University of Augsburg, Institute of Computer Science
Prof. Dr. P. Fischer
J. Kastner, L. Rudenko

SS 2019
20. May 2019
Solution 3

Analyzing Massive Data Sets

Exercise 1: MapReduce - Joins (live)

The solution was discussed in the exercise.

Exercise 2: Understanding Spark (homework)

a) Sequence a): *Map* and *filter* - **narrow** dependencies, *groupBy* - **wide** one.

To restore the lost data we can reuse the RDDs marked with '*'. We need to recompute one partition (which was not persisted) from RDD B and also one partition from RDD C which is used to compute the data in the failed node.

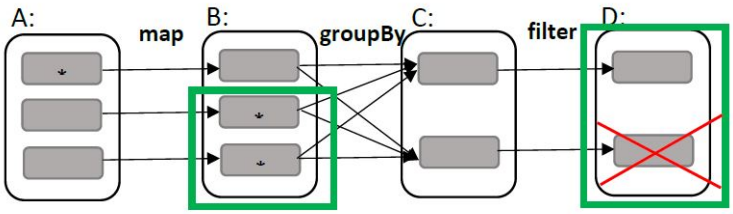

Abbildung 1: Sequence of Spark Operations a)

b) Sequence b): *join* - **narrow** dependency on the A RDD-side, **wide** one on the B RDD-side, *union* - **narrow**, *aggregateByKey* - in general case - **wide**, in this very special case (there is only one child partion, e.g. forced by argument numPartitions=1) - **narrow** dependency.

To restore the lost data we can reuse the RDDs marked with '*'. We need to recompute one partition (which was not persisted) from RDD D and three partitions from RDD E (they were also not persisted). Afterwards we can compute the lost data in the failed node.

Join operation: A to D join is a narrow dependency due to **join with fitting partitions**: All entries with the same key are on the same partition. B to D join is wide dependency because no fitting exists.

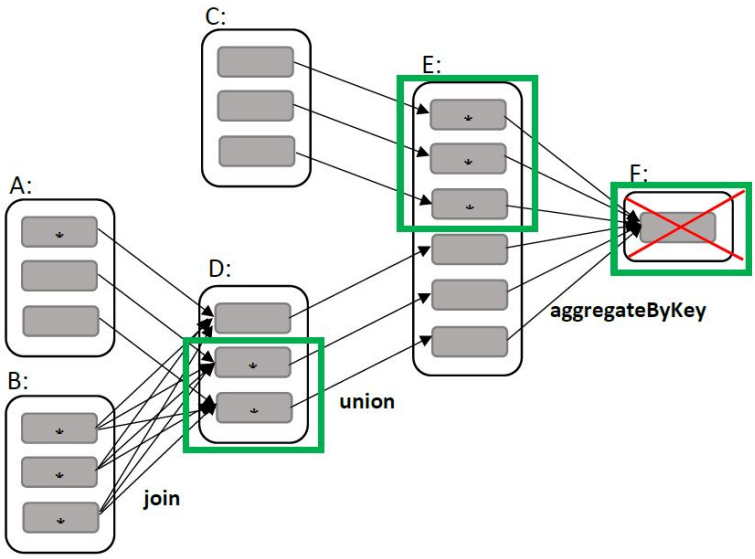

Abbildung 2: Sequence of Spark Operations b)

c) Sequence c): *Intersection* - **wide** dependency, *filter* and *union* - **narrow** ones.

To restore the lost data we can reuse the RDDs marked with '*'. We need to recompute the partitions from RDD F which was not persisted. Then the data in the failed nodes can be restored.

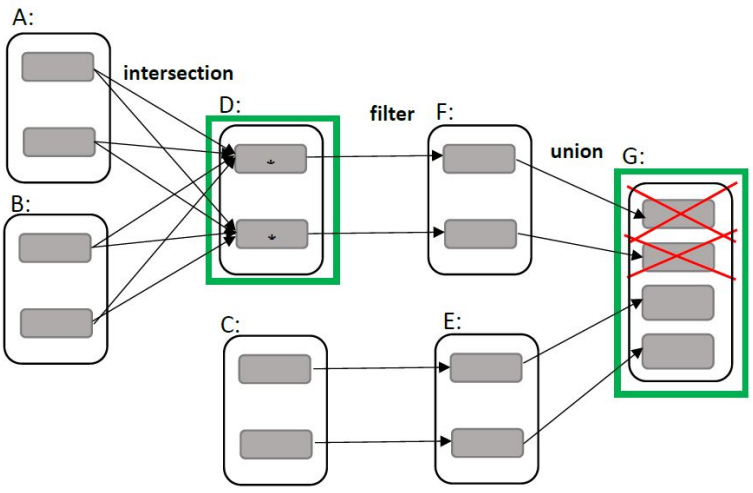

Abbildung 3: Sequence of Spark Operations c)

Exercise 3: Spark (live)

The solution was discussed in the exercise.