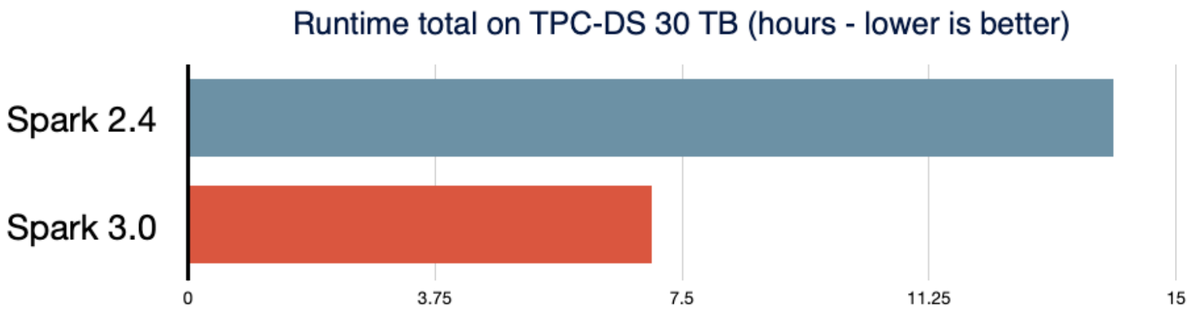
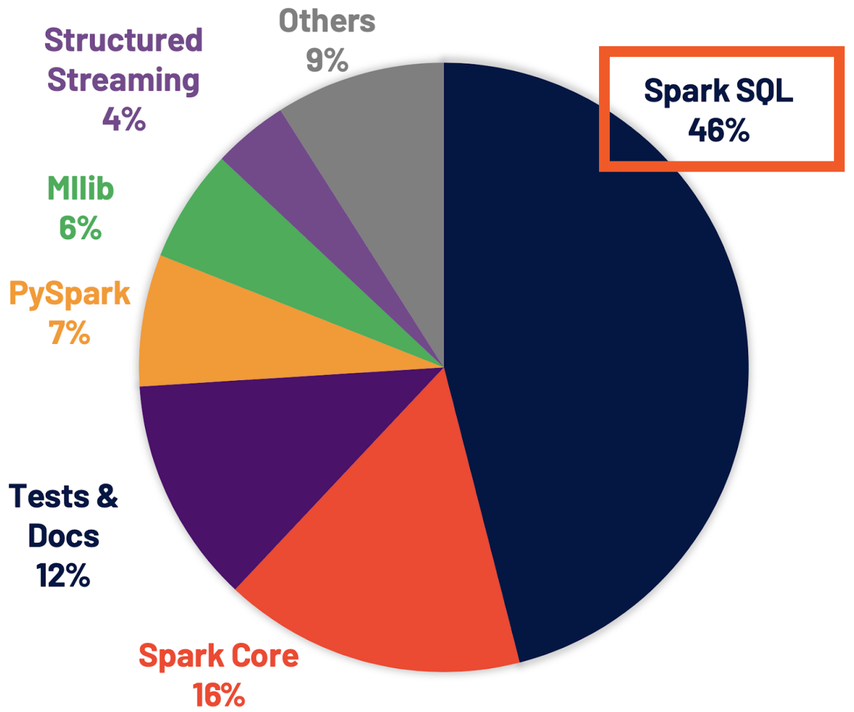
Features in Apache Spark 3

Mohit Batra
Founder, Crystal Talks

linkedin.com/in/mohitbatra

Apache Spark 3

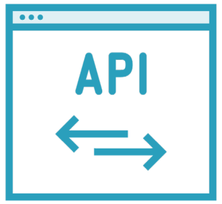




*Images Source: databricks.com/blog

Apache Spark 3 Features

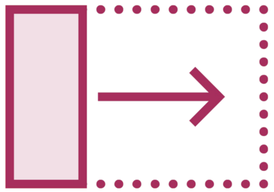
Performance



API Enhancements

SQL Compliance

Data Sources



Extensibility

Monitoring

1

Performance Improvements

Adaptive Query Execution (AQE) framework

- Reoptimizes query plan at runtime based on stats
- a) Dynamically coalescing shuffle partitions
- b) Dynamically switching join strategy
- c) Handling data skew in joins

Dynamic Partition Pruning (DPP)

- Improves on Partition Pruning technique

SQL join hints

- Spark has multiple join strategies
- Join Hint allows to enforce a particular join strategy
- Join Hints available for each join strategy

Faster query compilation

2

APIs, SQL and Monitoring

30+ new built-in functions

PySpark enhancements

- Pandas API enhancement & support
- Better error handling

Deep Learning improvements

- Project Hydrogen aims to unify data processing & deep learning

ANSI SQL compliance

- spark.sql.ansi.enabled *(disabled by default)*

Monitoring

- Better UI for Spark Structured Streaming
- Observable Metrics

3

Data Sources, Extensibility & Ecosystem

Built-in Data Sources

- New data sources like Apache Iceberg
- Performance improvements to existing sources like Parquet, Kafka, Delta Lake etc.

Extensibility

- Catalog API to use external catalog for managing tables (instead of Hive)

Ecosystem

- Support for Hadoop 3+, Hive 3+, JDK 11+

Overview

Adaptive Query Execution

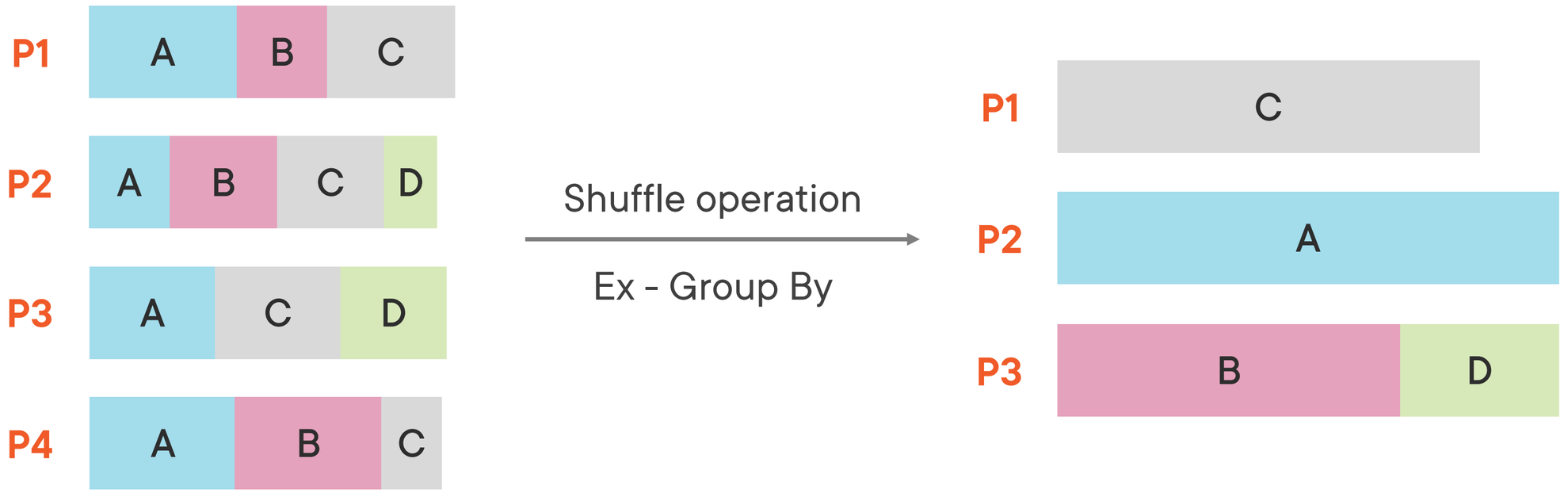
- Dynamically coalescing shuffle partitions
- Dynamically switching join strategy
- Handling data skew in joins

Dynamic Partition Pruning

Adaptive Query Execution: Dynamic Coalescing

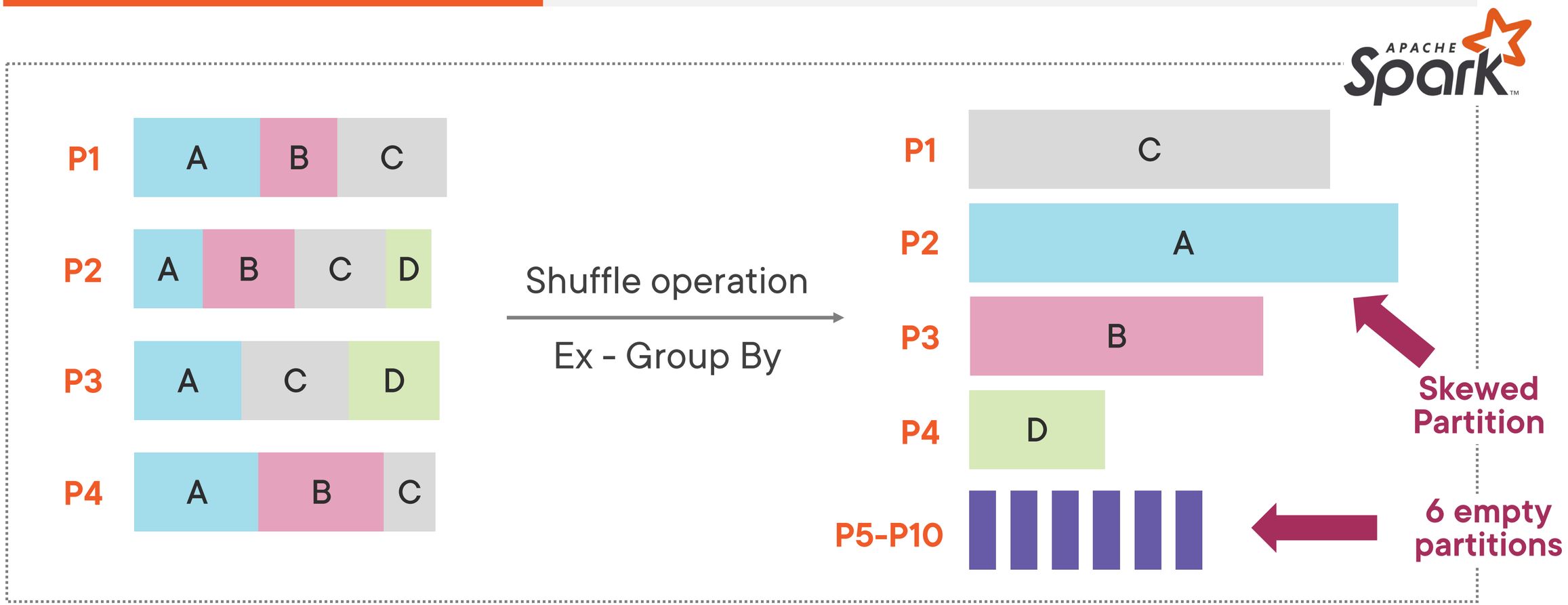
Shuffling Data

spark.sql.shuffle.partitions = 3 [default = 200]



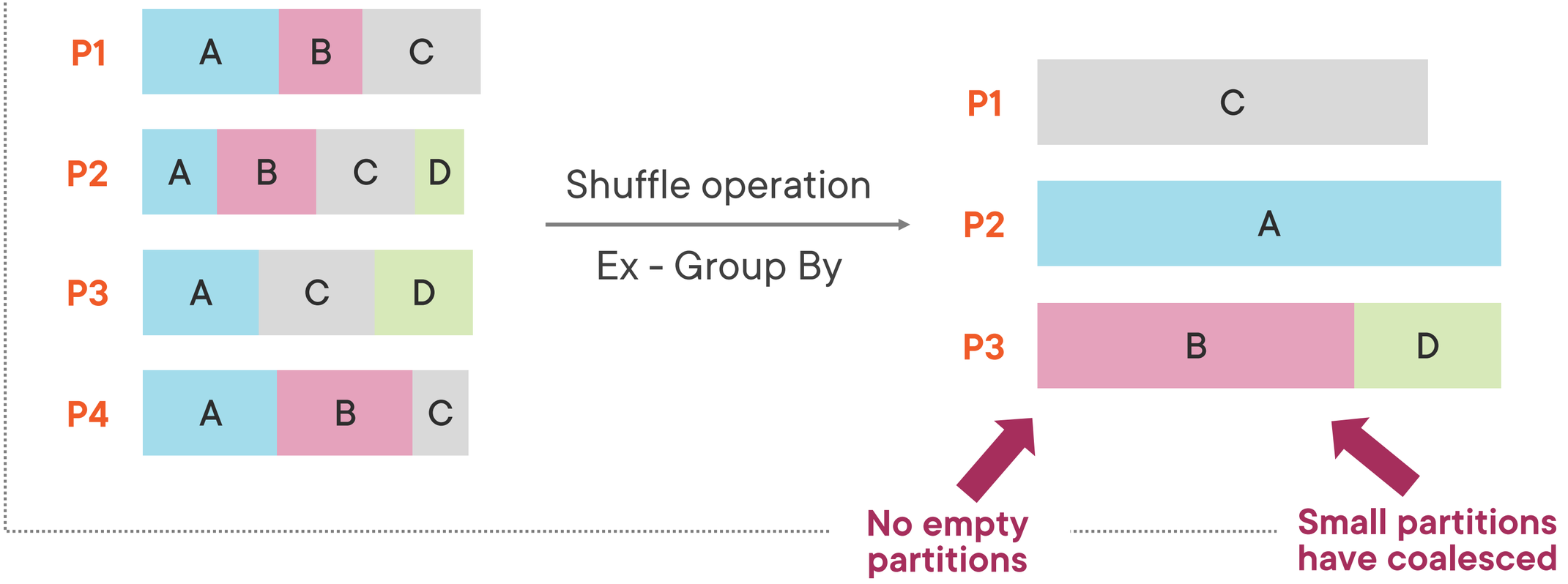
Shuffling Data

spark.sql.shuffle.partitions = 10 [default = 200]

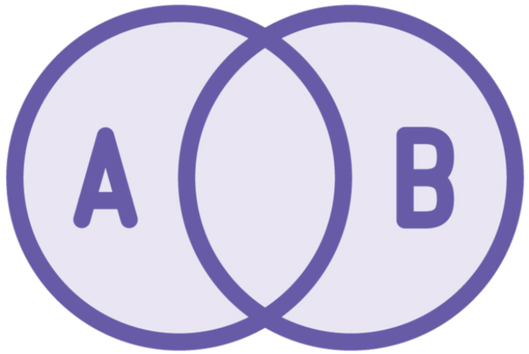


Shuffling With AQE

spark.sql.shuffle.partitions	= 10	[default = 200]
spark.sql.adaptive.enabled	= true	[default = true]
spark.sql.adaptive.coalescePartitions.enabled	= true	[default = true]



Dynamic Coalescing Shuffle Partitions



Having empty or lot of small partitions:

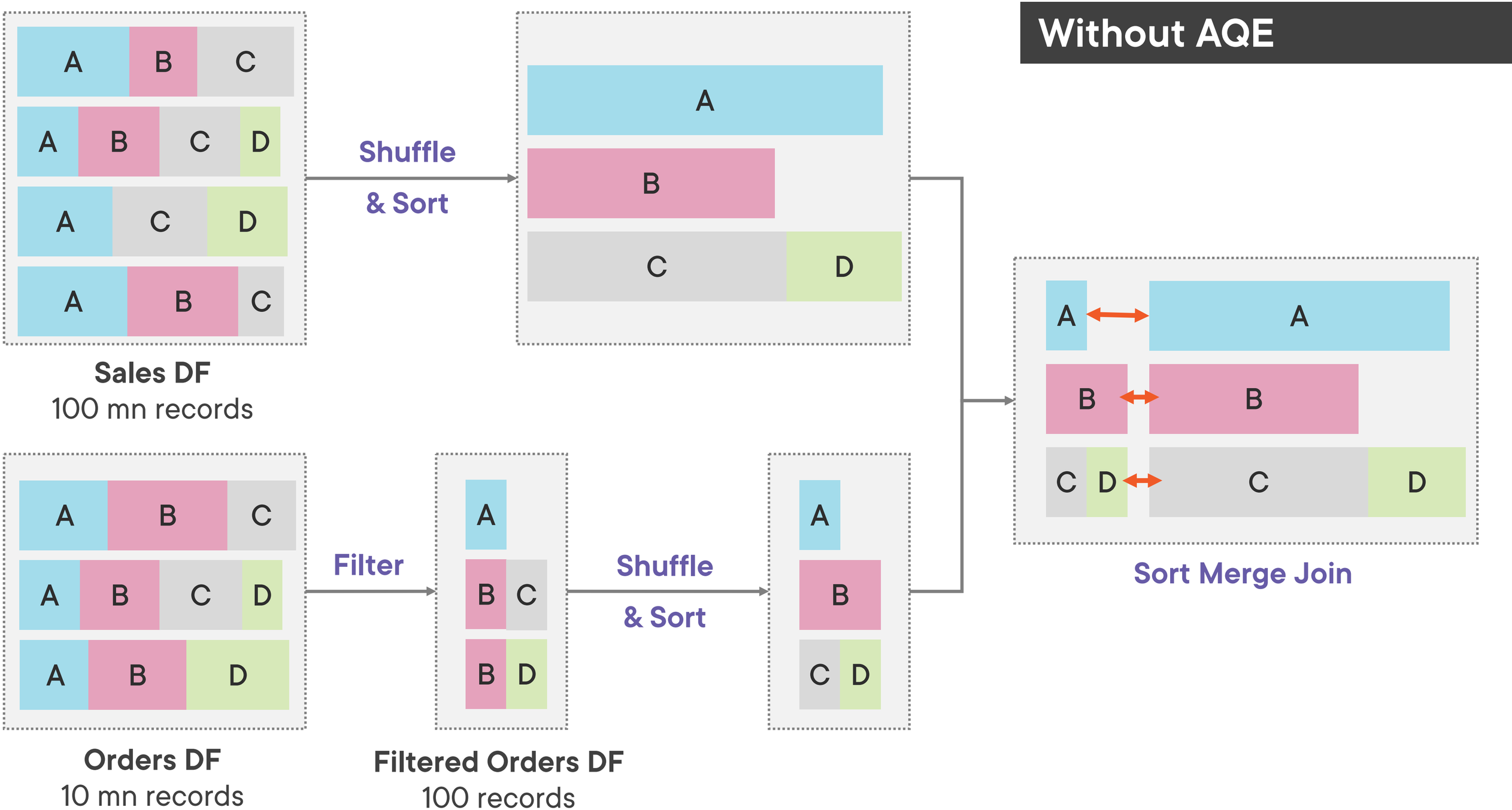
- Too many tasks are created
- Reduces parallelism and consumes time/resources

AQE dynamically coalesces shuffle partitions

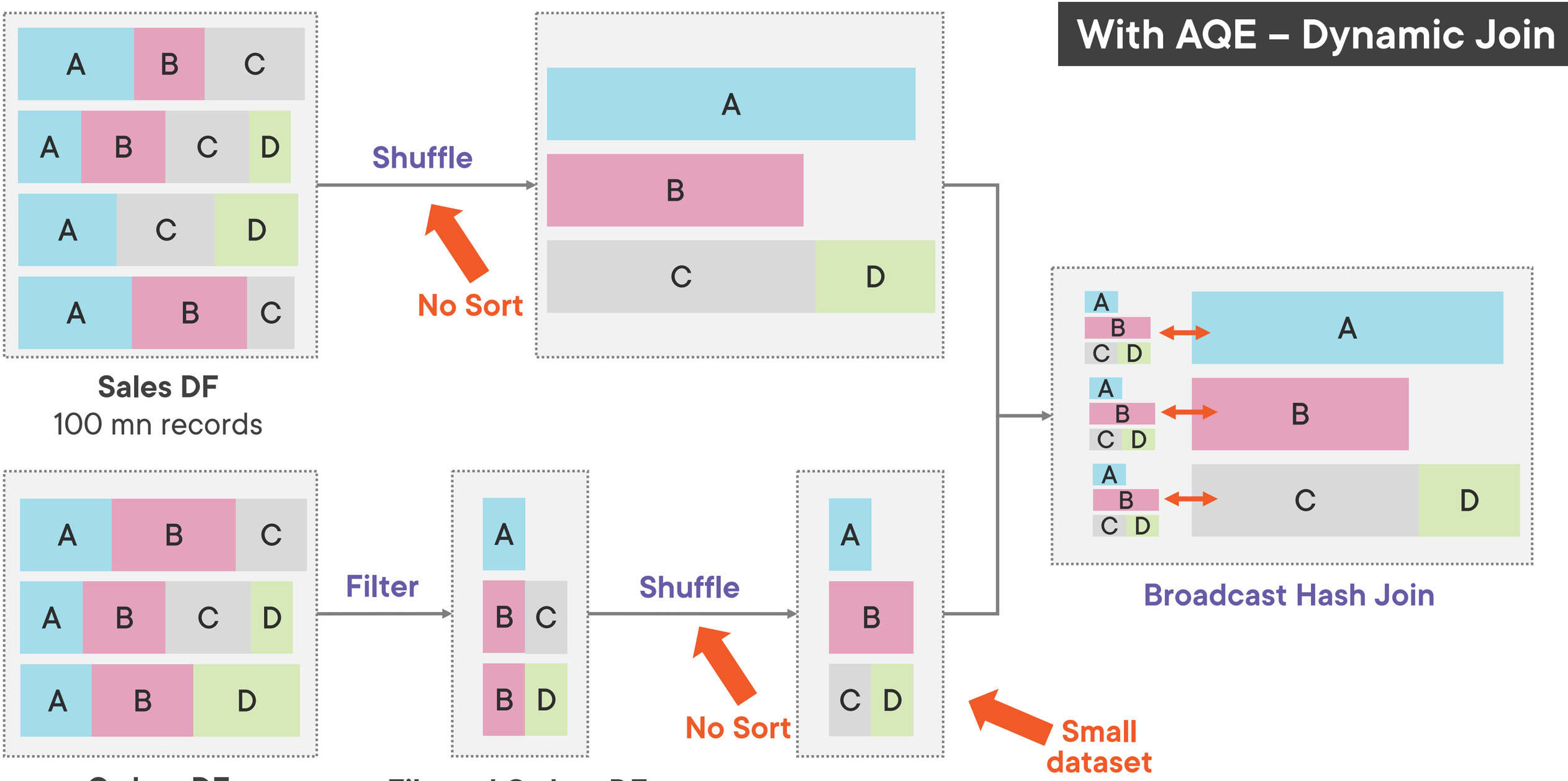
- Removes empty partitions
- Combines small partitions to produce optimal sized shuffle partitions

Adaptive Query Execution: Dynamic Join

Without AQE


A B C
A B C D
A C D
A B C
Shuffle
& Sort
A
B
C D
A A
B B
C D C D
Sales DF
100 mn records
A B C
A B C D
A B D
Filter
A
B C
B D
Shuffle
& Sort
A
B
C D
Sort Merge Join
Orders DF
10 mn records
Filtered Orders DF
100 records

With AQE – Dynamic Join



AQE – Dynamic Switching Join Strategy
OR
Manual Broadcast

Which is better?

Dynamically Switching Join Strategies

For large datasets, Shuffle Sort Merge Join is performed

For Broadcast Hash Join, one dataset must be small

- Small dataset should be less than setting spark.sql.autoBroadcastJoinThreshold
- Default threshold is 10 MB

If highly selective filter is applied on large dataset

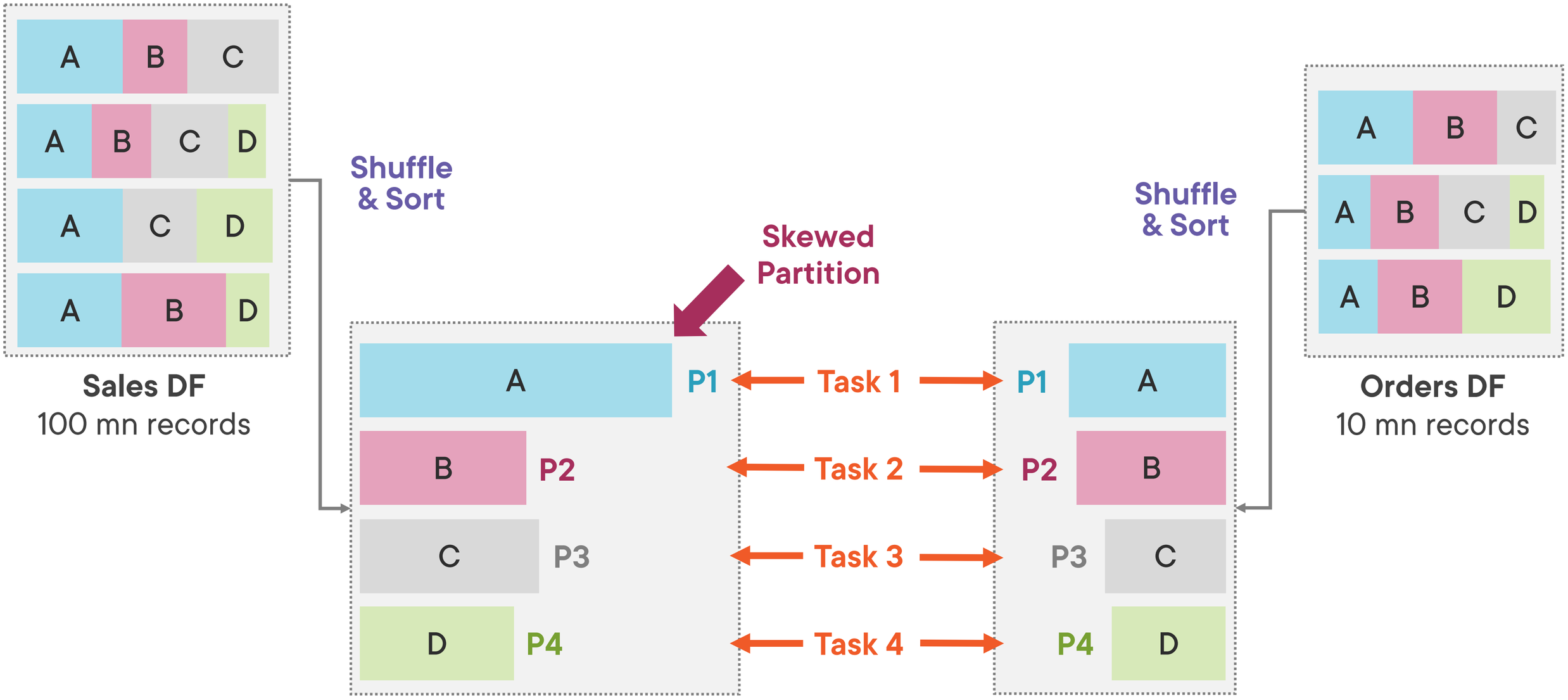
- It may become smaller than broadcast threshold
- Since execution plan is ready, still performs Shuffle Sort Merge Join (if joined with large dataset)

If enabled, AQE dynamically switches from Sort Merge Join to Broadcast Hash Join at runtime

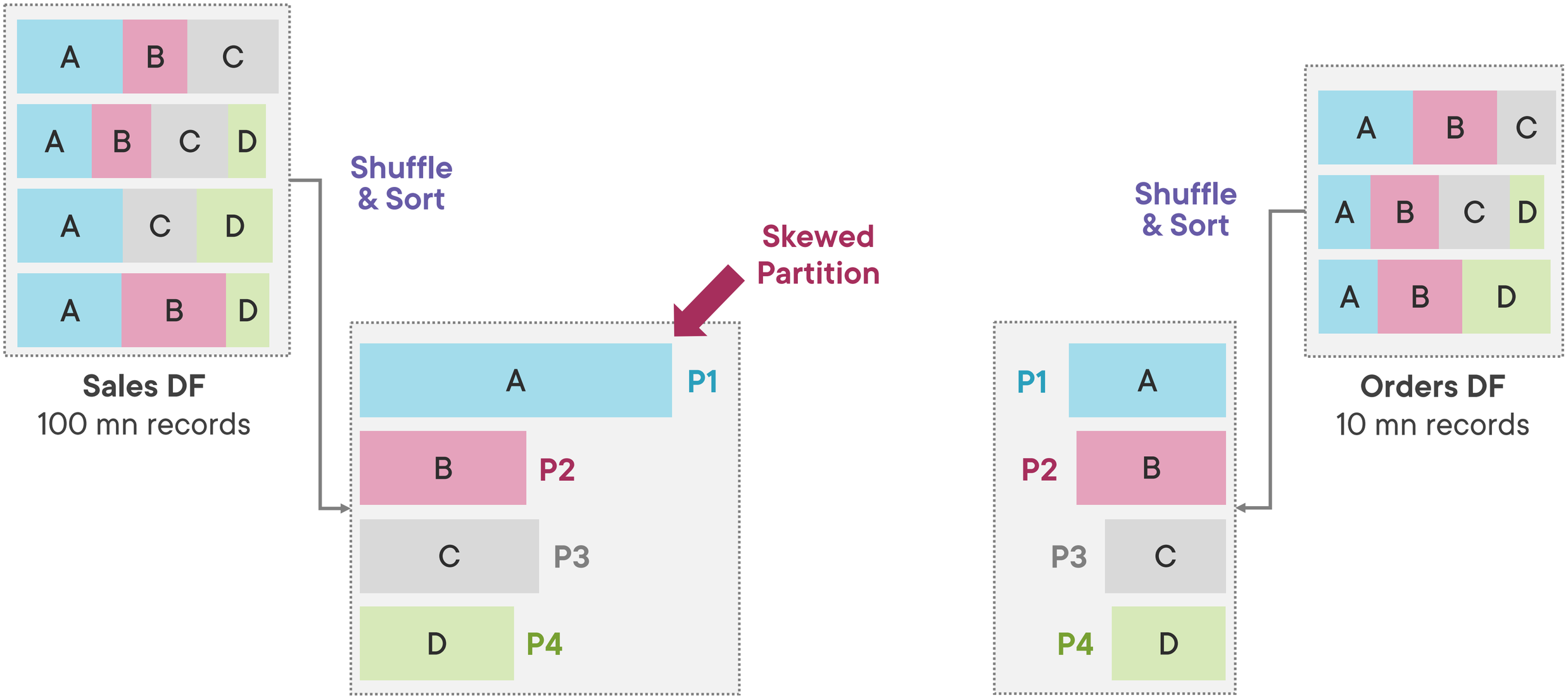
- AQE checks for dataset size after shuffle
- If size of a dataset is now less than broadcast threshold, switches to Broadcast Hash Join

Adaptive Query Execution: Handling Skew

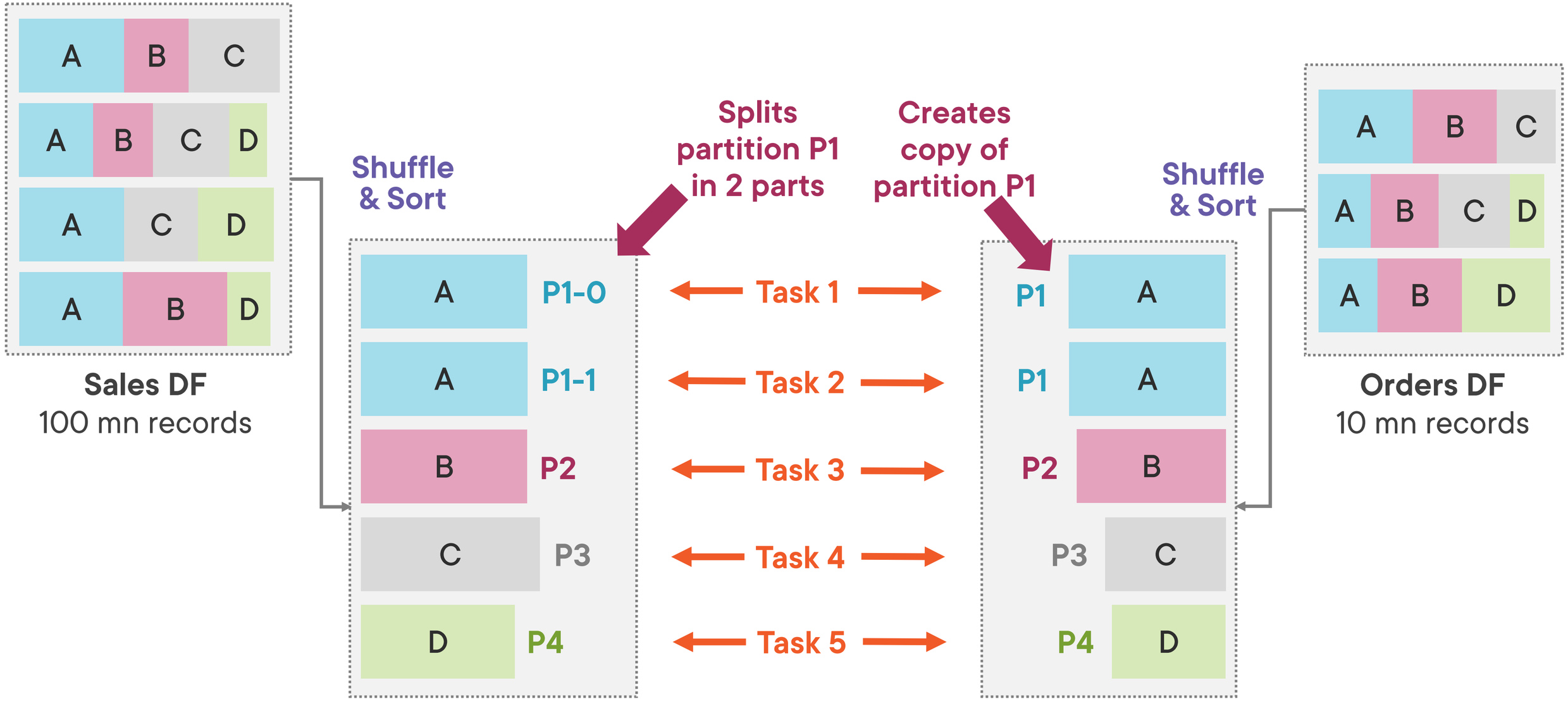
Without AQE


A B C
A B C D
A C D
A B D
Sales DF
100 mn records
Shuffle & Sort
Skewed Partition
A P1
B P2
C P3
D P4
Task 1
Task 2
Task 3
Task 4
P1 A
P2 B
P3 C
P4 D
Shuffle & Sort
A B C
A B C D
A B D
Orders DF
10 mn records

With AQE: Handling Skew



With AQE: Handling Skew



Dynamically Optimizing Skew Joins

Data Skew

- One partition has much more data than others

In join operations:

- After shuffle, data may be unevenly distributed among partitions
- Data skew can impact query performance
- Tasks processing larger partitions will take more time than ones handling smaller partitions

If enabled, AQE dynamically optimizes data skew in joins

- AQE checks for partition sizes after shuffle
- Splits skewed partitions into smaller sub-partitions
- Creates copy of corresponding partition on other side
- Number of tasks increase, but each one will almost take same time to finish

Dynamic Partition Pruning

Disk Partitions

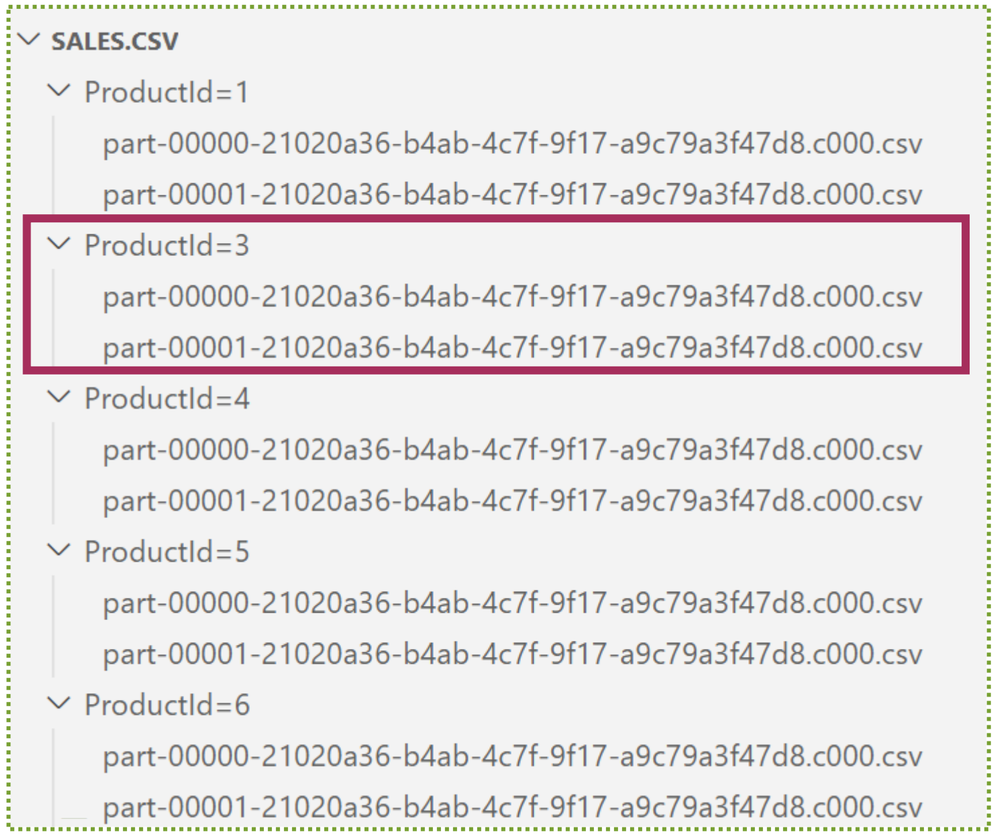
```
(
   salesDF
      .write
      .partitionBy("ProductId")
      .saveAsTable("Sales")
)
```

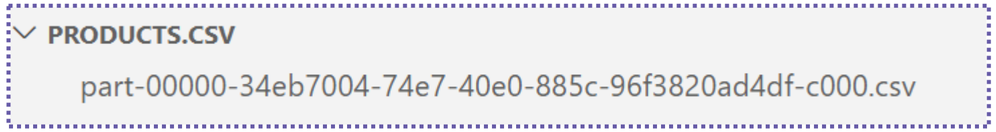
Partition Pruning

```
SELECT *
FROM Sales
WHERE ProductId = 3
```

Sales

100 mn records - Partitioned





Products

1000 records – Non-partitioned

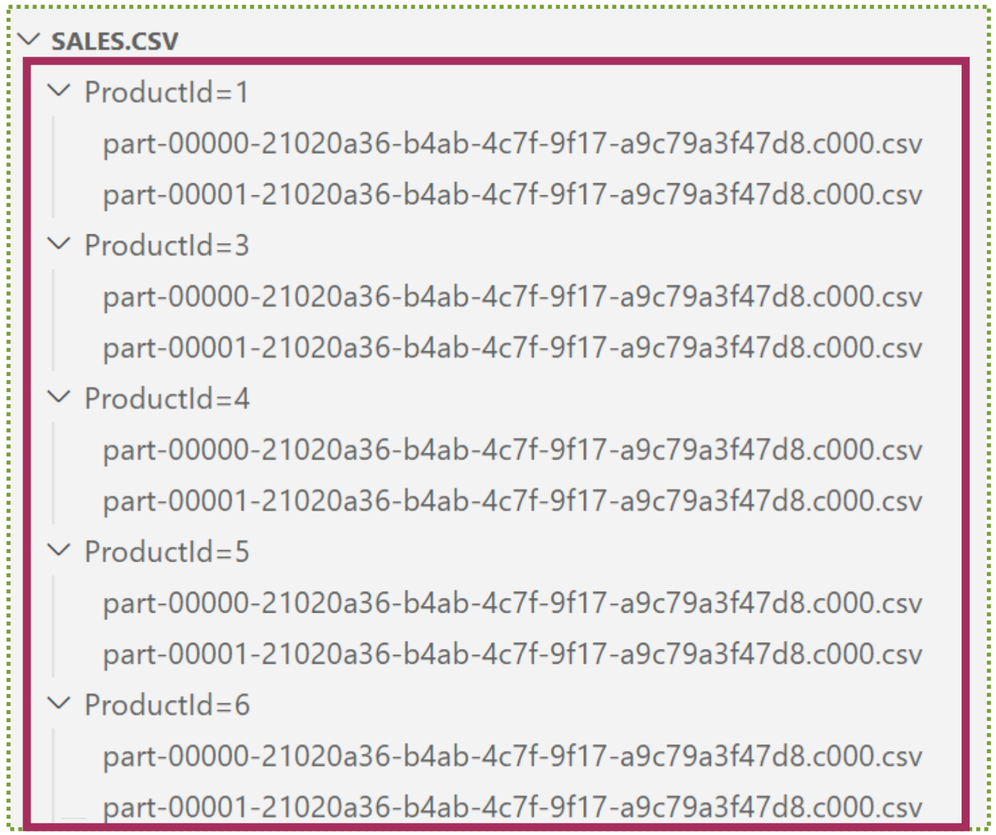
Partition Pruning Works!

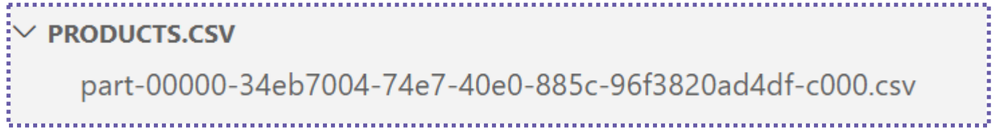
```
SELECT /*+  BROADCASTJOIN(p) */ *

FROM Sales s
   JOIN Products p
   ON s.ProductId = p.ProductId

WHERE s.ProductId = 3
```

Small Table

Sales

100 mn records - Partitioned





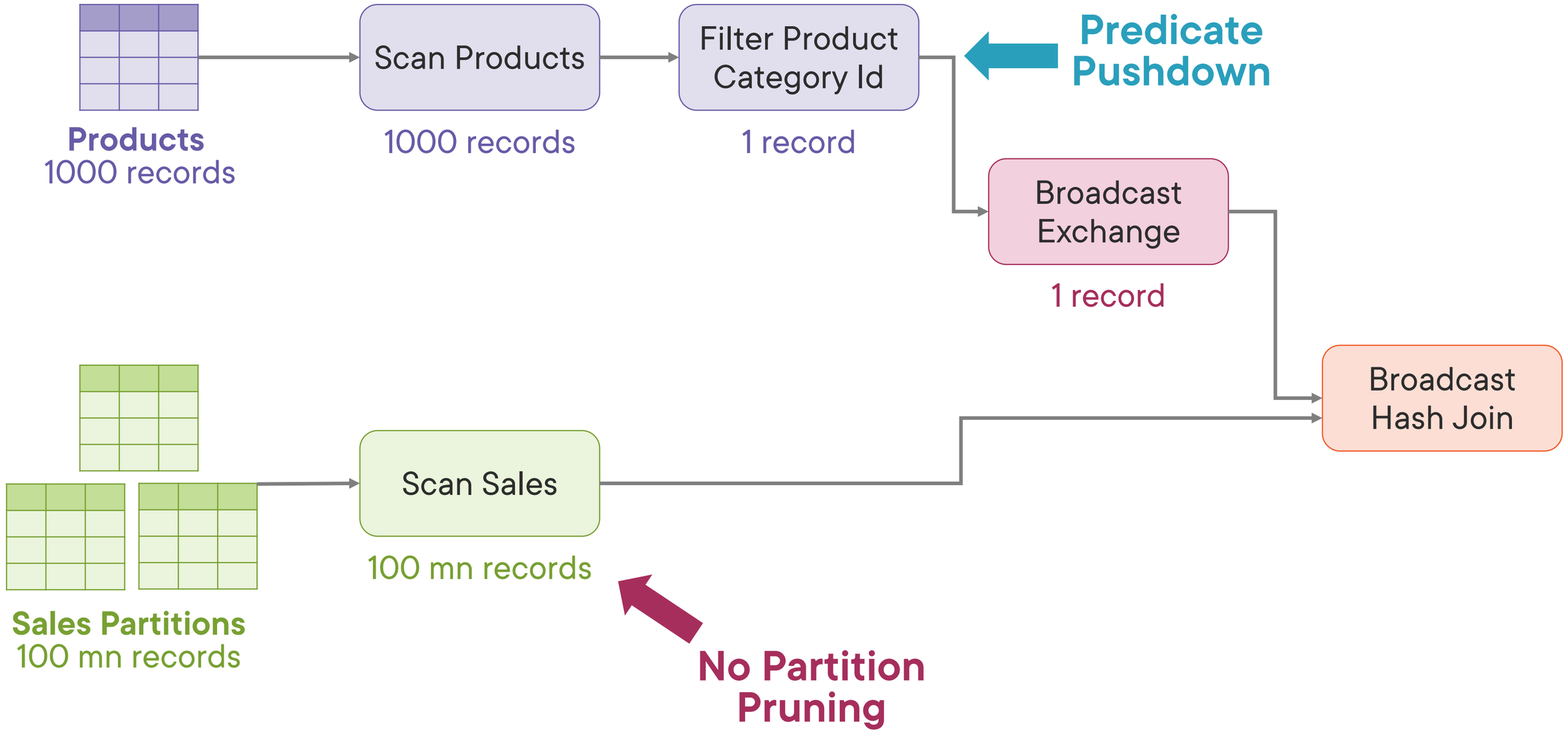
Products

1000 records – Non-partitioned

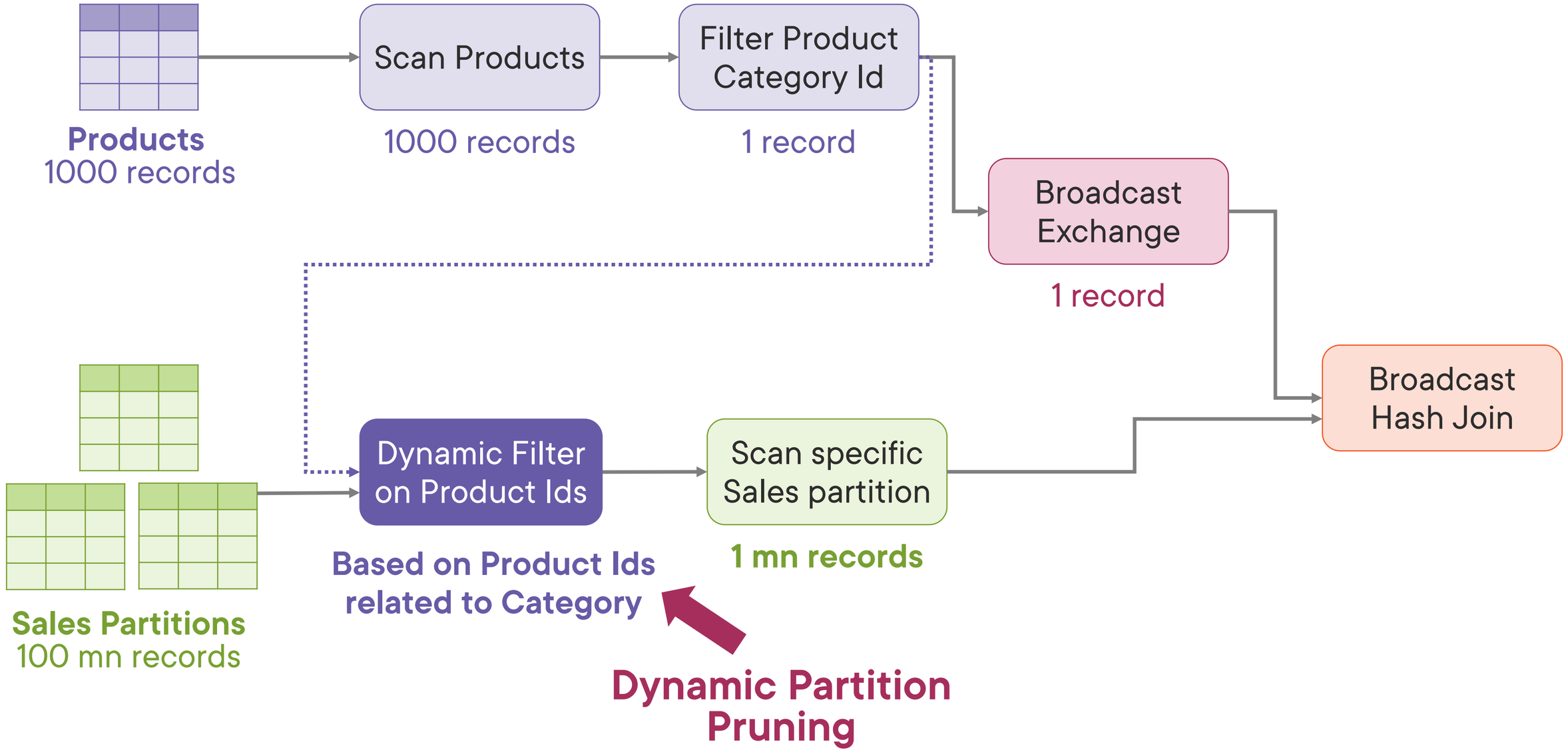
Partition Pruning Does Not Work!

```
SELECT /*+  BROADCASTJOIN(p) */ *

FROM Sales s
   JOIN Products p
   ON s.ProductId = p.ProductId

WHERE p.ProductCategoryId = 3
```

Without Dynamic Partition Pruning



With Dynamic Partition Pruning



Conditions

Dynamic Partition Pruning must be enabled

- spark.sql.optimizer.dynamicPartitionPruning.enabled

Large table must have disk partitions

During join, small table should be broadcasted

Summary

Spark 3 has several optimizations

Adaptive Query Execution

- Reoptimizes query plan at runtime based on stats

1. AQE: Dynamically coalescing shuffle partitions

- After shuffle, empty shuffle partitions are removed & small ones are merged

2. AQE: Dynamically switching join strategy

- At runtime, if one dataset becomes small, AQE can switch from Shuffle Sort to Broadcast Hash join

3. AQE: Handling data skew in joins

- After shuffle, if any partition is skewed, AQE can split that partition to multiple smaller partitions

Dynamic Partition Pruning

- During join, Spark can dynamically skip disk partitions at runtime that are not required by the query

Up Next:
Building Reliable Data Lake with Spark and Delta Lake