PySpark

Part – 02 Spark Architecture

Save this POST for your Next ***Interviews***

@code.with.maddy_

Spark Architecture looks scary? 😵‍💫

Driver… Executor…

Job… Stage… Task…

Why so many components?!

👉 Let’s break Spark Architecture

👉 Step by step

👉 With interview clarity

What Is Spark Architecture? 🧠

Spark Architecture = How Spark executes your code internally.

It answers:

✔ Who controls the job?

✔ Where does code run?

✔ How data is processed in parallel?

✔ How results are returned?

💡 *If you understand this - **PySpark** becomes EASY.*

Main Components of Spark Architecture 🧩

*Spark has **5 core** components:*

- Driver Program
- Cluster Manager
- Worker Nodes
- Executors
- Tasks

👉 ***Interviews** mostly revolve around these.*

Driver Program (The Brain 🧠)

Driver *= Master controller of Spark job*

Responsibilities:

✓ Reads your PySpark code

✓ Creates SparkSession

✓ Converts code into jobs

✓ Splits jobs into stages

✓ Schedules tasks

✓ Collects final results

📌 ***Important Interview Line:***

Driver decides what to do, not how to execute.

Cluster Manager (Resource Manager 🏗️)

Cluster Manager = Allocates resources to Spark

Examples:

- *Standalone*
- *YARN*
- *Kubernetes*
- *Mesos*

Responsibilities:

✓ *Allocates CPU & memory*

✓ *Starts executors*

✓ *Manages cluster resources*

📌 ***Interview Tip:***

Spark itself does NOT manage hardware resources — ***cluster manager does.***

Worker Nodes (The Machines 💻)

***Worker Node** = A machine in the cluster*

Responsibilities:

✔ *Hosts executors*

✔ *Executes tasks*

✔ *Stores data in memory/disk*

📌 ***Interview Tip:***

👉 A cluster has multiple worker nodes.

Worker Node ≠ Executor

(This confusion is very common)

Executors (The Real Workers ⚙️)

***Executor** = Process that actually runs your code*

Responsibilities:

✔️ *Executes tasks*

✔️ *Stores data in cache*

✔️ *Performs transformations*

✔️ *Sends results back to Driver*

📌 ***Important Points***

- *One worker node can have multiple executors*
- *Executors live till the application ends*

Job, Stage, Task (most confusing part 😵)

🔹 ***Job***

Created when you call an action

example: *count(), show()*

🔹 ***Stage***

Job is divided into stages

Based on shuffle boundaries

🔹 ***Task***

Smallest unit of work

Runs on one partition

📌 *Important Points*

💡 *One task = one partition*

How Spark Executes Code (Flow 🔄)

1. You write ***PySpark*** code
2. ***Driver*** creates a Spark job
3. Job is split into ***stages***
4. ***Stages*** are split into tasks
5. ***Tasks*** sent to executors
6. ***Executors*** process data
7. Results returned to ***Driver***
8. Final ***output*** produced 🎉

Spark Architecture with Example

Spark Architecture Example

```
df = spark.read.csv("data.csv")
df.filter(df.age > 30).count()
```

◆ *What happens?*

- **read & filter** ➛ transformations (lazy)
- **count()** ➛ action
- Job is triggered
- Driver ➛ Stages ➛ Tasks
- **Executors** execute
- Count returned to **Driver**

Common Interview Questions

❓ ***What triggers a Spark job?***

Action

❓ ***Who manages memory & CPU?***

Cluster Manager

❓ ***Where does Spark code run?***

Executors

❓ ***Can a worker node have multiple executors?***

Yes

❓ ***What is the smallest unit of work?***

Task

❓ ***Does Spark store data permanently?***

❌ *No (temporary in memory/disk)*

Most common Beginner Mistakes 🚫

❌ Thinking Driver executes data

❌ Confusing Node with Executor

❌ Not understanding Job–Stage–Task

❌ Ignoring partitions

❌ Assuming transformations execute immediately

Fix these ↠ you level up FAST 🚀

One-Slide Interview Summary 🎯

- ***Driver*** ⇝ Brain
- ***Cluster Manager*** ⇝ Resource allocator
- ***Worker Node*** ⇝ Machine
- ***Executor*** ⇝ Runs tasks
- ***Job*** ⇝ Triggered by action
- ***Stage*** ⇝ Shuffle boundary
- ***Task*** ⇝ One partition

💾 *Save*

*Spark Architecture is **interview** gold.*

💬 *Comment*

***Next:** RDD vs DataFrame vs Dataset”*

❤️ *Share*

*with someone learning **PySpark***