Seekho Bigdata
Data is the New Oil

PySpark Scenario-Based Interview Questions (Complete Notes Series)

DAY 22 — Spark Memory Management & GC Tuning

Karthik Kondpak
9989454737

PySpark Scenario-Based Interview Questions (Complete Notes Series)

DAY 22 — Spark Memory Management & GC Tuning

Concepts Covered Today

- Spark memory architecture
- Execution vs Storage memory
- On-heap vs Off-heap memory
- Garbage Collection (GC) basics
- GC tuning strategies Real Indian
- production incidents

Spark Memory Architecture (MUST KNOW)

```
Executor JVM Memory
│
├── Execution Memory
```

```
│ ├─ Shuffles
│ ├─ Joins
│ ├─ Aggregations
│ └─ Sorts
│
├── Storage Memory
│ ├─ Cached DataFrames
│ ├─ Broadcast variables
│ └─ Persisted RDDs
│
└── Reserved Memory (System)
```

Execution & Storage memory **share space dynamically**.

Key Spark Memory Configurations

```
spark.executor.memory
spark.executor.memoryOverhead
spark.m emory. fract ion        (default  0.6)
spark.m emory. stora geFra ction(default 0.5)
```

Scenario

A Spark job processing **IRCTC ticket booking data**:

- Heavy joins
- Cached lookup tables
- Frequent executor crashes

Error seen:

```
ExecutorLostFailure (OOM)
```

Why Spark Jobs Fail with OOM?

- Large shuffles
- Uncontrolled caching
- Broadcasting big tables
- Skewed partitions
- Poor GC behavior

Execution vs Storage Memory (COMMON QUESTION)

Area	Used For	Priority
Execution	Joins, shuffles	High
Storage	Cache, persist	Lower

Execution memory can evict cached data.

Caching & Persistence Best Practices

```
df.persist(StorageLevel.MEMORY_AND_DISK)
```

Cache only reused data

Do not cache large intermediate results

Garbage Collection (GC) in Spark

- **What is GC?**

GC cleans unused JVM objects to free memory.

Problem:

- Excessive GC pauses slow jobs dramatically

Signs of GC Issues

- Tasks slow despite low CPU usage
- Executors alive but not progressing
- Long GC time in Spark UI

GC Tuning

🔹 Use G1GC (Recommended)

```
spark.executor.extraJavaOptions = -XX:+UseG1GC
```

🔹 Tune Heap Occupancy

```
-XX:InitiatingHeapOccupancyPercent=35
```

On-Heap vs Off-Heap Memory

Type	Pros	Cons
On-Heap	Managed by GC	GC overhead
Off-Heap	Less GC	Manual tuning

```
spark.memory.offHeap.enabled=true
spark.memory.offHeap.size=4g
```

Seekho Bigdata

Data is the New Oil

Let's build your Data Engineering journey together!

📩 Call us directly at: 9989454737

https://seekhobigdata.com/