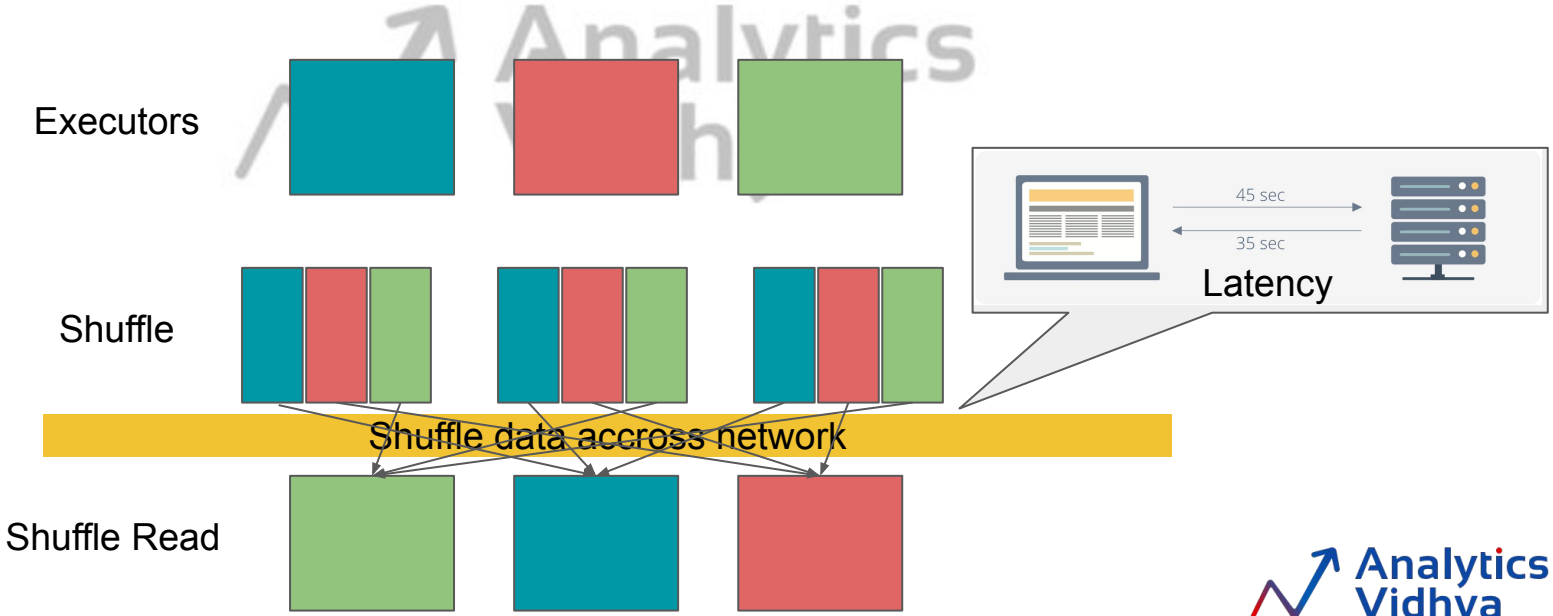
Shuffling

Shuffling

- A **shuffle** occurs when data is rearranged between partitions.
- When transformation requires information from other partitions.
- Data is gathered and combined into a new partition, likely on a different executor.



Latency Matters (Humanized)

Shared Memory

Seconds

L1 cache reference…….. 0.5s
L2 cache reference……..7s
Mutex Lock/Unlock…….25s

Minutes

Main memory reference…….1m 40s

Distributed

Days

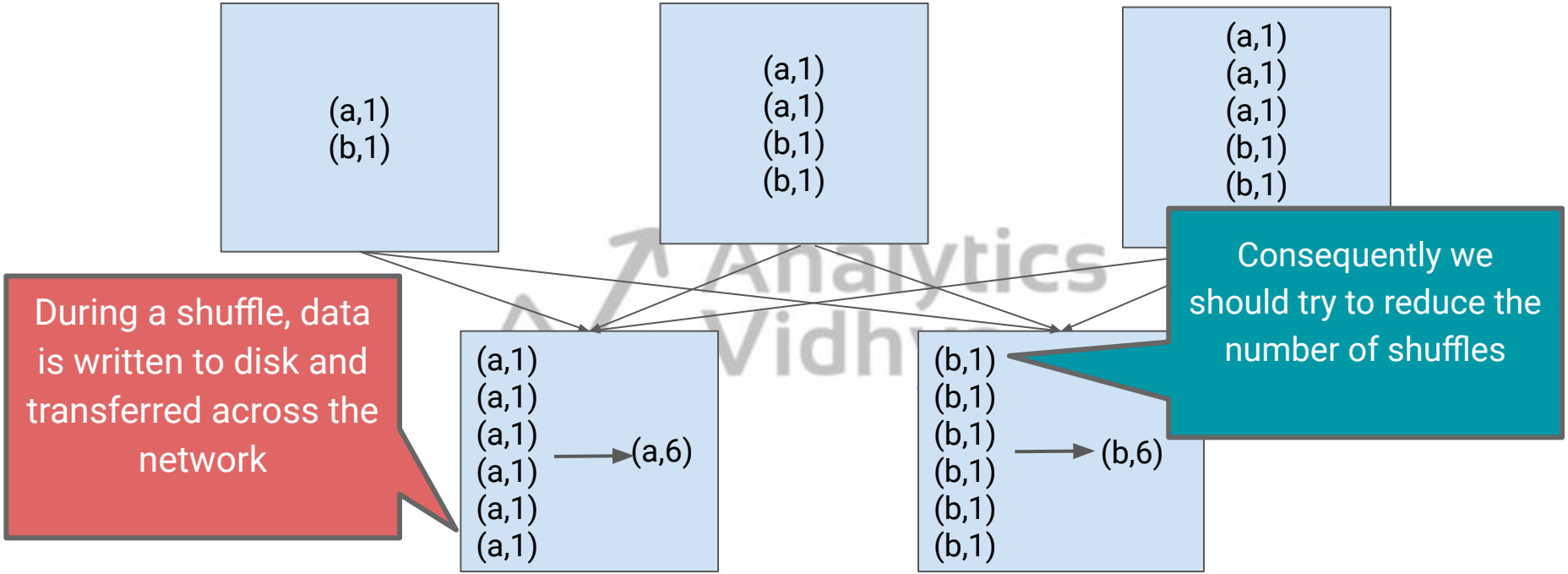
Round Trip within same datacenter…….5.8days

Years

Sent packet CA->Netherlands->CA…….4.8years

We don't want to be sending all our data over the network if its not absolutely required. Too much network communication kills performance.

GroupByKey: Shuffle



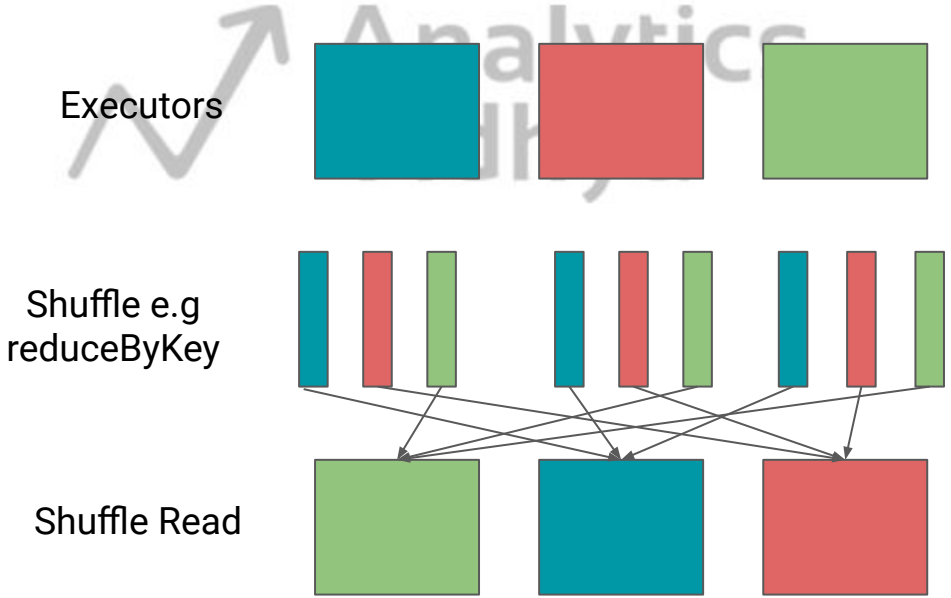
With GroupByKey, all the data is wastefully sent over the network and collected on the reduce workers.

Can we do a better Job?

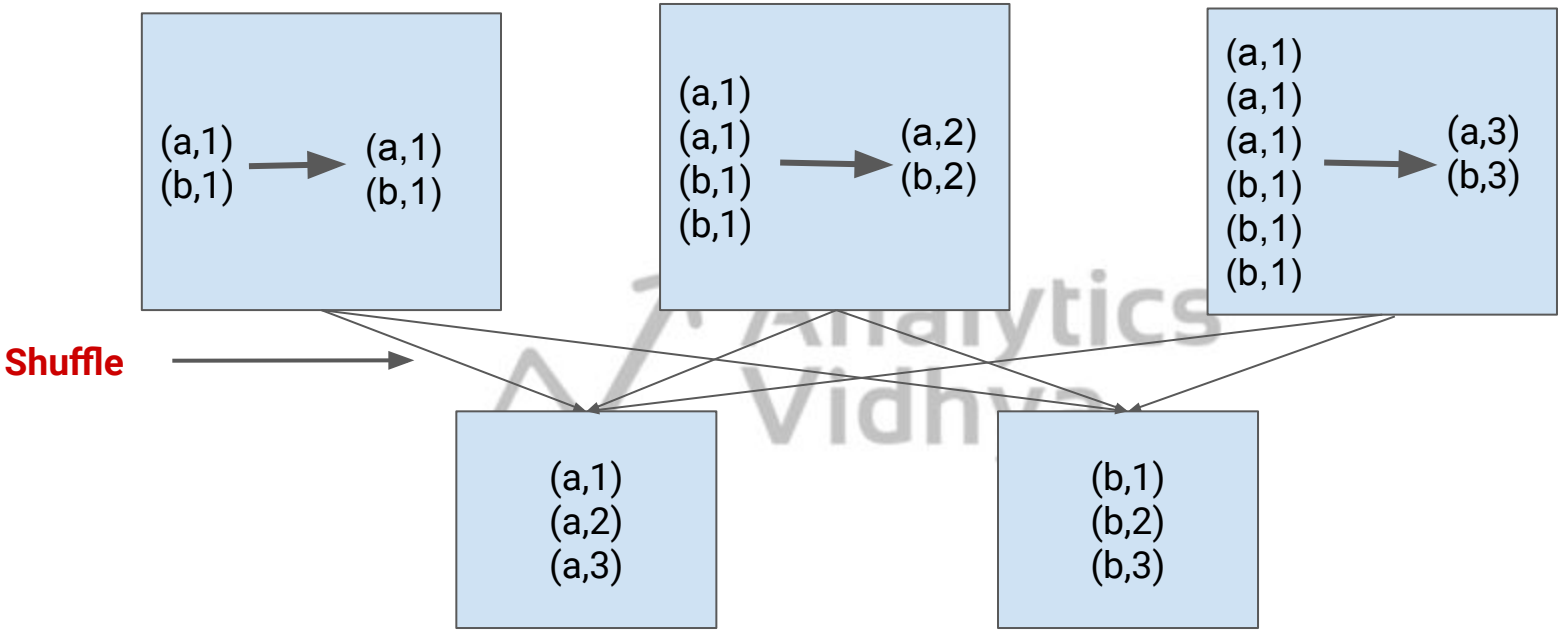
- Perhaps we don't need to send all the data over the network
- Perhaps we can reduce before we shuffle
- This could greatly reduce the amount of data to be sent over the network
- Aggregating data before shuffling is known as **Map Side Reduction**

Map-Side Reduction

- Rather than passing all the data, combining data is preferred before shuffling
- **Map-Side Reduction** improves performance



ReduceByKey: Shuffle

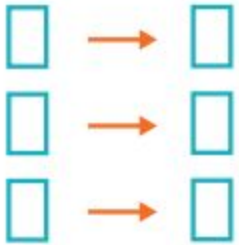


With ReduceByKey, data is combined so each partition outputs at most one value for each key to send over the network.

Analytics Vidhya

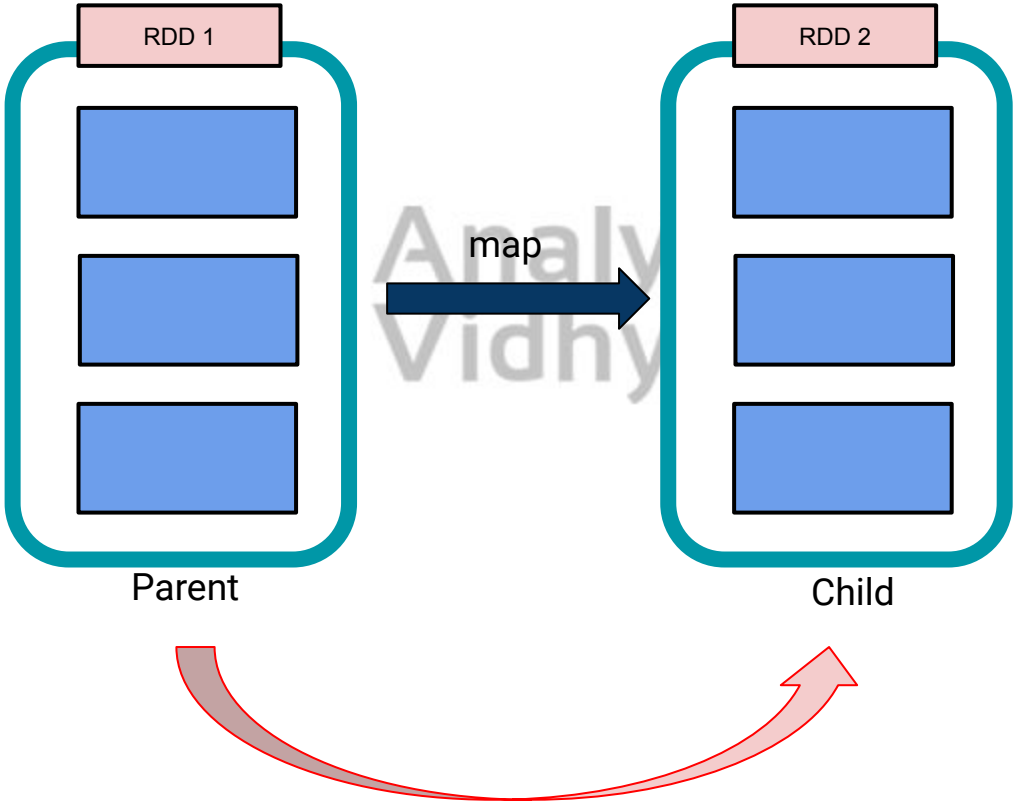
Types of Transformations

Narrow Transformations
1 to 1

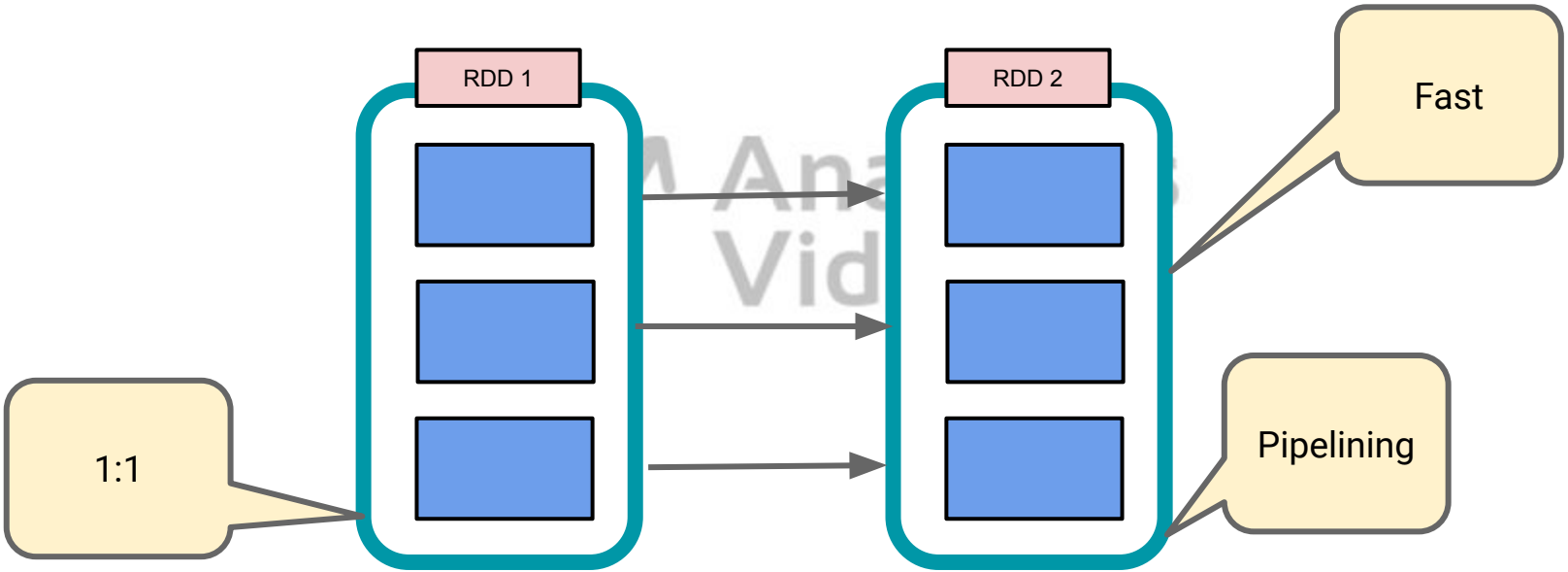


Wide Transformations (shuffles)
1 to N

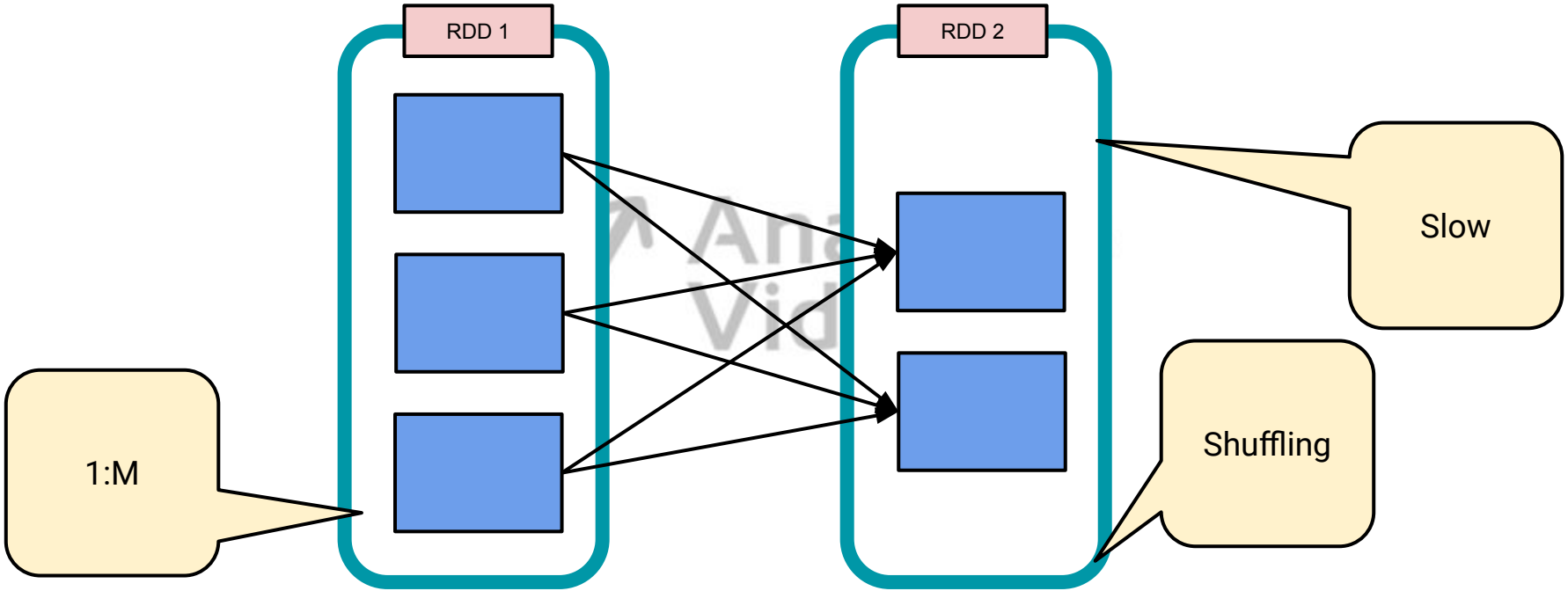
Transformation



Narrow Transformation

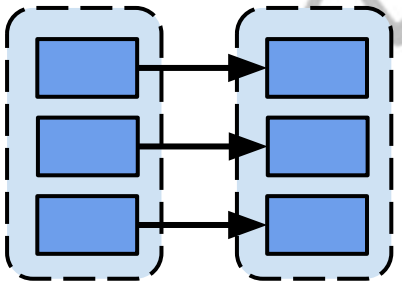


Wide Transformation



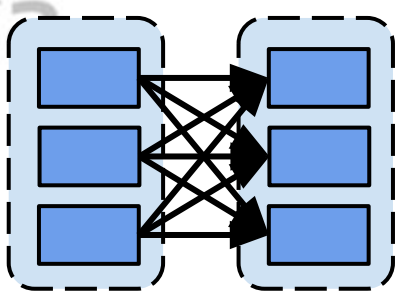
Narrow Transformations

- Map
- FlatMap
- Filter
- Union



Wide Transformations

- GroupByKey
- ReduceByKey
- Intersection
- Distinct



Narrow or Wide Transformation?

- sample
- intersection
- join
- sort
- cartesian
- repartition
- cogroup
- foldByKey
- combineByKey

Analytics Vidhya

Thank You!!

Analytics
Vidhya