Seekho Bigdata
Data is the New Oil

Adaptive Query Execution (AQE)

Karthik Kondpak
9989454737

Day 9 — Spark Optimization Topic

Adaptive Query Execution (AQE)

How Spark Fixes Your Plan *While Running*

Adaptive Query Execution (AQE) is one of Spark's most powerful optimization features.

Unlike the Catalyst optimizer (which plans before execution), **AQE optimizes your query *during execution*** based on real data statistics.

Think of AQE as:

> *"Spark adjusts the engine while the car is running."*

What Is AQE?

AQEdynamicallychanges the execution plan **after seeing actual runtime statistics** such as:

- Shuffle partition sizes
- Skew distribution
- Output row counts

- File sizes

This makes Spark **more accurate, faster, and more resource-efficient**.

To enable AQE (usually ON by default):

```
spark.conf.set("spark.sql.adaptive.enabled", "true")
```

Why AQE Is Needed

Traditional Spark plans had limitations:

- They relied on inaccurate **estimated statistics**
- Couldn’t predict **skew**
- Couldn’t merge **tiny files**
- Couldn't change join strategies mid-way

AQE solves all of this.

AQE Has 3 Core Features

Dynamic Coalescing ofShufflePartitions

Dynamic Switching of Join Strategies

Handling Skewed Joins Automatically

Let's break these down.

Dynamic Shrinking of Shuffle Partitions

Before Spark 3.x, shuffle partitions had to be set manually:
`spark.sql.shuffle.partitions = 200`

If 200 partitions were too many → tiny tasks, overhead.

If too few → large tasks, out-of-memory.

With AQE:

Spark automatically **reduces** partitions based on data size.

Example:

- You set 200 shuffle partitions
- Actual data is small
- AQE may shrink it to 10 partitions automatically

Check in physical plan:
`AdaptiveSparkPlan isFinalPlan=true`

You'll also see:

`Coalesced Shuffle Partitions`

Dynamic Switching of Join Strategies

AQE can change join strategies **during runtime**.

Examples:

✓Case1:Smalltabledetected

If Spark finds a table is smaller than expected (<10 MB), it switches to:

➡ **Broadcast Hash Join**

This removes shuffle completely.

✓Case2:Tablestoolargefor broadcast

Spark switches back to:

➡ **Sort-Merge Join**

✔**Case 3: Hash join becomes cheaper**

AQE automatically uses:

➡ **Shuffle Hash Join**

You don't need to specify the join type.

Automatic Skew Join Handling

Skew happens when one key has far more records than others.

Example:

customer_id = 999 has **20 million** rows

Other customer_ids have **5k–10k** rows

Without AQE:

- One partition becomes huge
- Entire job slows down
- Executors may fail due to memory pressure

With AQE:

AQE automatically detects skew (>3× average partition size)

And then:

✔ Splits the skewed partition into smaller chunks

✔ Adds parallel tasks

✔ Balances execution speed

✔ Avoids OOM errors

You'll see in plan:

`SplitSkewedPartitions`

Scenario Example — Swiggy

You are joining:

- 300M-row order table
- 5M-row customer table

Customer table has skew on:

```
city = "Bengaluru"
```

AQE helps you by:

- Detecting skew
- Rewriting the join plan
- Splitting Bengaluru bucket
- Broadcasting smaller maps
- Reducing shuffle size

End result:

✓Faster join

✓Balanced partitions

✓Reduced shuffle cost

How to See AQE in Action (Spark UI)

Go to:

SQL Tab → View “AdaptiveSparkPlan” in the physical plan

Look for:

- `AdaptiveSparkPlan`
- `DynamicPruning`
- `SplitSkewedPartitions`
- `CoalescedShuffle`
- `BroadcastExchange`
- `FinalPlan=true`

Which confirm AQE is applied.

Minimal Code Example

```
spark.conf.set("spark.sql.adaptive.enabled", "true")

df = spark.read.parquet("/delta/orders")
```

```
result = df.groupBy("customer_id").count()
result.explain(True)
```

Let's build your Data Engineering journey together!

Call us directly at: 9989454737

https://seekhobigdata.com/