Seekho Bigdata
Data is the New Oil

PySpark Scenario-Based Interview Questions (Complete Notes Series)

DAY 24 — Fault Tolerance, Lineage & Checkpointing

Karthik Kondpak
9989454737

PySpark Scenario-Based Interview Questions (Complete Notes Series)

DAY 24 — Fault Tolerance, Lineage & Checkpointing (How Spark Recovers) 🔥

Concepts Covered Today

- What fault tolerance means in Spark
- RDD lineage (DAG lineage graph)
- How Spark recovers lost data
- Checkpointing vs caching
- Driver vs executor failures
- Streaming checkpointing (brief)

What is Fault Tolerance?

Fault tolerance is Spark’s ability to **recover lost data and continue execution** when:

- Executor crashes
- Nodes go down
- Tasks fail

Spark achieves fault tolerance using **lineage**, not data replication.

RDD Lineage

🔹 What is Lineage?

Lineage is a **logical graph of transformations** used torebuild lost partitions.

```
Source → map → filter → join → aggregation
```

Spark stores *how to recompute data*, not the data itself.

How Spark Recovers from Executor Failure

Executor crashes

Tasks running on it fail

Spark checks lineage

Lost partitions are recomputed

Tasks rerun on another executor

🇮🇳 Indian Real-Time Scenario

You process **Aadhaar verification logs** using Spark.

- One executor node crashes mid-job
- Job continues without restart

➡ Spark recomputes missing partitions using lineage.

Interview Question: Why Doesn’t Spark Replicate Data Like HDFS?

✅ Correct Answer

Replication is expensive. Spark uses **lineage-based recomputation** for efficiency.

Caching vs Checkpointing

Aspect	Cache / Persist	Checkpoint
Storage	Memory / Disk	HDFS / Cloud
Lineage	Kept	Truncated
Recovery speed	Fast	Slower
Use case	Reuse data	Long lineage

When Lineage Becomes a Problem

- Very long DAGs
- Iterative algorithms
- Streaming pipelines

➡ Leads to recomputation overhead.

Checkpointing

◆ Enable Checkpoint Directory

```
spark.sparkContext.setCheckpointDir("/checkpoints")
```

◆ Apply Checkpoint

```
df.checkpoint()
```

Checkpoint cuts off lineage.

Structured Streaming Checkpointing

```
.writeStream \
.option("checkpointLocation", "/chk/upi")
```

✅ Why Needed

- Exactly-once processing
- Offset tracking
- Failure recovery

Driver Failure vs Executor Failure

Failure	Recovery
Executor	Automatic
Driver	Job restarts required

Driver is a single point of failure.

SB

Seekho Bigdata

Data is the New Oil

Let's build your Data Engineering journey together!

📩 Call us directly at: 9989454737

🌐 https://seekhobigdata.com/