VISUAL EXAMPLE HOW SHUFFLE WORKS IN Spark

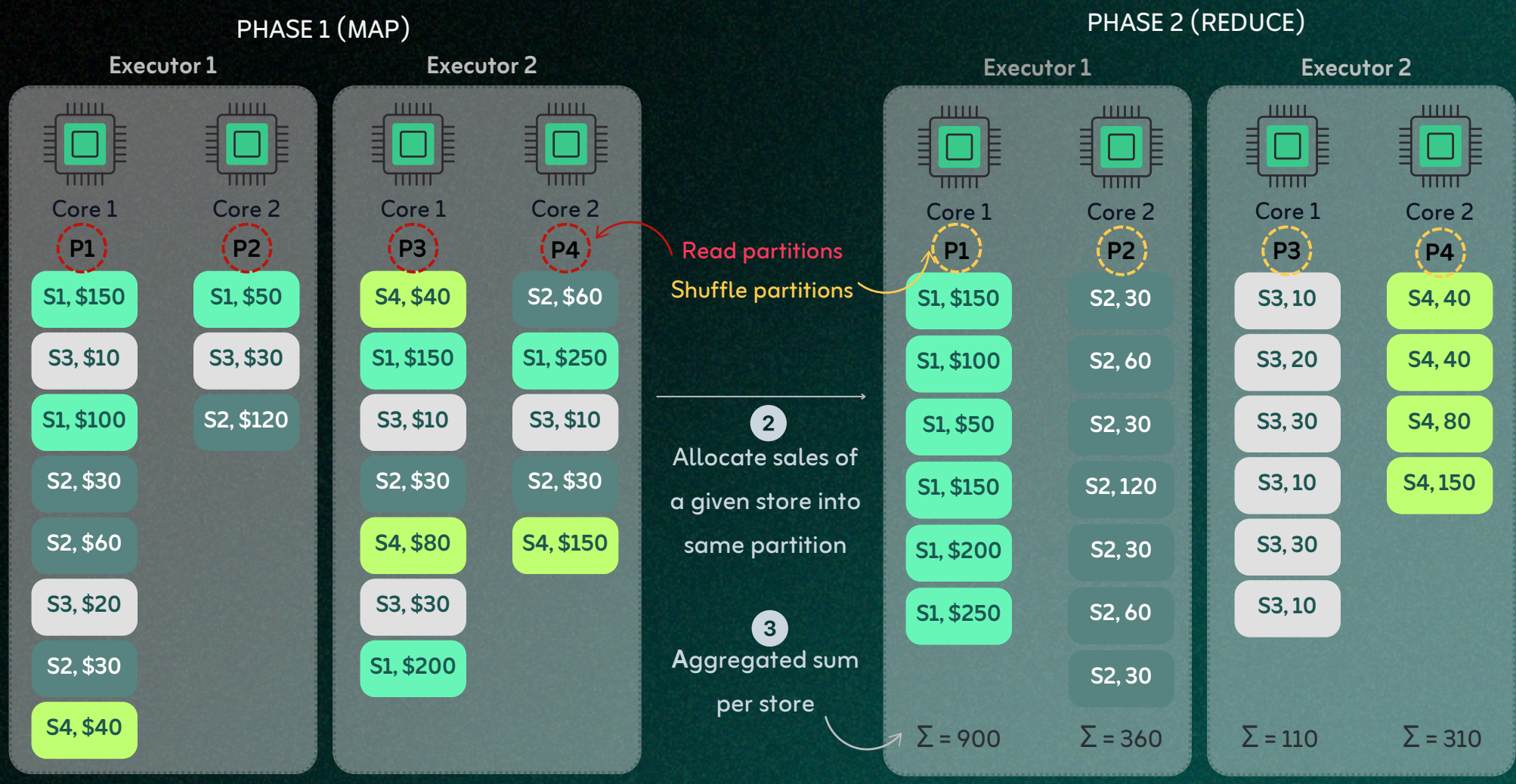
Shuffle: bring together data that is related but resides in different nodes. It happens when doing a wide transformation (JOIN, GROUP BY)

Q: What is the total sales per store? Perform an aggregated sum.

ie. Process data for 4 stores in n files, each containing Store_ID & Sales_volume data

How does it work?

1 Spark reads the data from files into partitions



Why shuffle partitions matter?

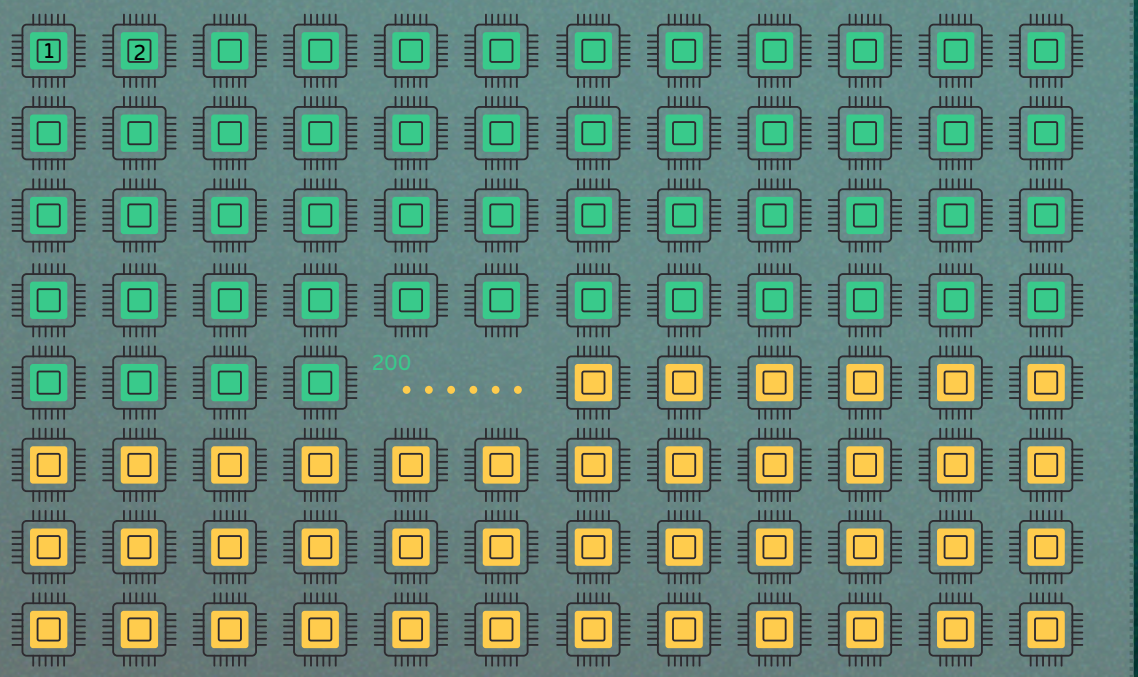
- Shuffle partitions run JOIN/ GROUP BY operations (Spark default: 200 partitions)
- Each partition ran by one core.
- Ideal processing per partition: ~1-200Mb

Consequences of not effective partition handling for your Spark job:

1. Slow completion time
2. Cluster underutilization

   ie. a 1000 core cluster running with default setup: only **200** partitions work while the other **800** are idle.

   Larger processing per working core makes the whole cluster slower and underutilized



Albert Campillo

Repost

VISUAL EXAMPLE HOW SHUFFLE WORKS IN Spark

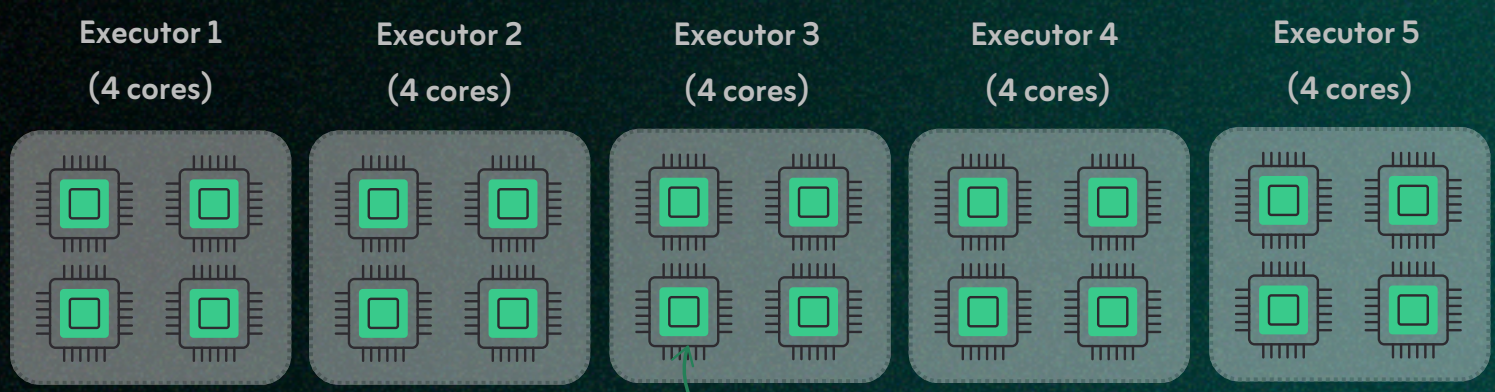
SCENARIO 1: The data partition per shuffle is very large (300 GB)

- 5 executors, **20 cores** (4 cores per executor); default shuffle partitions (spark.sql.shuffle.partition = **200**)
- Amount of data shuffled: **300Gb** (shuffle.write = 300)

Case 1

No change in DSP (= 200)

- Size per suffle partition: (300Gb / 200 DSP) = 1.5Gb/core
- 1 processing round (1.5Gb x 20 cores). Unoptimal size per core!

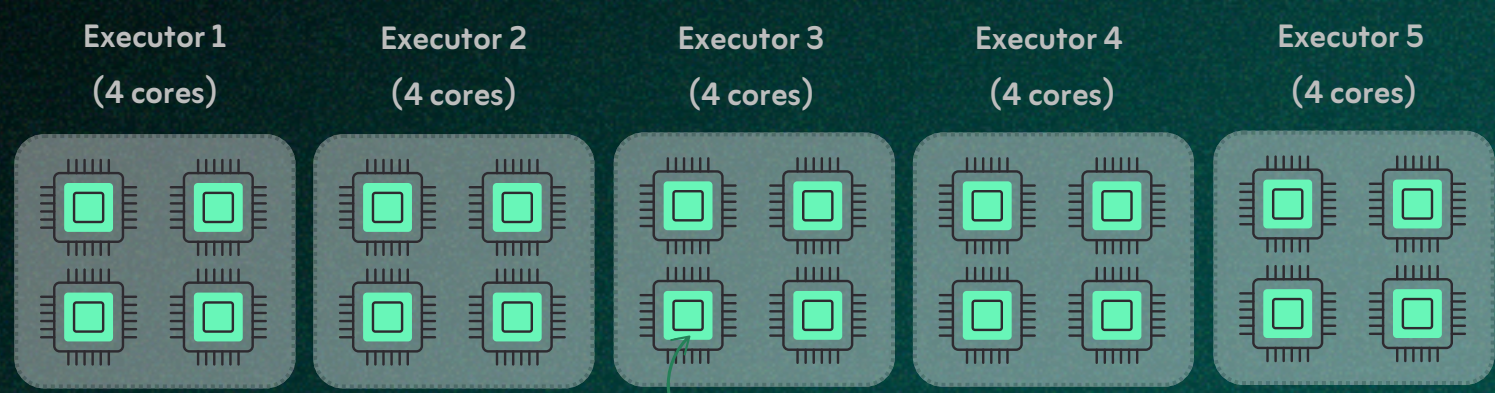


1.5Gb processed per core in 1 processing round
Unoptimal size per core!

Case 2

Change SP based on ideal processing size (ie. 200 Mb)

- 200mb/SP : (300Gb / X DSP) = 1500 SP
- 75 processing rounds (1500 SP/ 20 cores)



200Mb processed per core in 75 processing rounds
Ensures utilization of each core per round is adequate

SCENARIO 2: The data partition per shuffle is very small (50 MB)

- 3 executors, **12 cores** (4 cores per executor); default shuffle partitions (spark.sql.shuffle.partition = **200**)
- Amount of data shuffled: **50Mb** (shuffle.write = 50)

Case 1

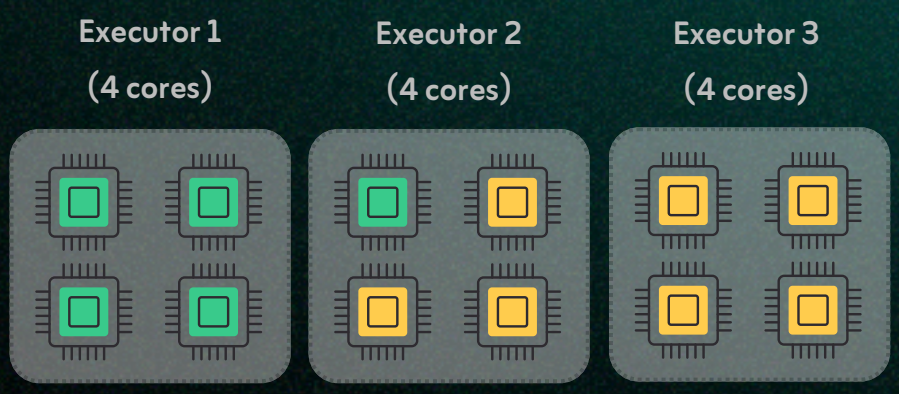
No change in DSP (= 200) --> Size per shuffle partition: (50Mb / 200 DSP) = **250Kb/core** (very small vs. optimal ~200Mb)

Case 2

Adjust the SP to higher value (ie. 10Mb/ core)

- Number of SP = 50Mb / 10Mb = 5 SP

Underutilized, as 7 cores are idle



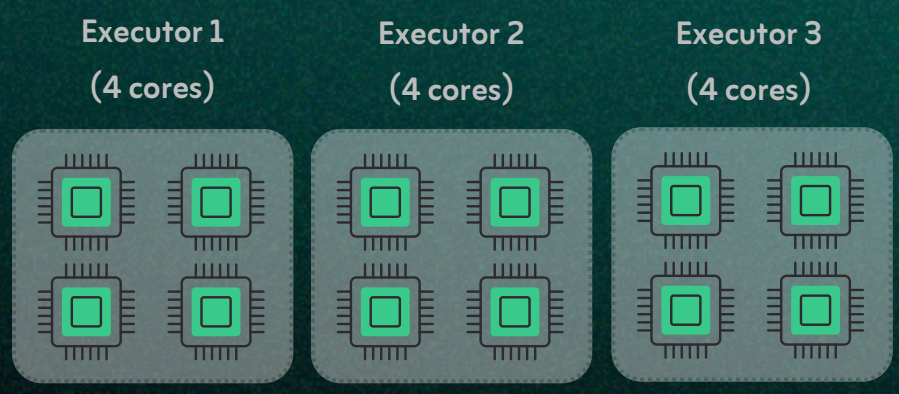
Working | Idle

Case 3

Adjust the number of SP (all 12 cores)

- Size of SP = 50Mb/ 12 cores =~ 4.16Mb/core

Ensures job completion is faster (more codes working)



Albert Campillo

Repost