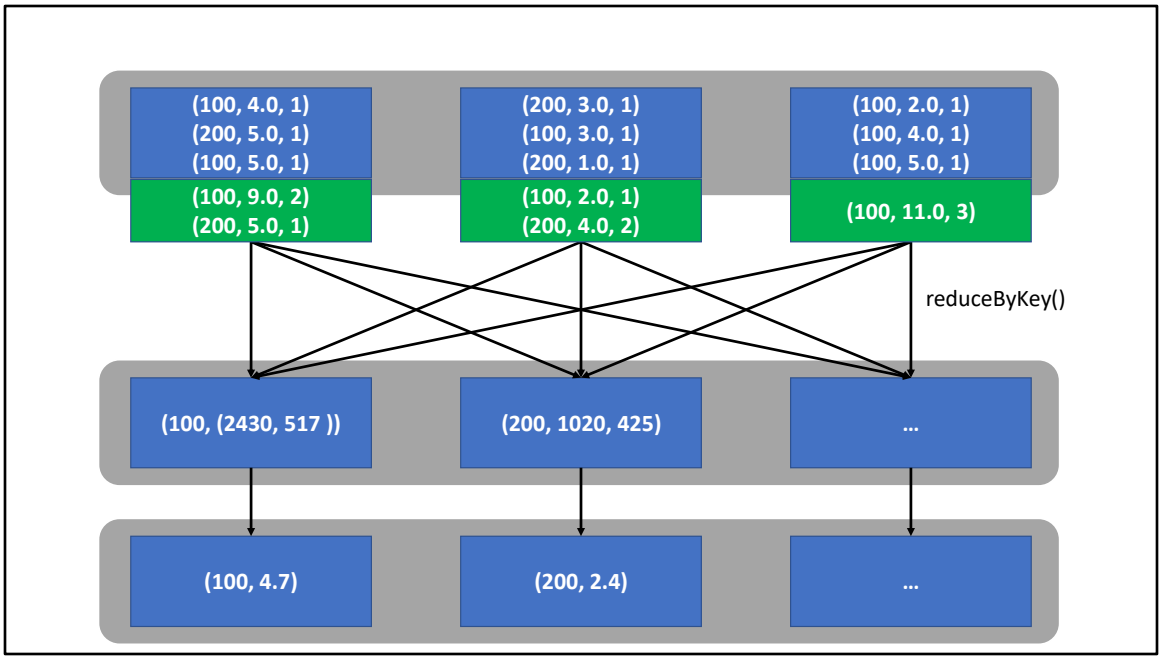
These slides are available at https://www.student.cs.uwaterloo.ca/~cs451/

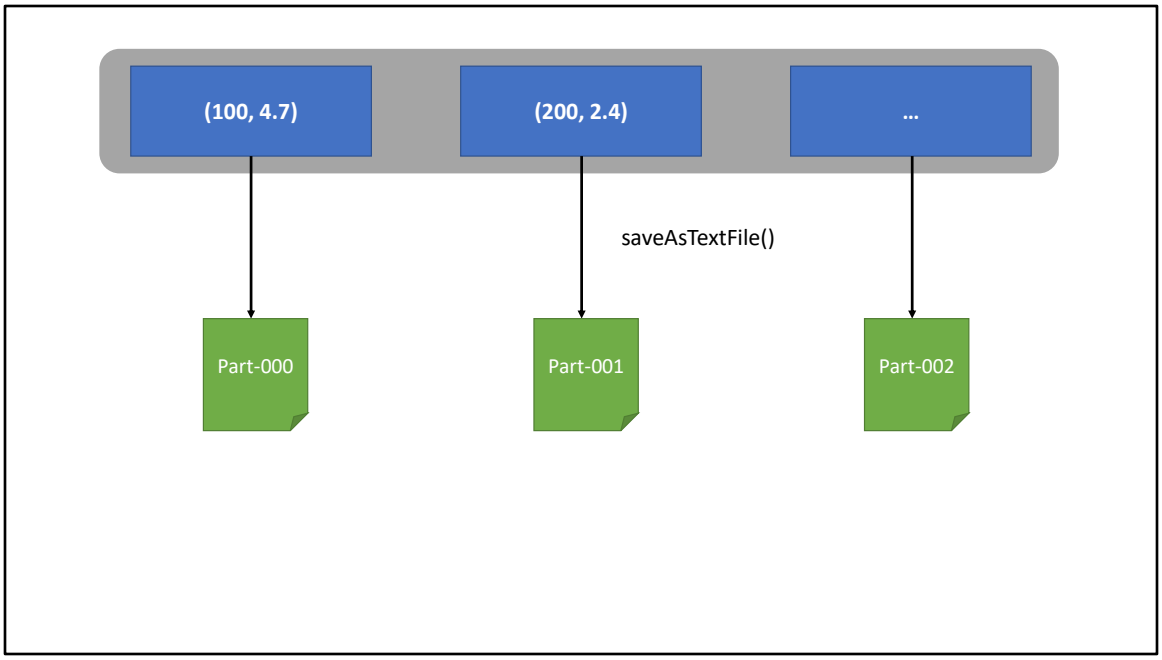
Movie rating example

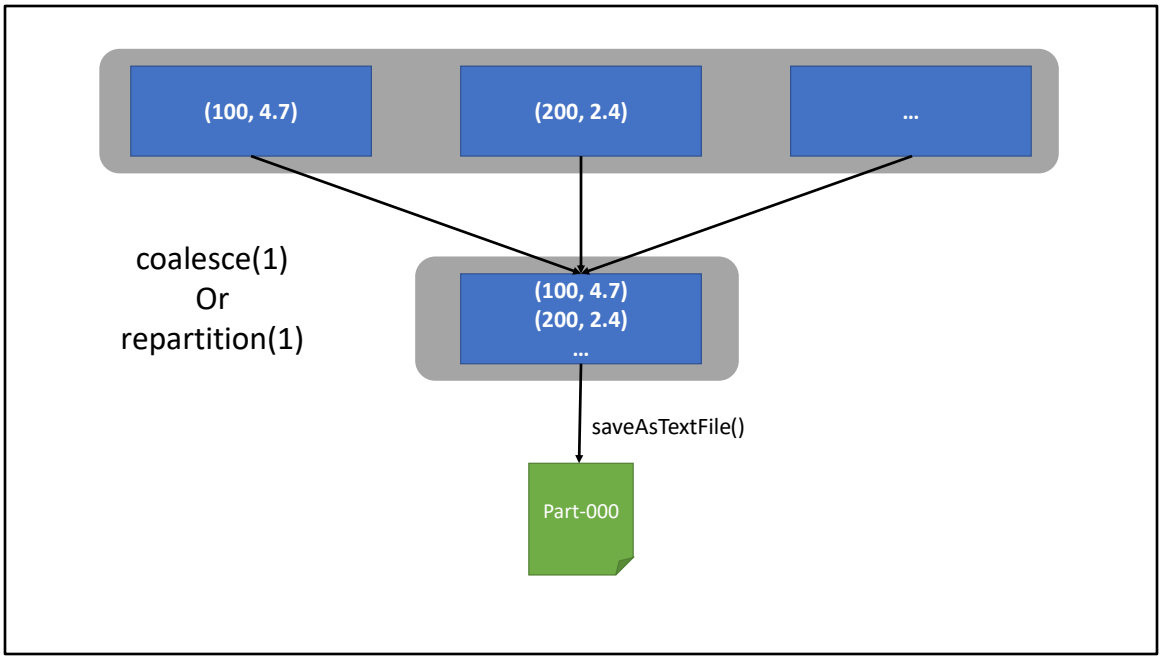

RDD
"1, 100, 4.0, …"
"1, 200, 5.0, …"
"2, 100, 5.0, …"
"60, 200, 3.0, …"
"61, 100, 3.0, …"
"61, 200, 1.0, …"
"80, 100, 2.0, …"
"81, 100, 4.0, …"
"82, 100, 5.0, …"
Map (lambda x:x.split(","))
(1, 100, 4.0, …)
(1, 200, 5.0, …)
(2, 100, 5.0, …)
(60, 200, 3.0, …)
(61, 100, 3.0, …)
(61, 200, 1.0, …)
(80, 100, 2.0, …)
(81, 100, 4.0, …)
(82, 100, 5.0, …)
Map (lambda x:(x[1], x[2]]))
(100, 4.0)
(200, 5.0)
(100, 5.0)
(200, 3.0)
(100, 3.0)
(200, 1.0)
(100, 2.0)
(100, 4.0)
(100, 5.0)


(100, 4.0)
(200, 5.0)
(100, 5.0)
(200, 3.0)
(100, 3.0)
(200, 1.0)
(100, 2.0)
(100, 4.0)
(100, 5.0)
groupByKey()
(100, [4.0, 5.0, 3.0, 2.0, 4.0, 5.0, ...])
(200, [5.0, 3.0, 1.0, ...])
...
(100, 4.7)
(200, 2.4)
...


(100, 4.0)
(200, 5.0)
(100, 5.0)
(200, 3.0)
(100, 3.0)
(200, 1.0)
(100, 2.0)
(100, 4.0)
(100, 5.0)
Can we do better?
Combiner?
groupByKey()
(100, [4.0, 5.0, 3.0, 2.0, 4.0, 5.0, …])
(200, [5.0, 3.0, 1.0, …])
…
(100, 4.7)
(200, 2.4)
…


(100, 4.0, 1)
(200, 5.0, 1)
(100, 5.0, 1)
(100, 9.0, 2)
(200, 5.0, 1)
(200, 3.0, 1)
(100, 3.0, 1)
(200, 1.0, 1)
(100, 2.0, 1)
(200, 4.0, 2)
(100, 2.0, 1)
(100, 4.0, 1)
(100, 5.0, 1)
(100, 11.0, 3)
reduceByKey()
(100, (2430, 517))
(200, 1020, 425)
...
(100, 4.7)
(200, 2.4)
...


(100, 4.7)
(200, 2.4)
...
saveAsTextFile()
Part-000
Part-001
Part-002



Repartition triggers shuffling but it gives more balanced partitions. It can be used to increase or decrease the number of partitions.
Coalesce can be used to only reduce the number of partitions. It avoids full shuffling so it is faster than repartition but it may give unbalanced partitions.

Other topics …

- Cache vs Persist
- Broadcast variables vs Accumulators