Seekho Bigdata
Data is the New Oil

Shuffle Partitions

Karthik Kondpak
9989454737

Day 16 — Spark Optimization Topic

🔥 Shuffle Partitions — How to Choose the Right Number (Not 200!)

Most people never tune **shuffle partitions**, and that's why their Spark jobs are slow, costly, and imbalanced.

By default:

```
spark.sql.shuffle.partitions = 200
```

This default is **WRONG for 90% of workloads**.

Choosing the right shuffle partition count is one of the **most important Spark performance optimizations**.

What Are Shuffle Partitions?

Whenever Spark performs operations like:

- groupBy

- join
- orderBy
- DISTINCT
- repartition
- window with order
- cube/rollup

Spark **shuffles data** and splits it across **shuffle partitions**.

Each partition becomes **one Spark task**.

Why Default (200) Is Often Wrong

You may get:

- ✓ *Too many small tasks* (overhead, slow scheduling)
- ✓ *Too few large tasks* (executors overloaded, memory errors)
- ✓ *Skewed tasks* (1 task runs 10 minutes, others finish in seconds)

Tuning shuffle partitions fixes all these problems.

General Rule for Choosing Partitions

Use:

Total Data Size / Ideal Partition Size

Spark's ideal partition size is:

`128 MB to 256 MB`

So:

```
numPartitions = total_data_size_MB / 200 MB
```

Example:

Input data = **2 TB**

```
2000 GB / 0.2 GB ≈ 10,000 partitions
```

But to avoid too many small tasks, you may round down:

4000–6000 partitions.

Example :

You work at **PhonePe**.

A daily transaction table has:

```
Data size: 300 GB
groupBy customer_id
```

Rule:

```
300 GB / 200 MB = 1500 partitions
```

So:

```
spark.conf.set("spark.sql.shuffle.partitions",
"1500")
```

This gives optimal load distribution across executors.

When Shuffle Happens

Shuffle is triggered by:

Operation	Shuffle?
groupBy	✓Yes
join	✓Yes
distinct	✓Yes
orderBy	✓Yes
repartition(N)	✓Yes
coalesce	No shuffle
filter/select	No shuffle

Good Partition Count Strategy

Small data (< 2 GB)

```
spark.sql.shuffle.partitions = 50
```

Medium data (2–50 GB)

```
200–800 partitions
```

Large data (50–500 GB)

```
800–3000 partitions
```

Very large data (> 500 GB)

```
3000–10000 partitions
```

Smart Tip — Use AQE to Auto-Tune (Best Practice)

Adaptive Query Execution dynamically adjusts shuffle partition count.

Enable:

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

AQE will:

- Merge tiny shuffle partitions
- Split skewed partitions
- Adjust partition count during runtime

No manual tuning needed.

How to Set Shuffle Partitions

Session level

```
spark.conf.set("spark.sql.shuffle.partitions", "800")
```

Spark-submit

```
--conf spark.sql.shuffle.partitions=800
```

Within a notebook

```
spark.conf.set("spark.default.parallelism", "800")
```

Example Before & After Tuning

Case	Partitions	Runtime
Default Spark (200)	200	32 mins
Tuned (1500)	1500	7 mins

Same data.

Same cluster.

Only partition count changed → **4.5× faster**.

Let's build your Data Engineering journey together!

Call us directly at: 9989454737

https://seekhobigdata.com/