dimajix

Big Data | Analytics | Algorithms

Spark Runtime Architecture

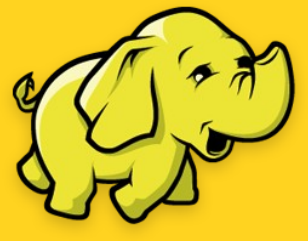
dimajix | Dr. Kaya Kupferschmidt | k.kupferschmidt@dimajix.de

Spark is Fast to Write

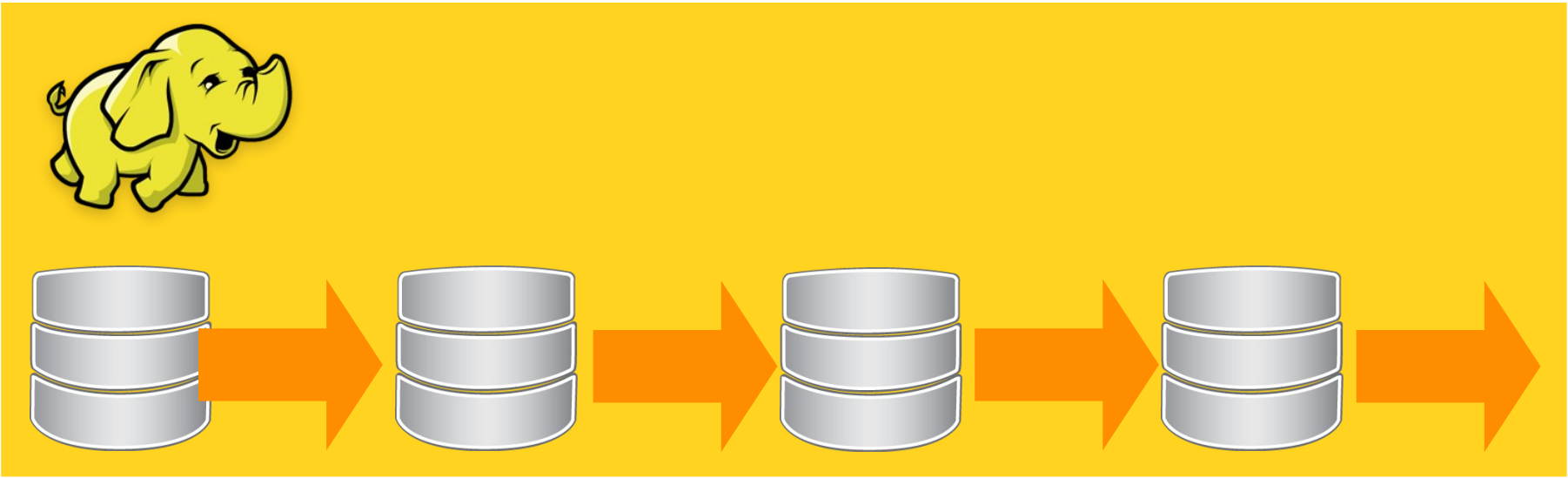
```
sc.textFile("alice")
  .flatMap(_.split(" "))
  .filter(_ != "")
  .map(x => (x,1))
  .reduceByKey(_ + _)
  .saveAsTextFile("alice_wordcount")
```

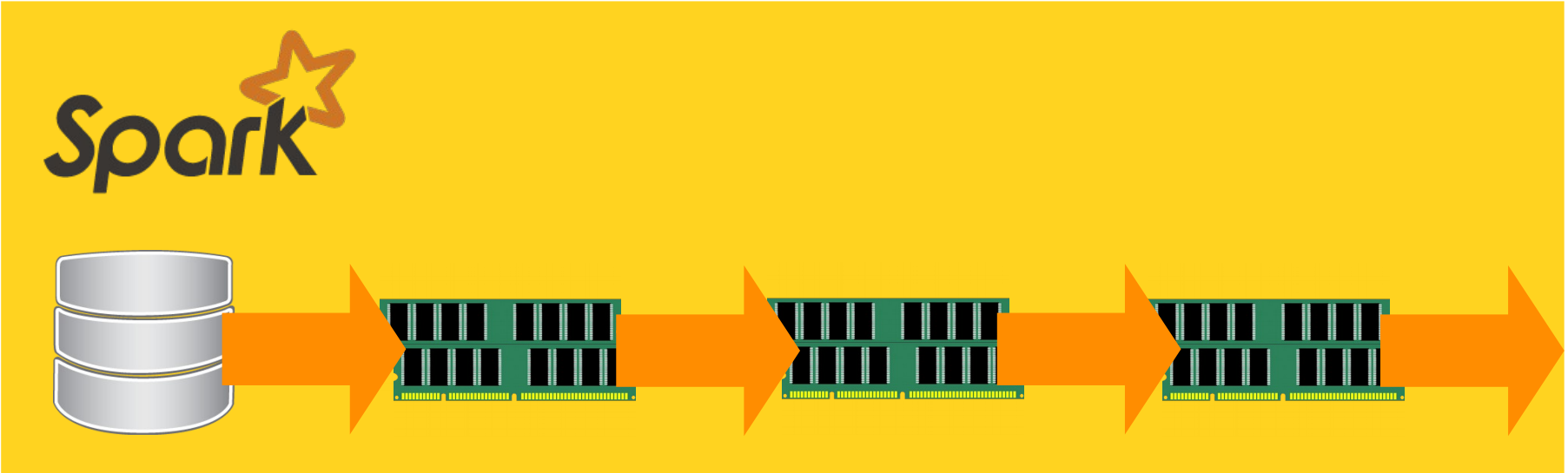
```
class WordCountMapper {
 ...
}
class WordCountReducer {
 ...
}

...
```



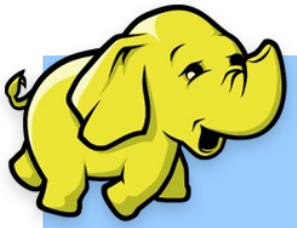
Spark is Fast to Run





Hadoop vs Spark

dimajix



Hadoop

- Data Sets are Files
- Map/Reduce
- Physical Plan= Logical Plan
- Every step involves I/O
- New JVM for every Task

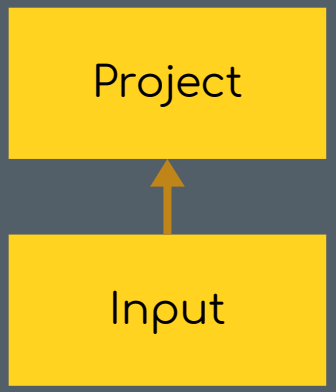
Spark

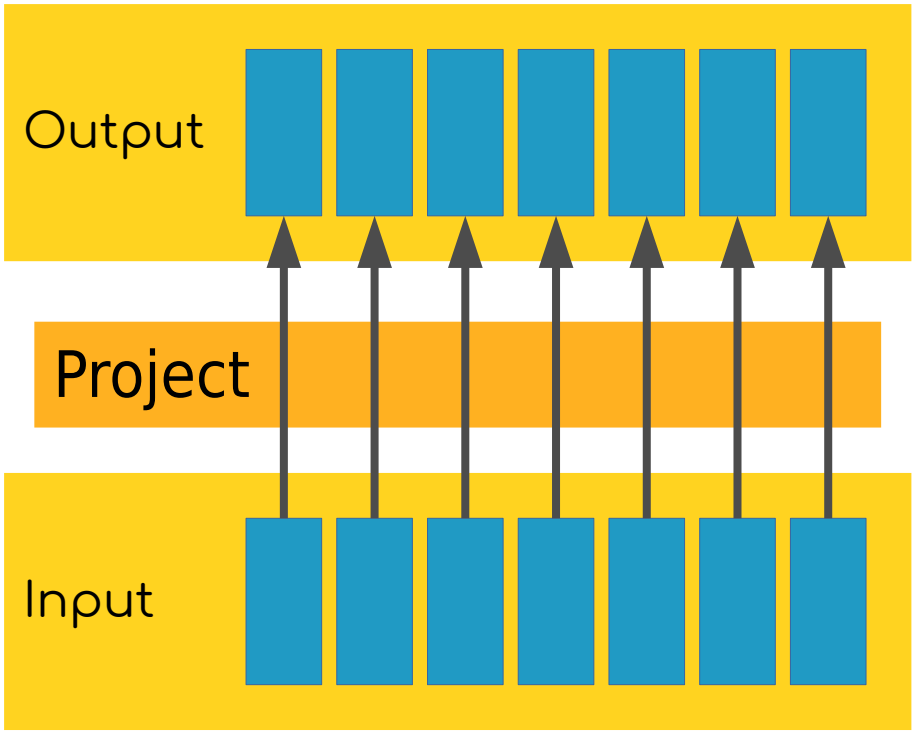
- Data Sets are RDDs
- RDD API
- Physical Plan is optimized
- Caching
- Long Living Executors

Spark

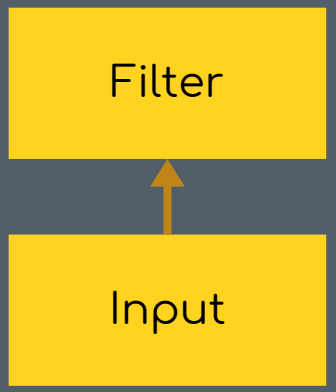
DataFrame Operations

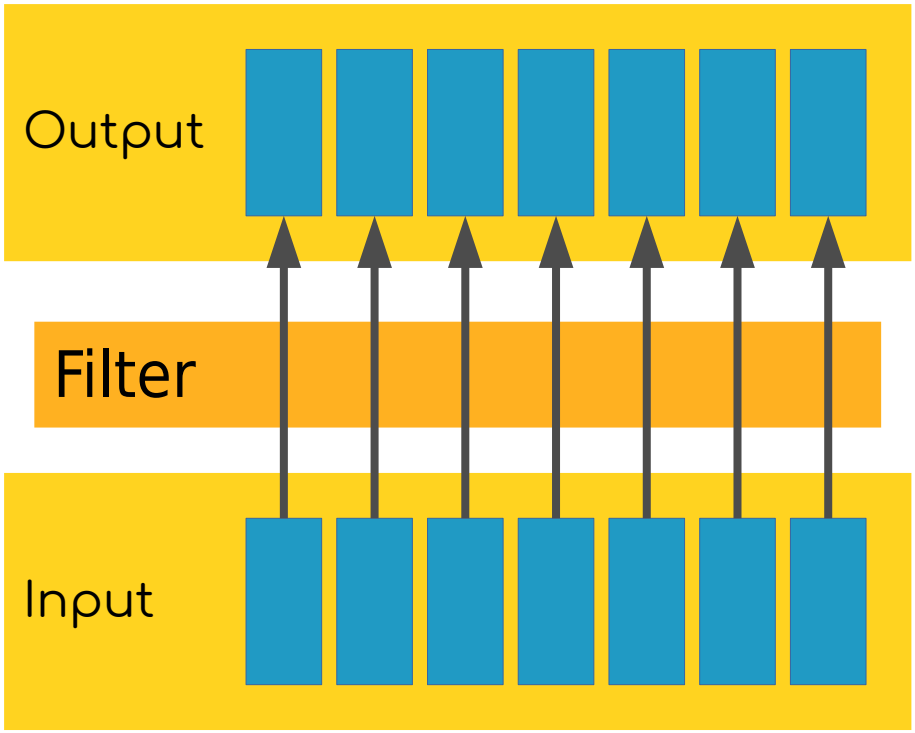
SELECT



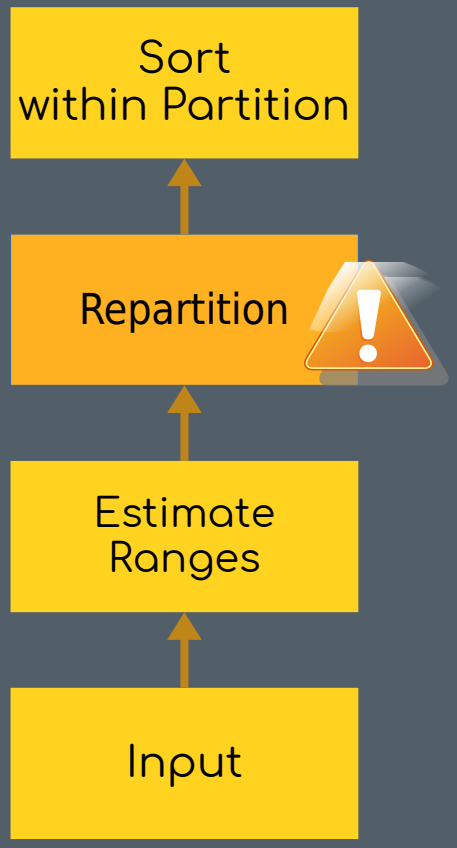


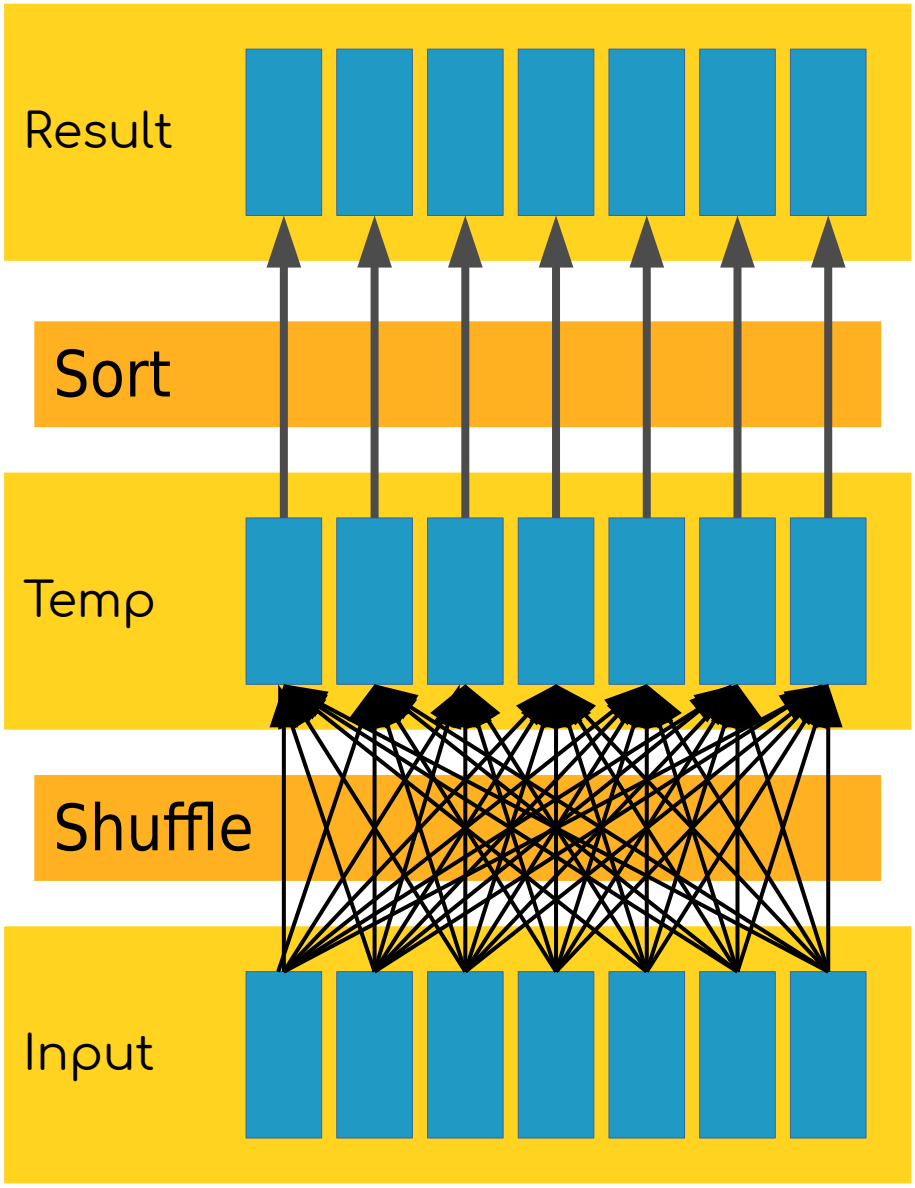
WHERE



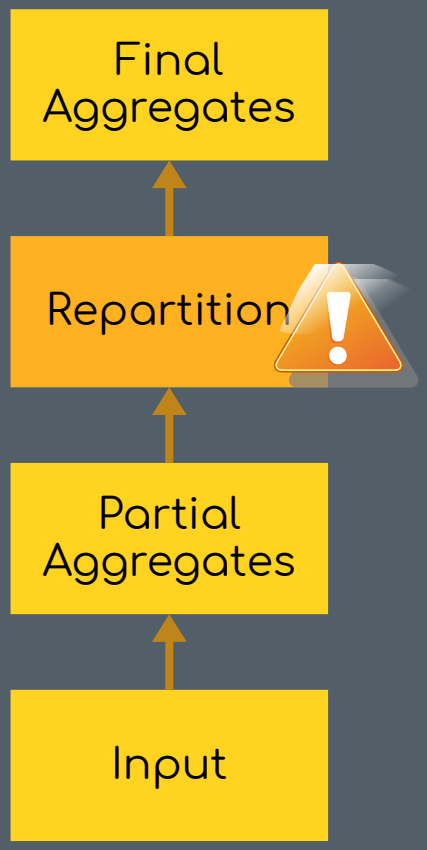


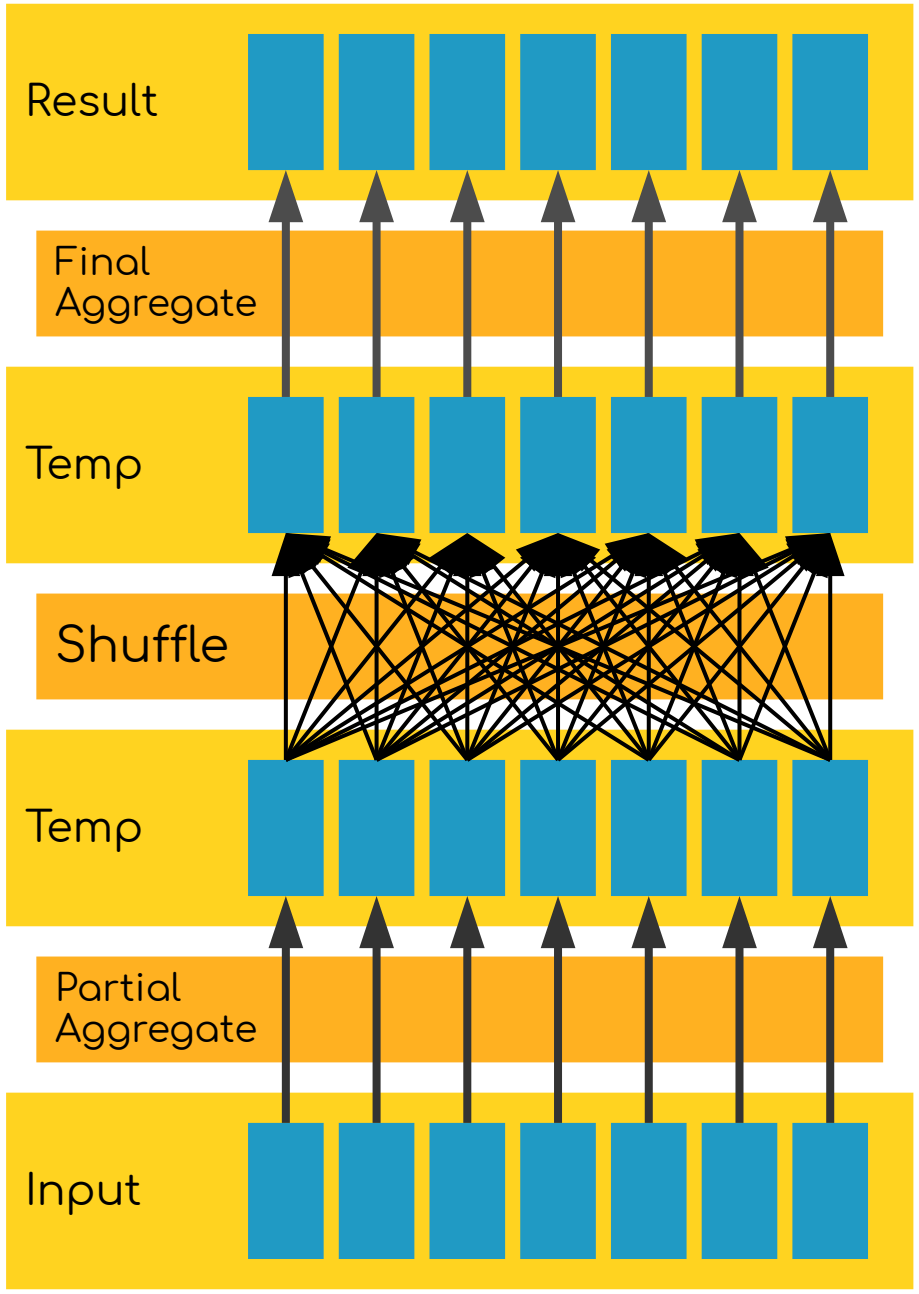
ORDER BY


Sort
within Partition
Repartition
Estimate
Ranges
Input

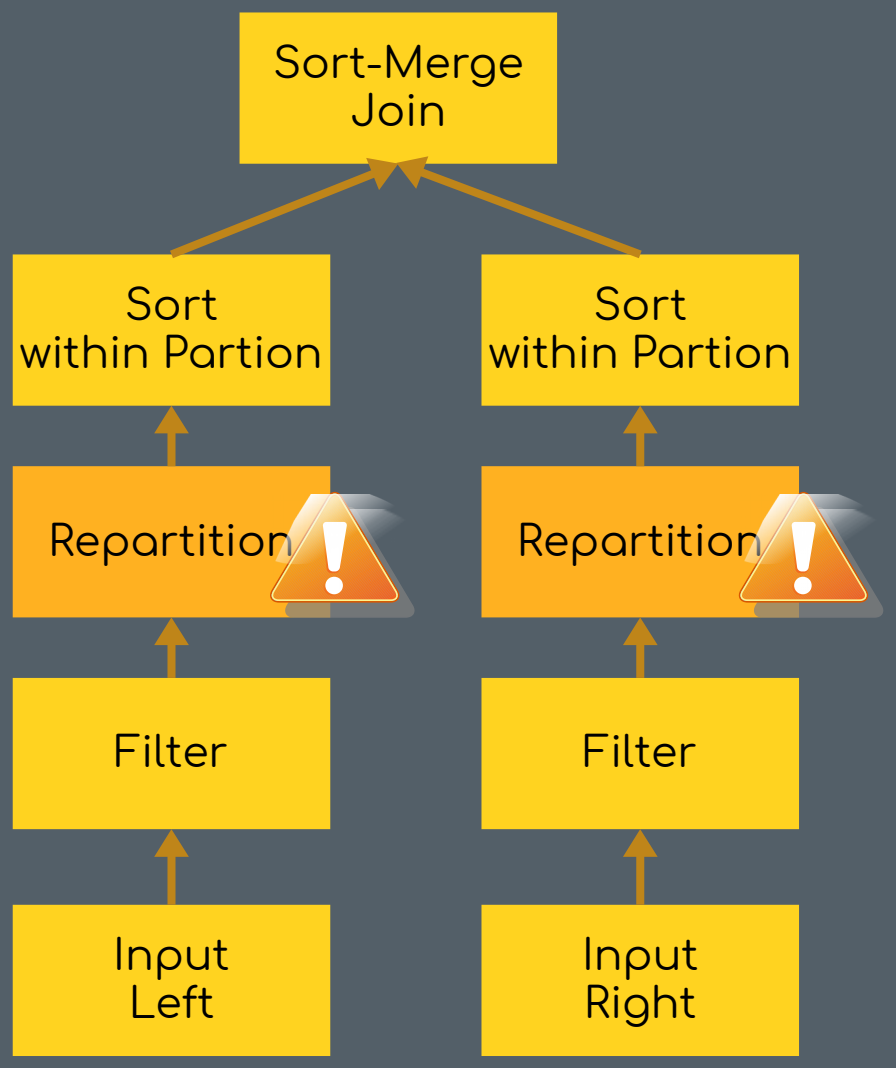

Result
Sort
Temp
Shuffle
Input

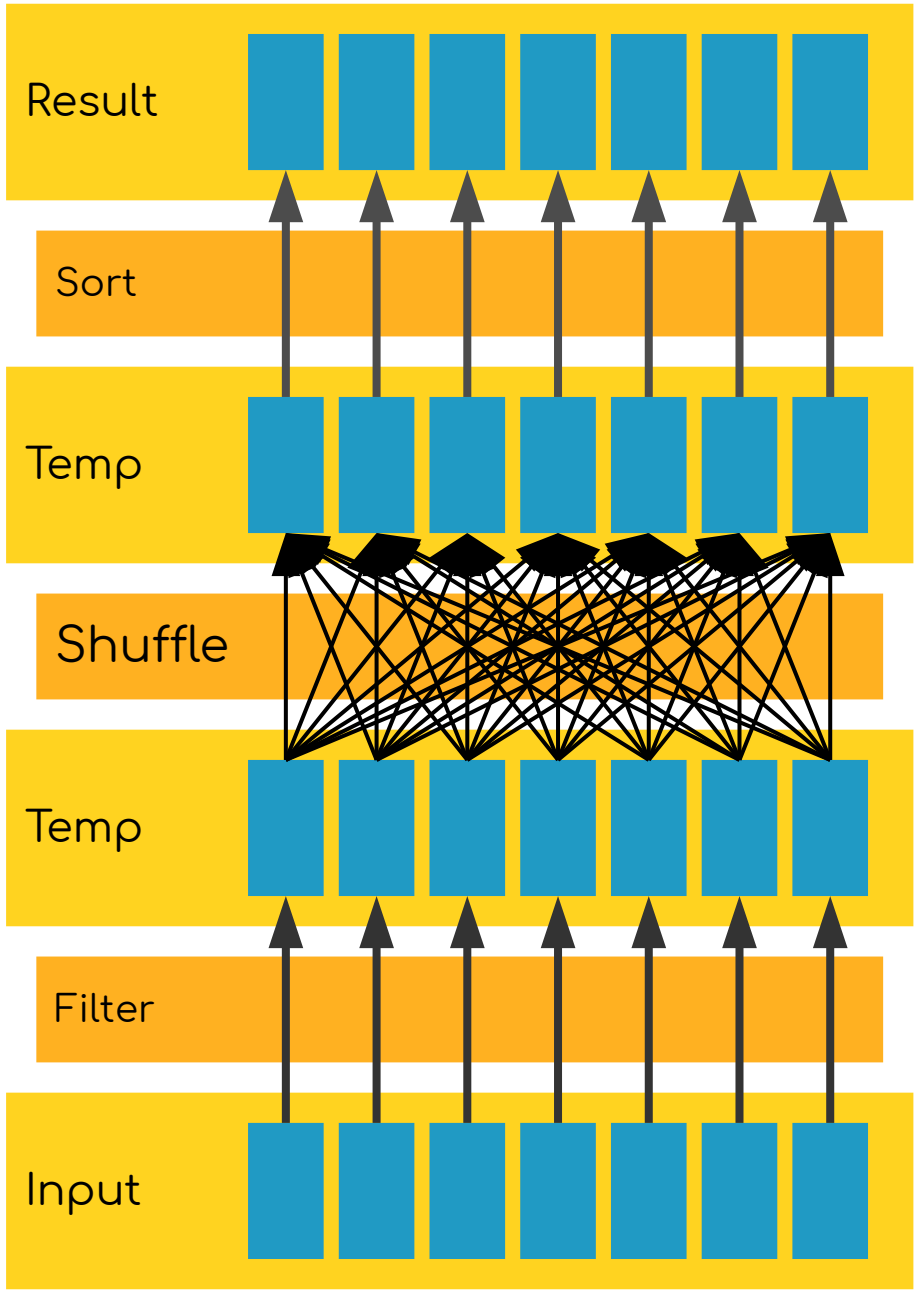
GROUP BY


Final Aggregates
Repartition
Partial Aggregates
Input

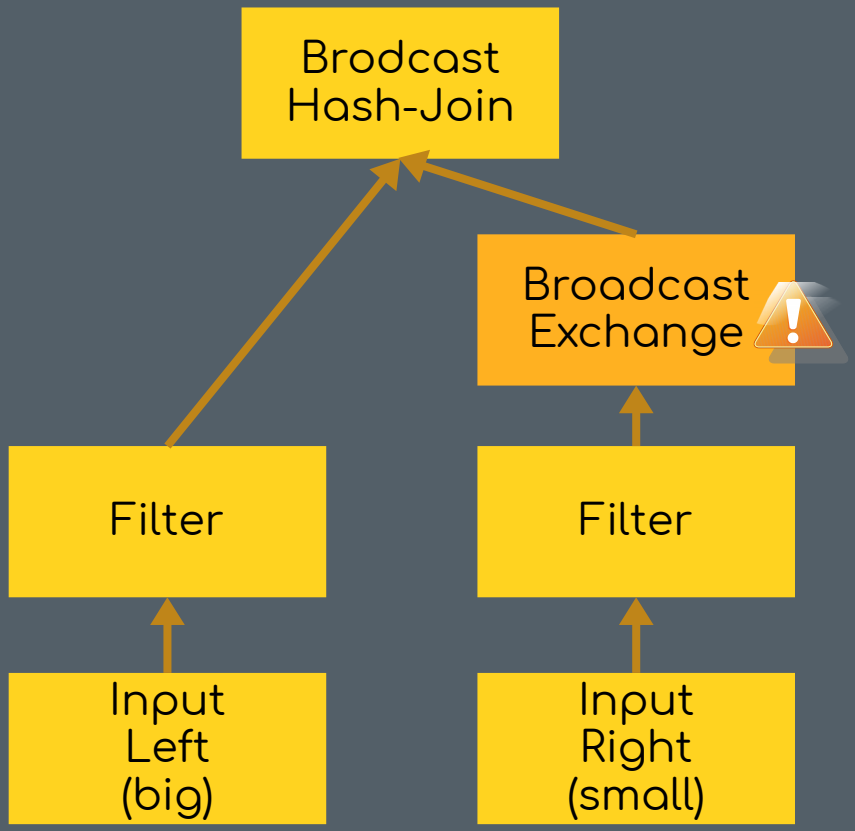

Result
Final Aggregate
Temp
Shuffle
Temp
Partial Aggregate
Input

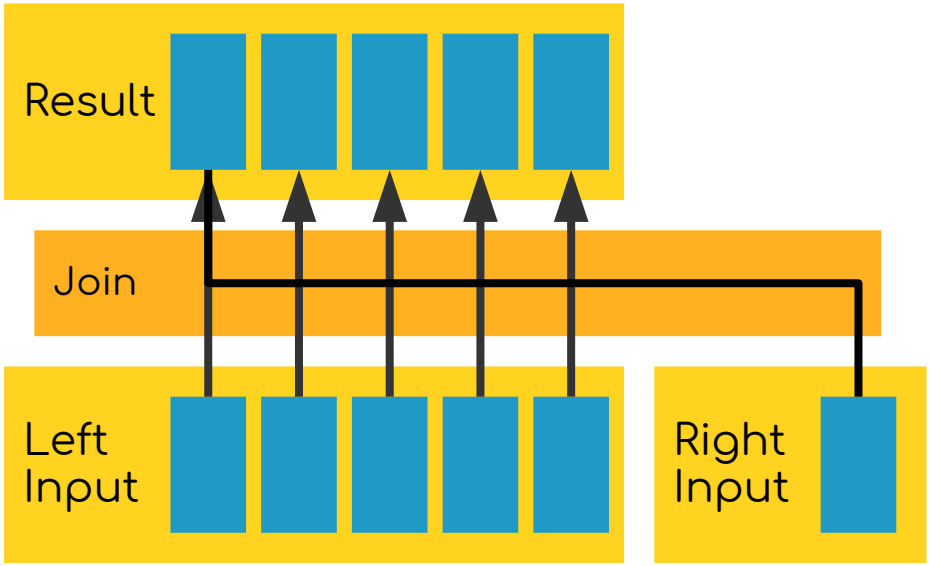
JOINs


Sort-Merge Join
Sort within Partion
Sort within Partion
Repartition
Repartition
Filter
Filter
Input Left
Input Right


Result
Sort
Temp
Shuffle
Temp
Filter
Input

Broadcast JOINs





Required Atoms

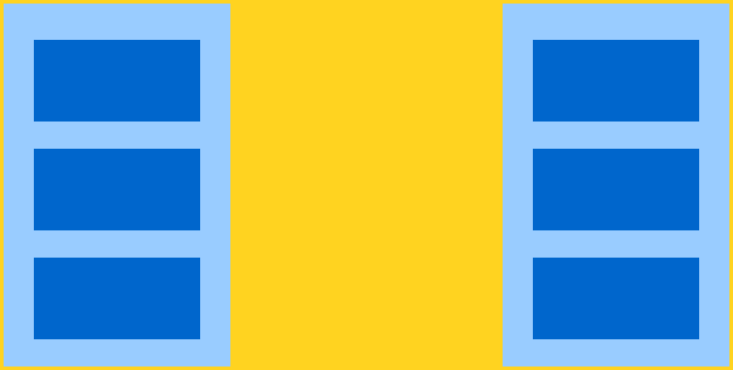
dimajix

Map Operation

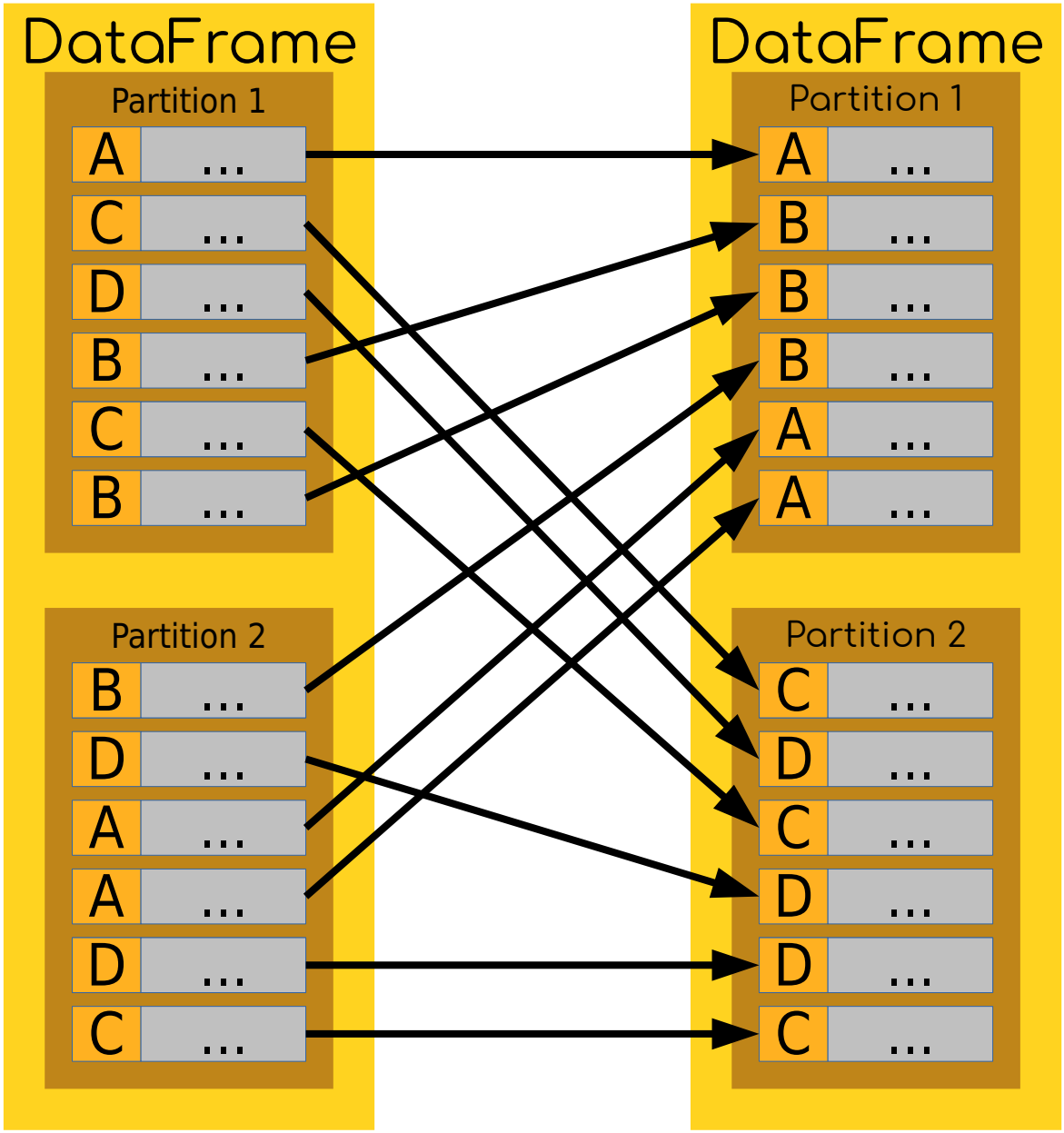
- Reads one record
- Applies any transformation
- Emits 0..n records
- Each output record depends on exactly one input record

Shuffle Operation

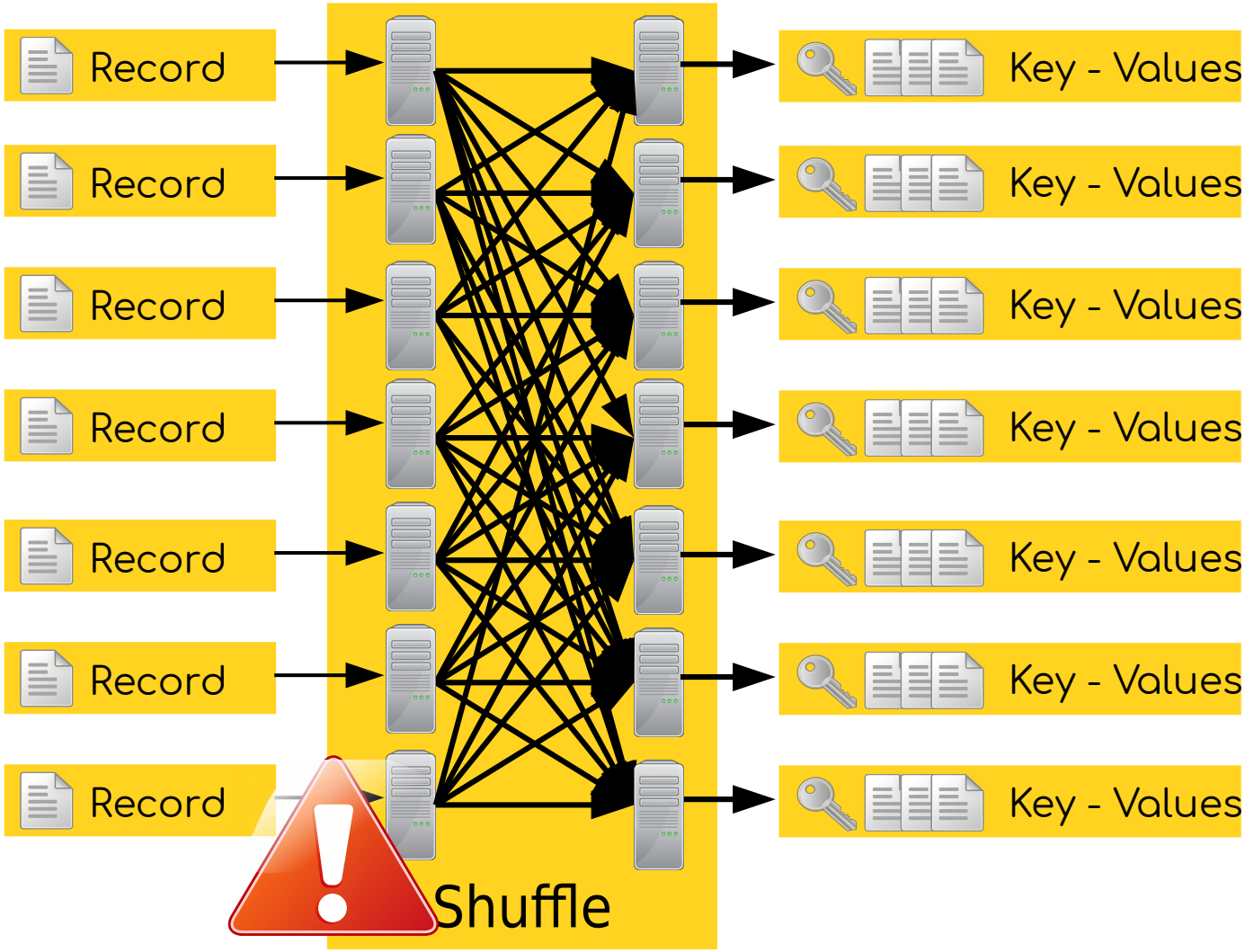
- Collects multiple records with same key
- Results are groups of records



Shuffle Operations

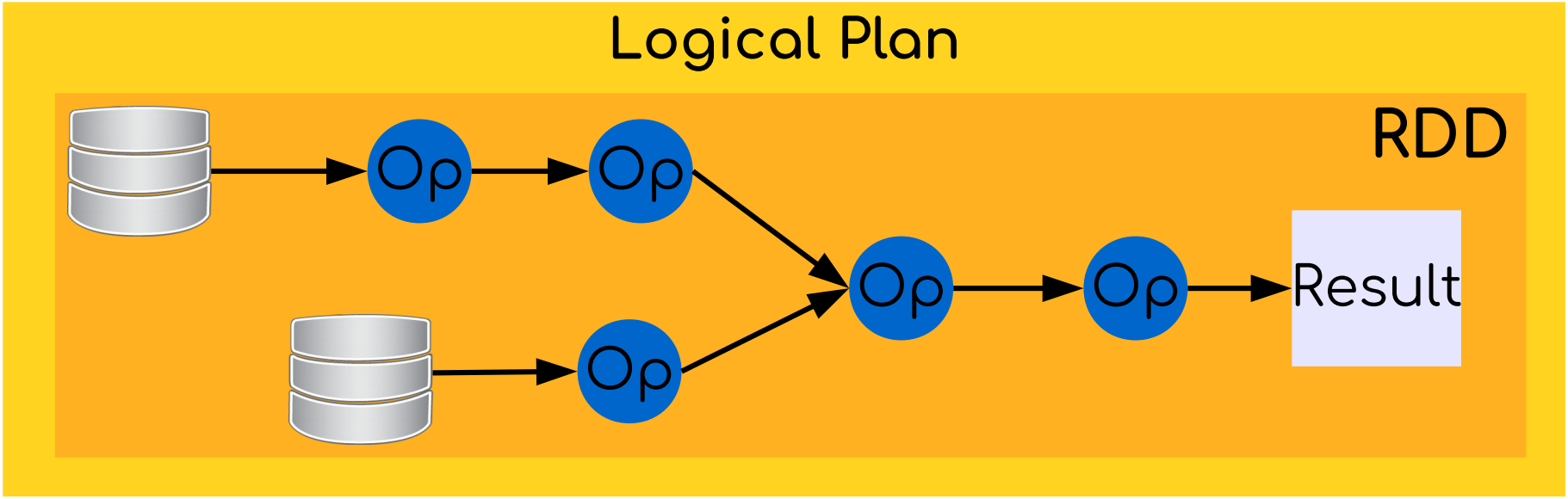


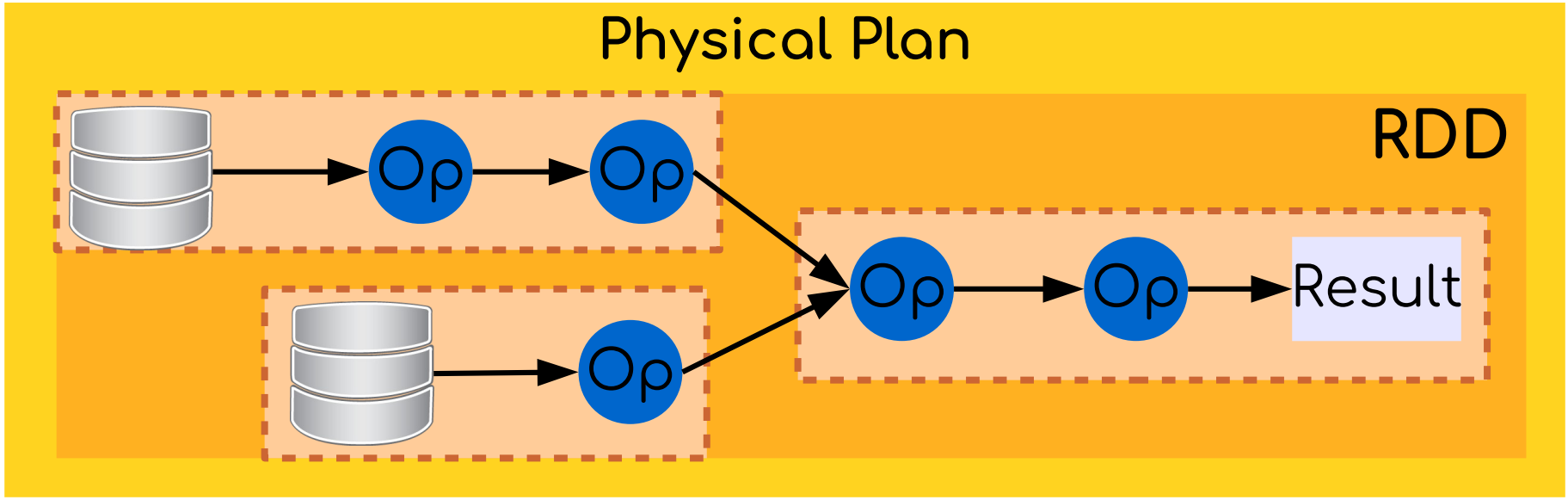
Shuffles in Clusters



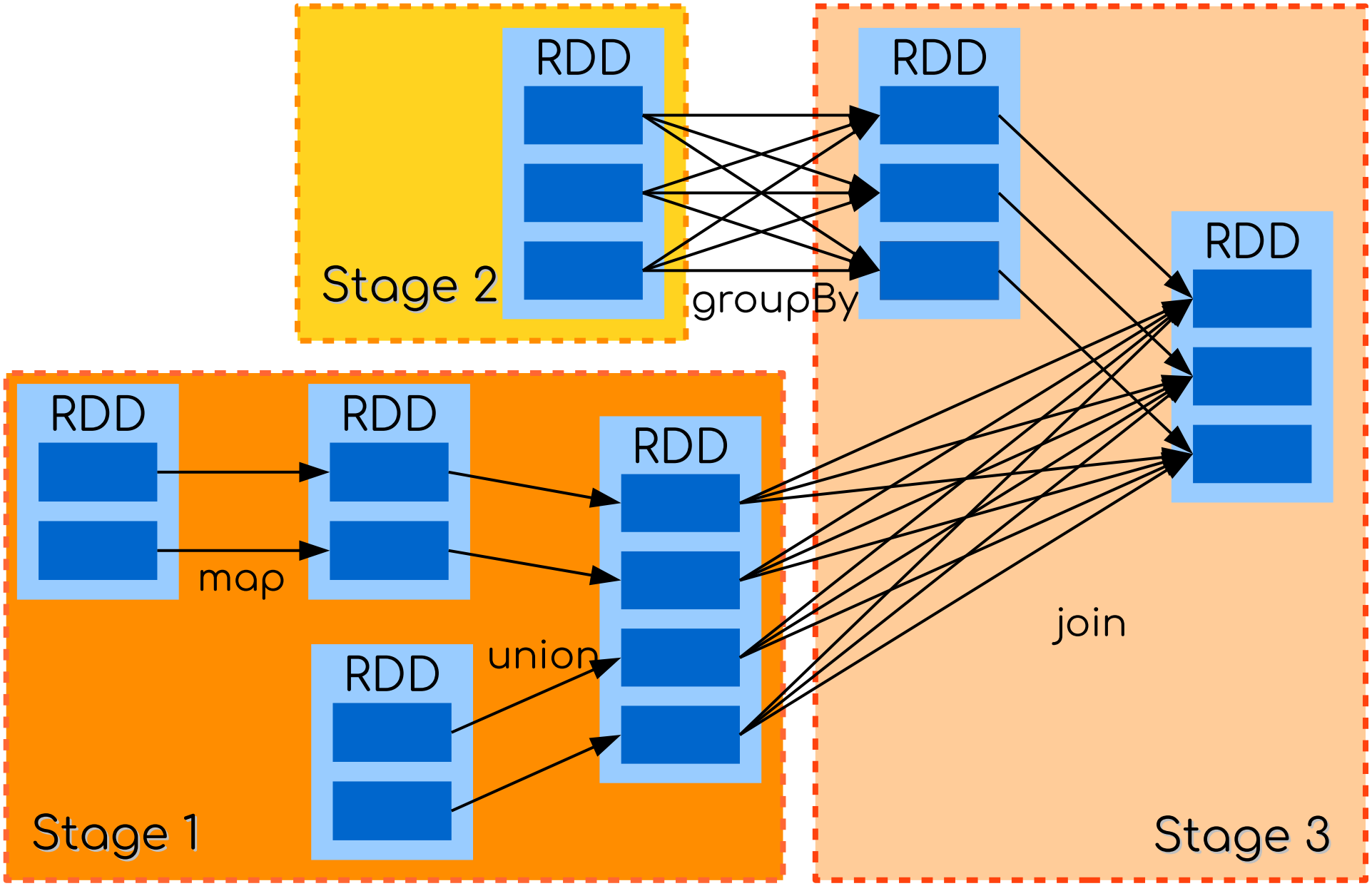
Spark Runtime Architecture

Execution Planning

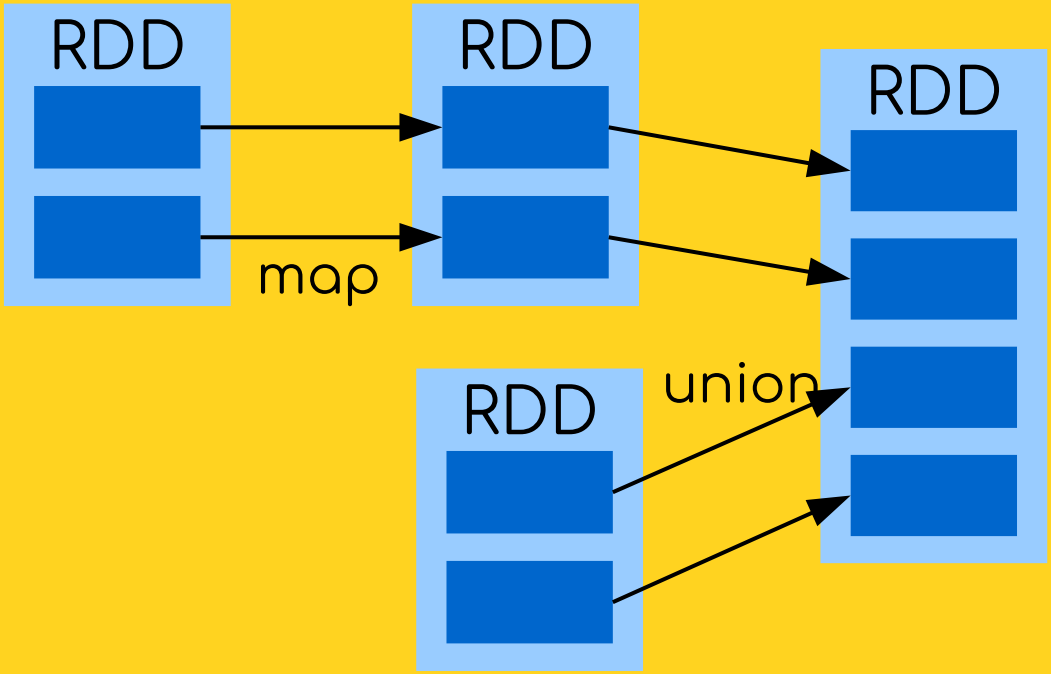




Spark Transformations

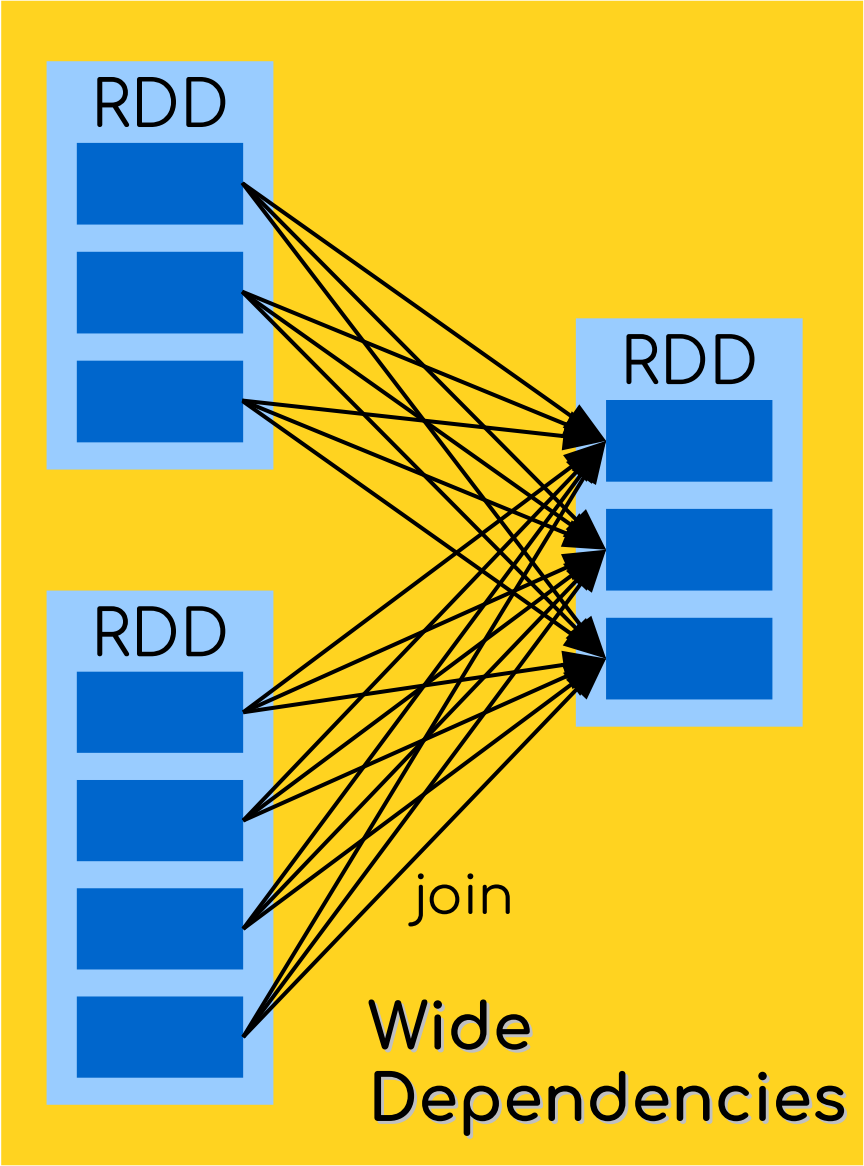


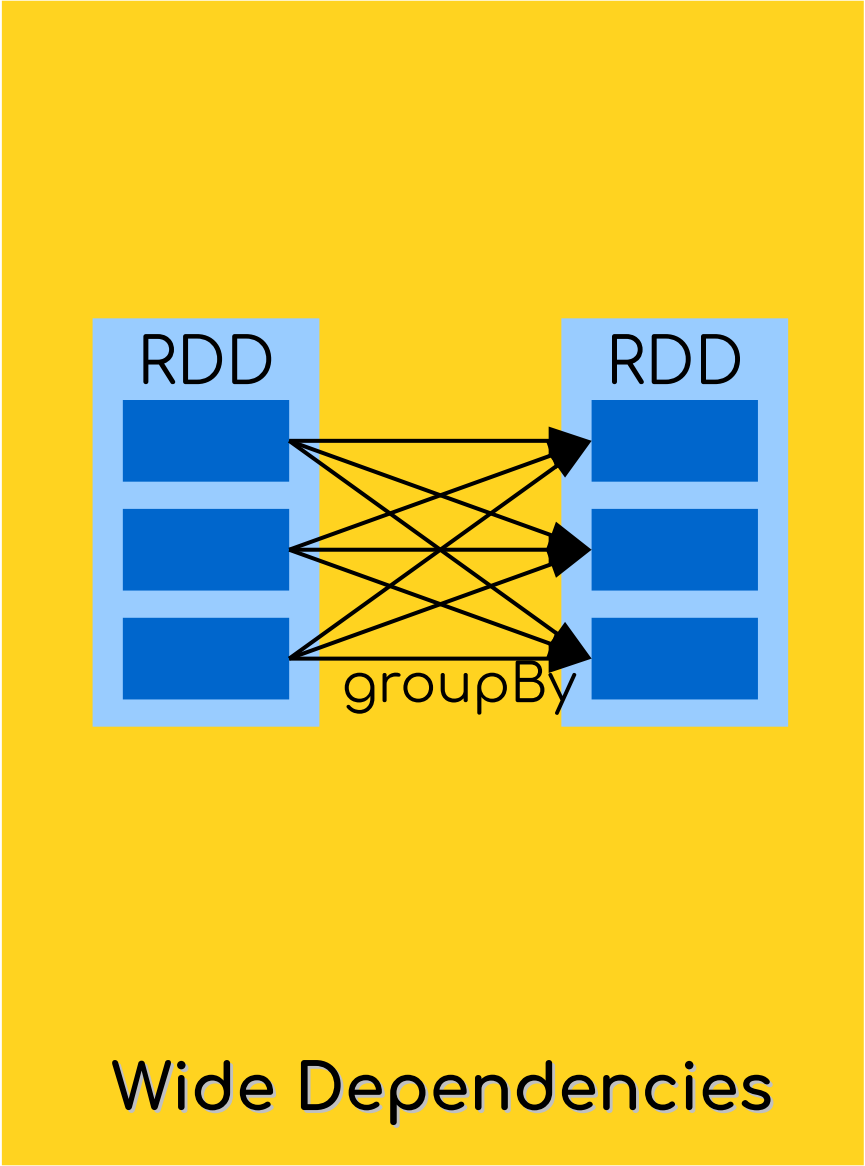
RDD Dependency Types



Narrow Dependencies

RDD Dependency Types

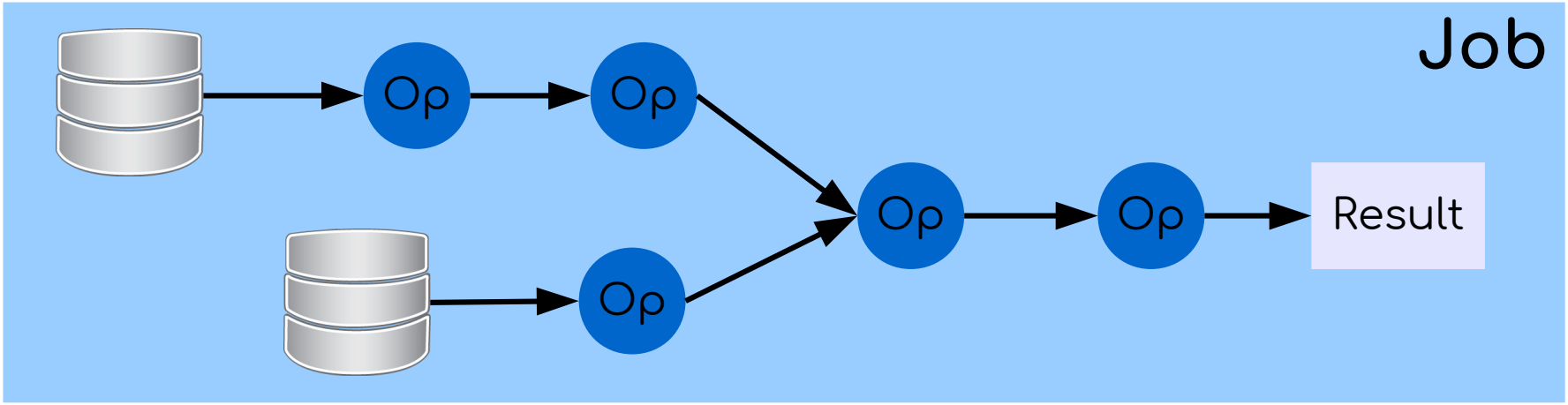




RDD
Execution Model

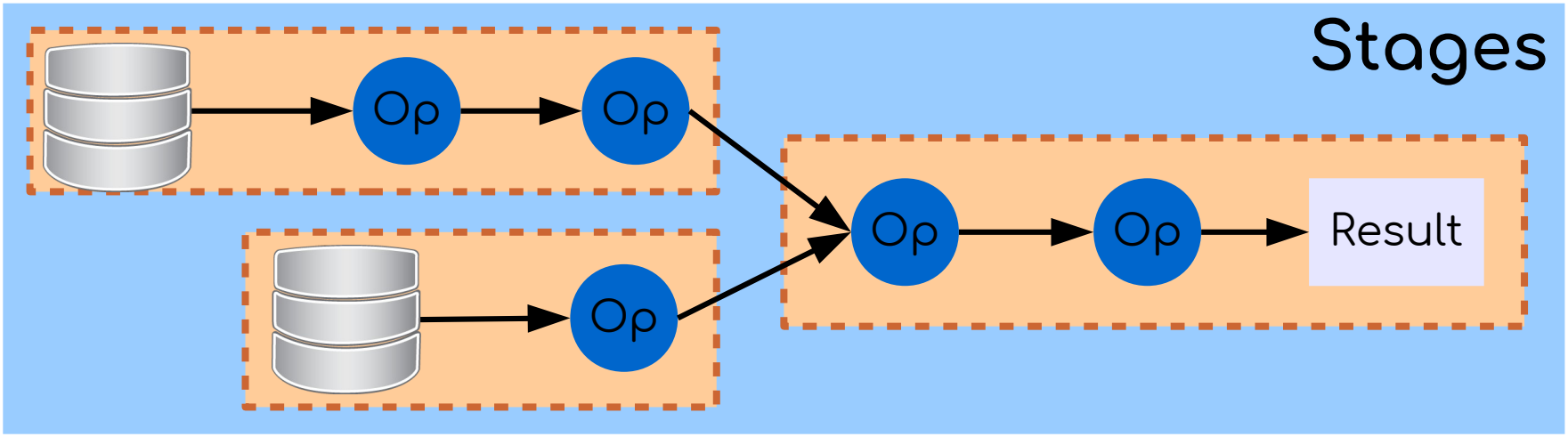
Spark Execution Model

1. A Spark Job is created by issuing a Spark Action



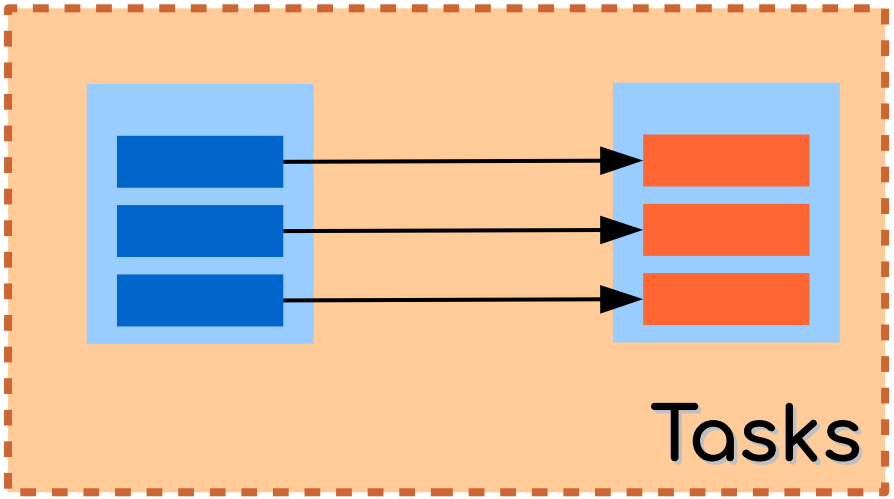
Spark Execution Model

2. The Job is dissected into Stages



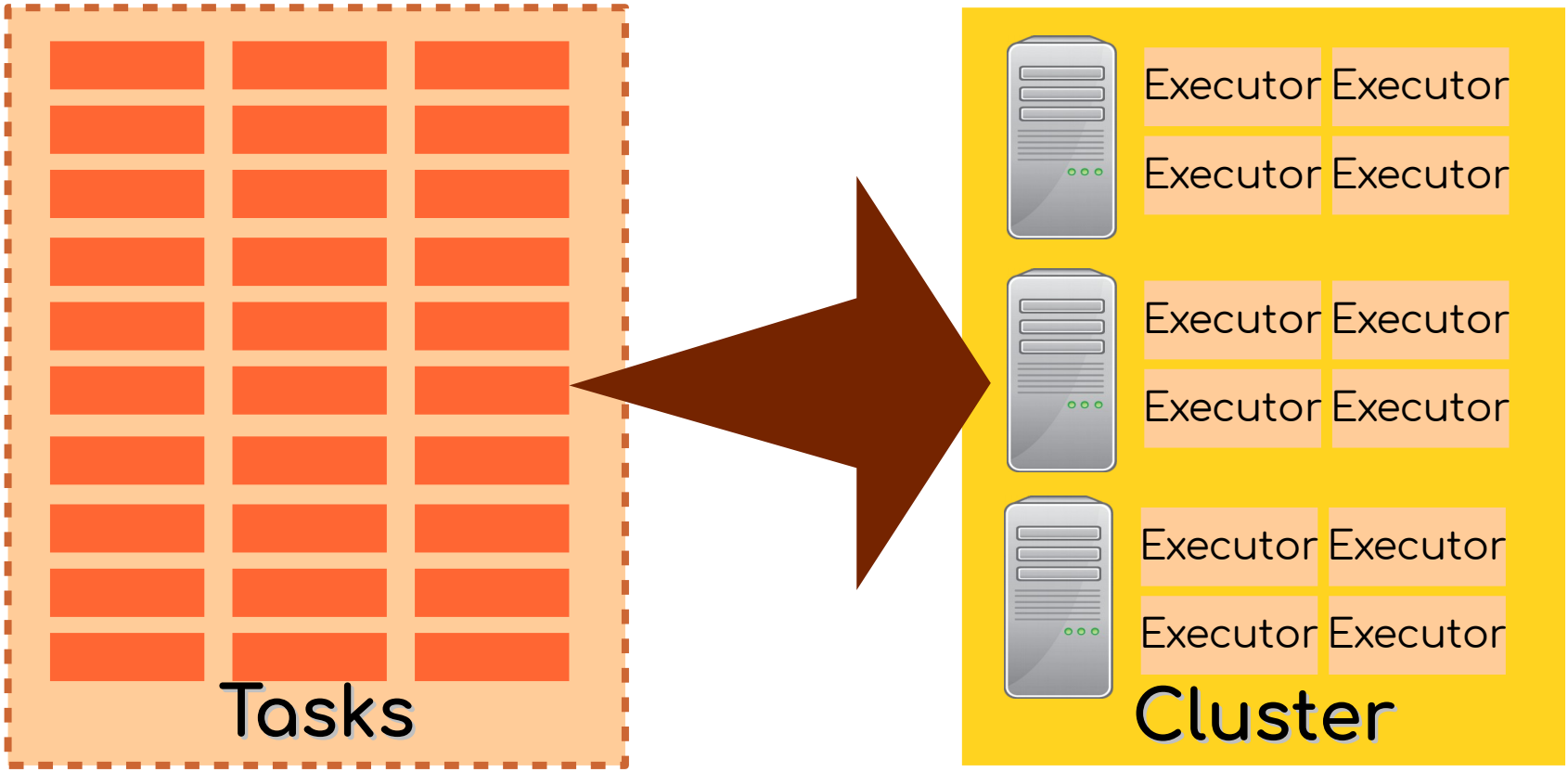
Spark Execution Model

3. Each Stage is split up into Tasks along Partitions



Spark Execution Model

4. Tasks are scheduled to Executors



Spark Execution Model

- Job: Work required to compute resulting RDD
- Stage: A wave of transformations inside a job, corresponding to one or more pipelined RDDs.
- Task: A unit of work within a stage, corresponding to one RDD partition.
- Shuffle: The transfer of data between stages.

DataFrames Execution Model

Spark Execution Model

dimajix

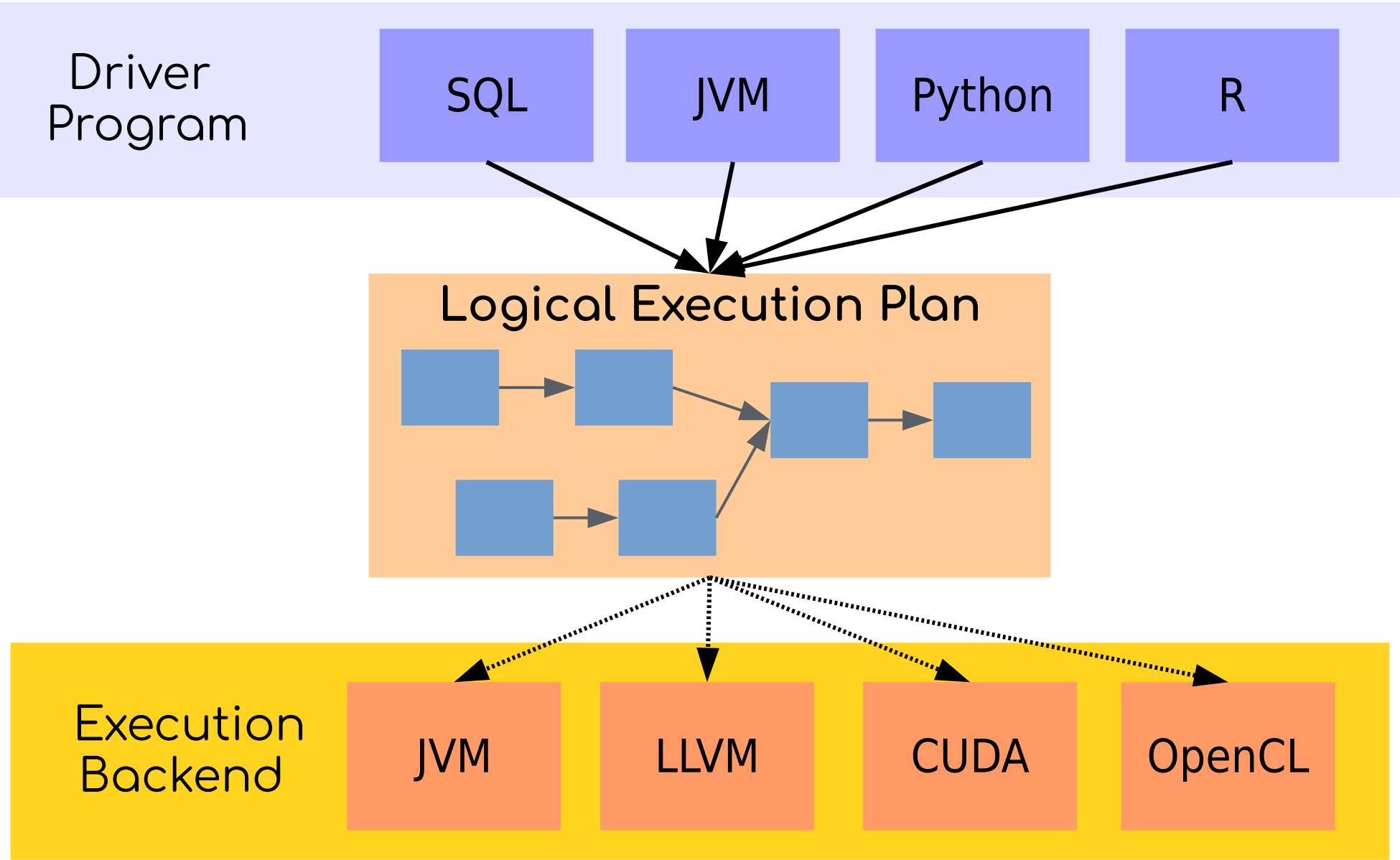
1. The required transformations are extracted into a ***parsed logical execution plan***

2. The logical plan is analysed and all dependencies are resolved. The result is an ***analyzed logical execution plan***.

3. The analyzed logical plan is optimized. The result is an ***optimized logical execution plan***.

4. The optimized logical plan is broken down into possibly multiple jobs and stages. This is the ***physical execution plan***.

DataFrame Execution



Spark on YARN

Runtime Architecture