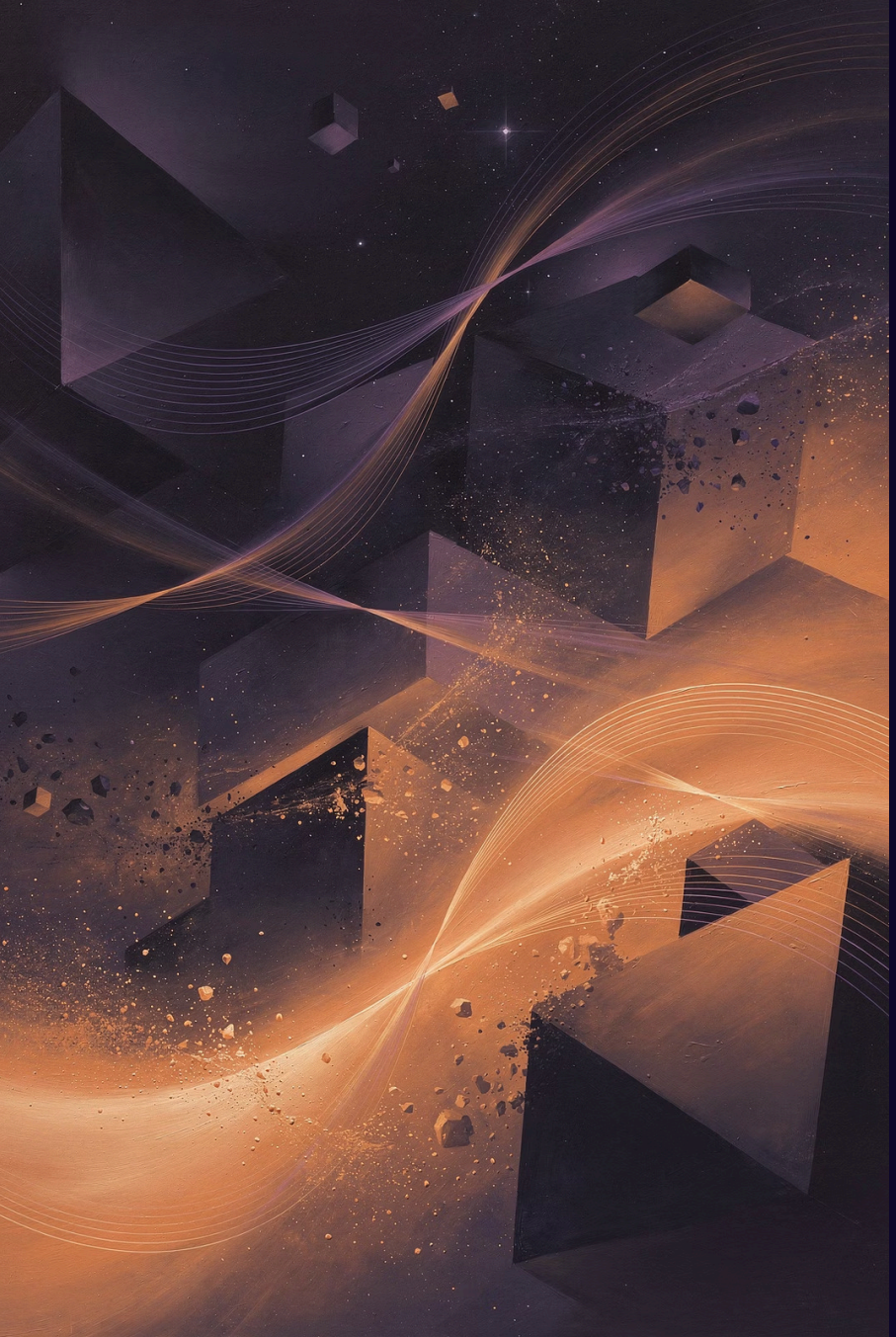


Spark Overview and Features

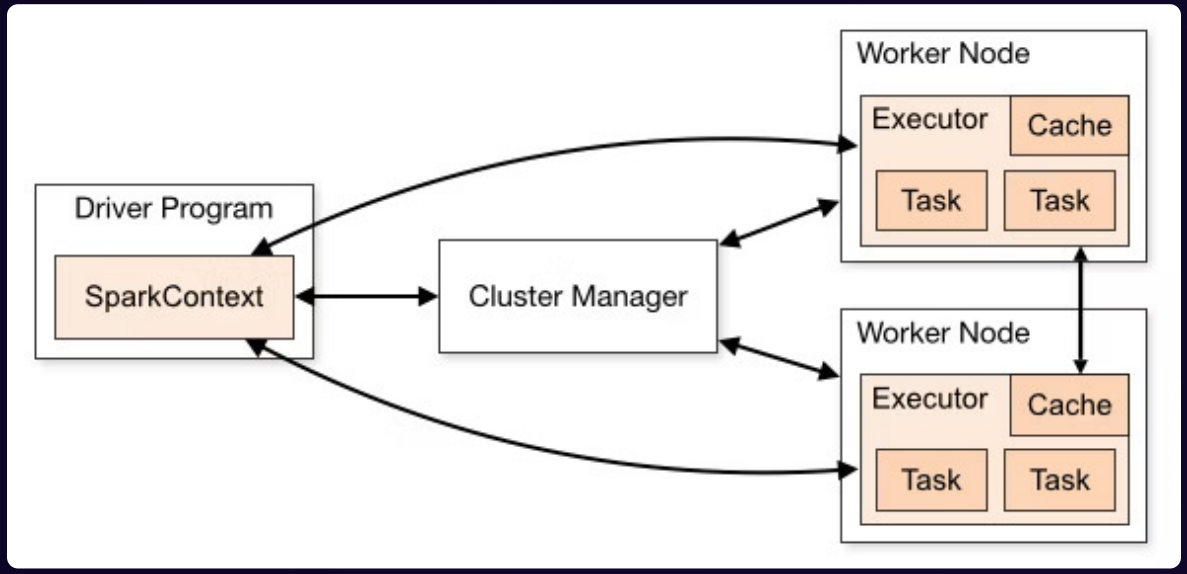
A deep dive into Apache Spark — the distributed data processing engine powering modern analytics, machine learning, and real-time data pipelines.

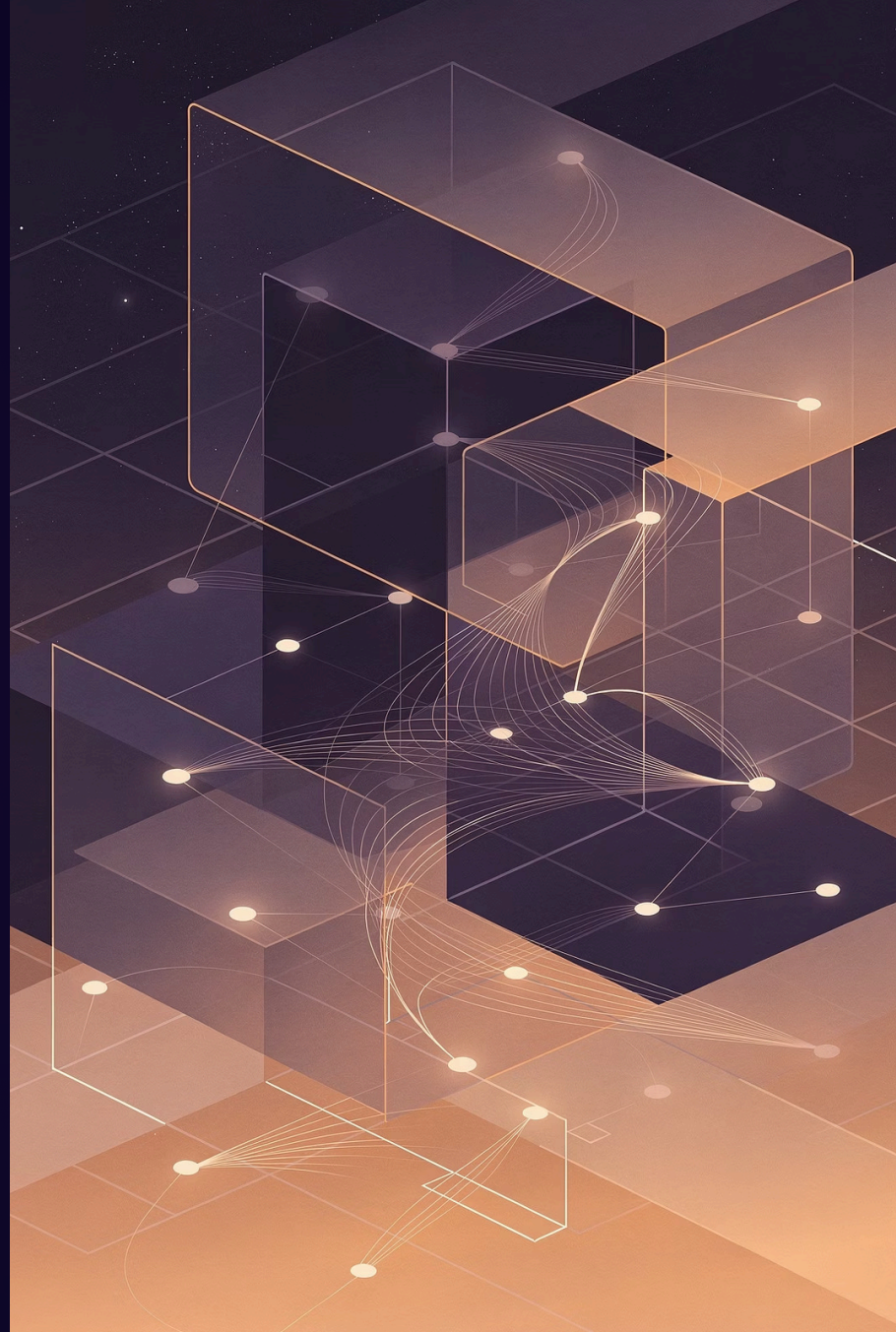
What is Apache Spark?

Apache Spark is a **distributed data processing engine**. Spark distributes data and computation across multiple machines.

It is designed for:

- Large-scale data processing
- Parallel computation
- Batch and real-time analytics
- SQL and Machine Learning workloads


Worker Node
Executor
Cache
Task
Task
Driver Program
SparkContext
Cluster Manager
Worker Node
Executor
Cache
Task
Task



Why Spark Was Created

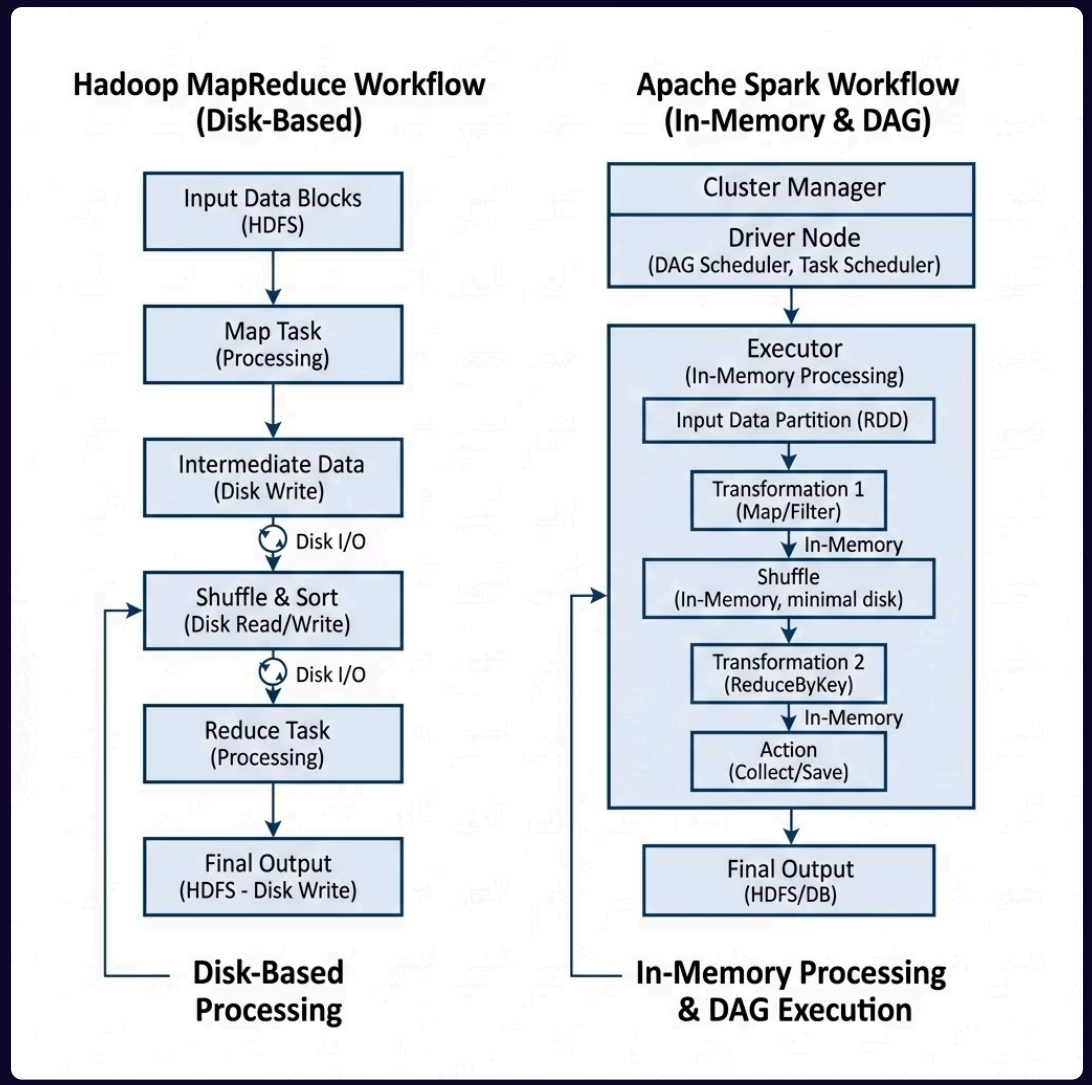
Before Spark, big data processing relied on **Hadoop MapReduce**. Spark was built to improve speed and simplify distributed computing.

Problems with MapReduce

- Disk-heavy processing
- High latency
- Complex programming model
- Separate engines for SQL, ML, Streaming

Spark's Solution

Spark was designed from the ground up to address every one of these pain points — delivering in-memory speed, a unified engine, and a developer-friendly API.


Hadoop MapReduce Workflow
(Disk-Based)
Input Data Blocks
(HDFS)
Map Task
(Processing)
Intermediate Data
(Disk Write)
Disk I/O
Shuffle & Sort
(Disk Read/Write)
Disk I/O
Reduce Task
(Processing)
Final Output
(HDFS - Disk Write)
Disk-Based
Processing
Apache Spark Workflow
(In-Memory & DAG)
Cluster Manager
Driver Node
(DAG Scheduler, Task Scheduler)
Executor
(In-Memory Processing)
Input Data Partition (RDD)
Transformation 1
(Map/Filter)
In-Memory
Shuffle
(In-Memory, minimal disk)
Transformation 2
(ReduceByKey)
In-Memory
Action
(Collect/Save)
Final Output
(HDFS/DB)
In-Memory Processing
& DAG Execution

Core Features of Spark

In-Memory Processing

- Stores intermediate results in memory
- Reduces disk I/O
- Improves performance

Lazy Evaluation

- Builds execution plan first
- Executes only when action is triggered

More Spark Features

Fault Tolerance

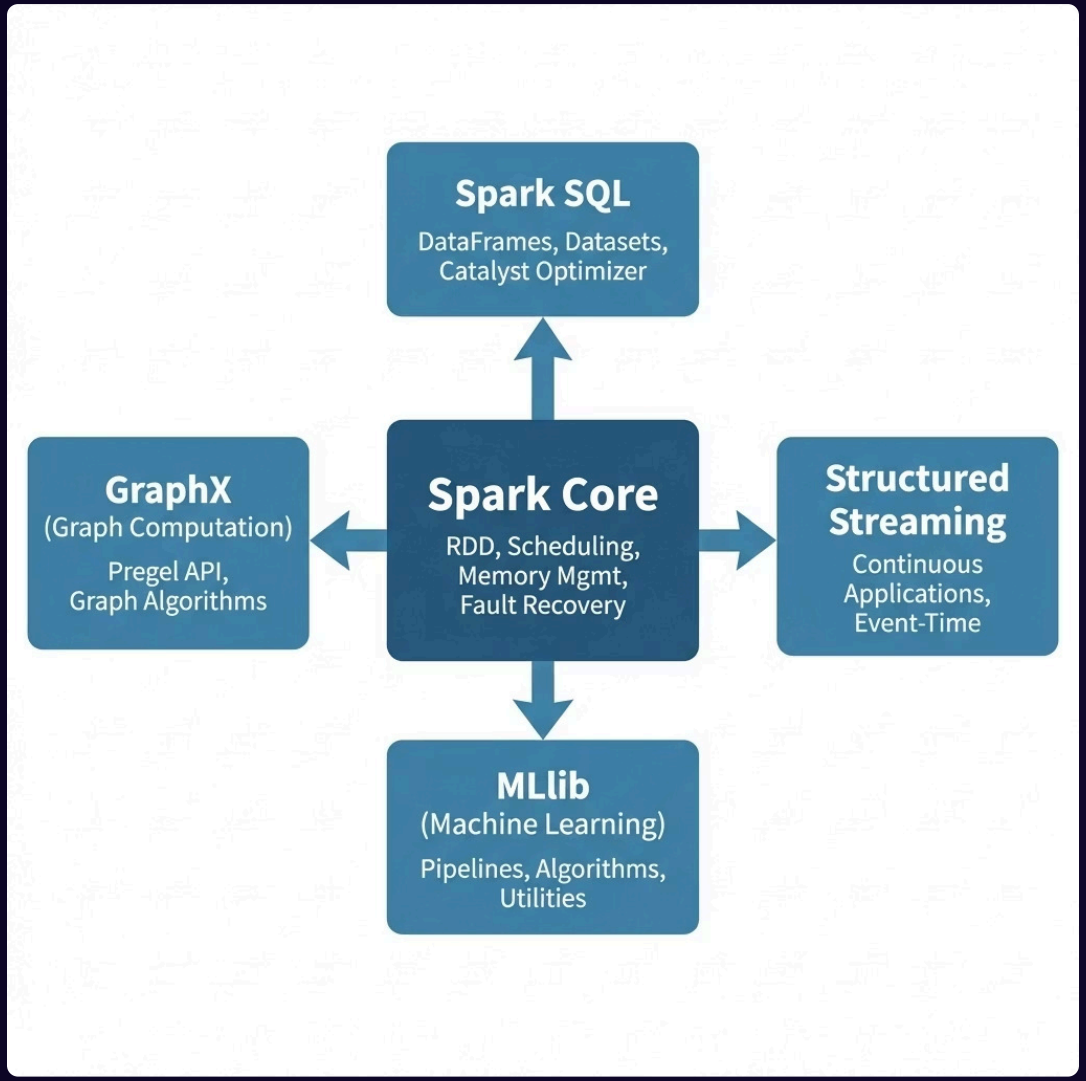
Uses lineage to recompute lost partitions

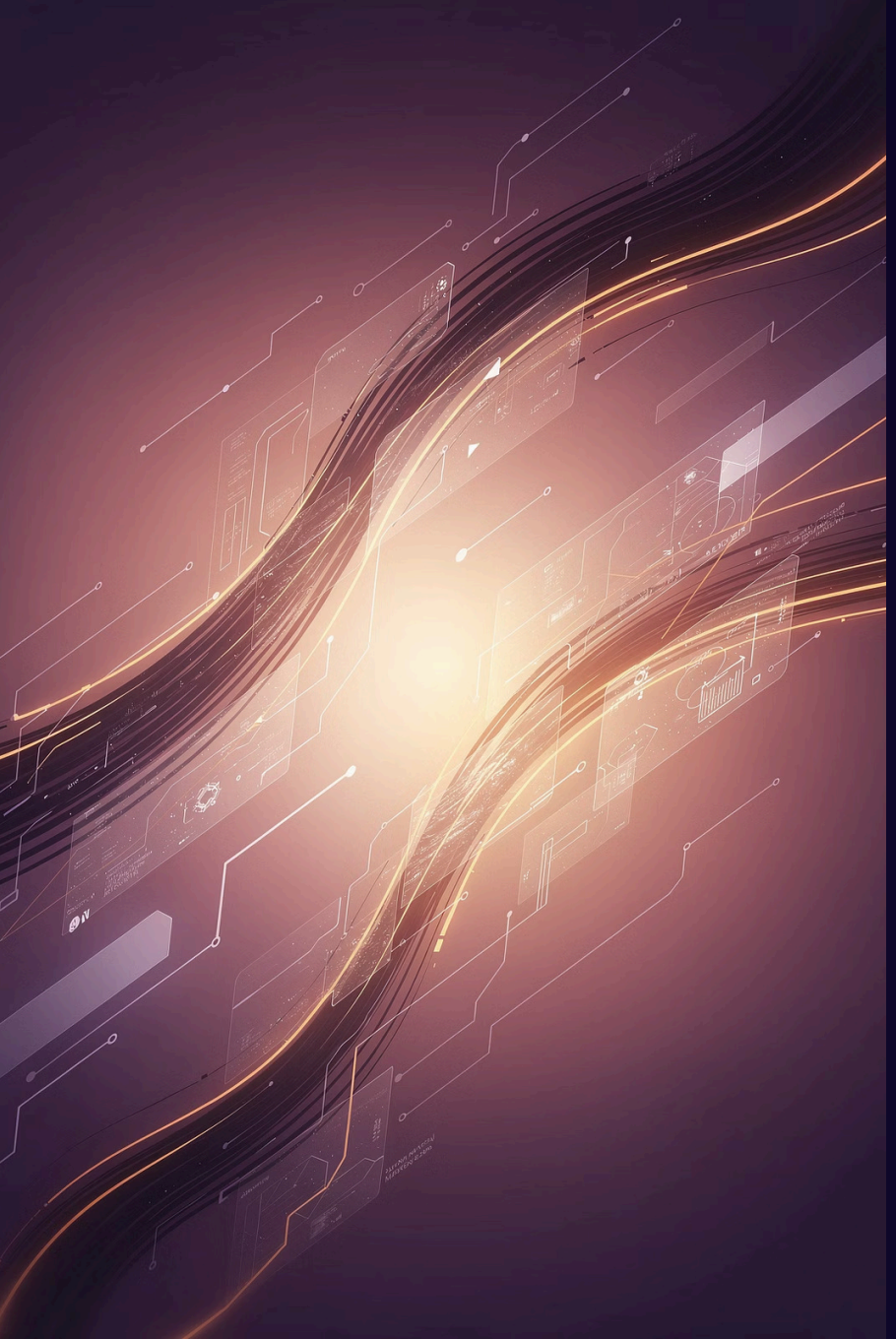
Scalability

Works on laptop, clusters, and in the cloud

Unified Engine

SQL · Streaming · ML · Graph processing


Spark SQL
DataFrames, Datasets, Catalyst Optimizer
GraphX
(Graph Computation)
Pregel API, Graph Algorithms
Spark Core
RDD, Scheduling, Memory Mgmt, Fault Recovery
Structured Streaming
Continuous Applications, Event-Time
MLlib
(Machine Learning)
Pipelines, Algorithms, Utilities



Importance of Spark in Data Processing

Modern data systems require a unified platform that handles every workload. **Spark provides all of this in one engine.**

- Batch processing
- Real-time analytics
- Interactive queries
- Machine learning pipelines

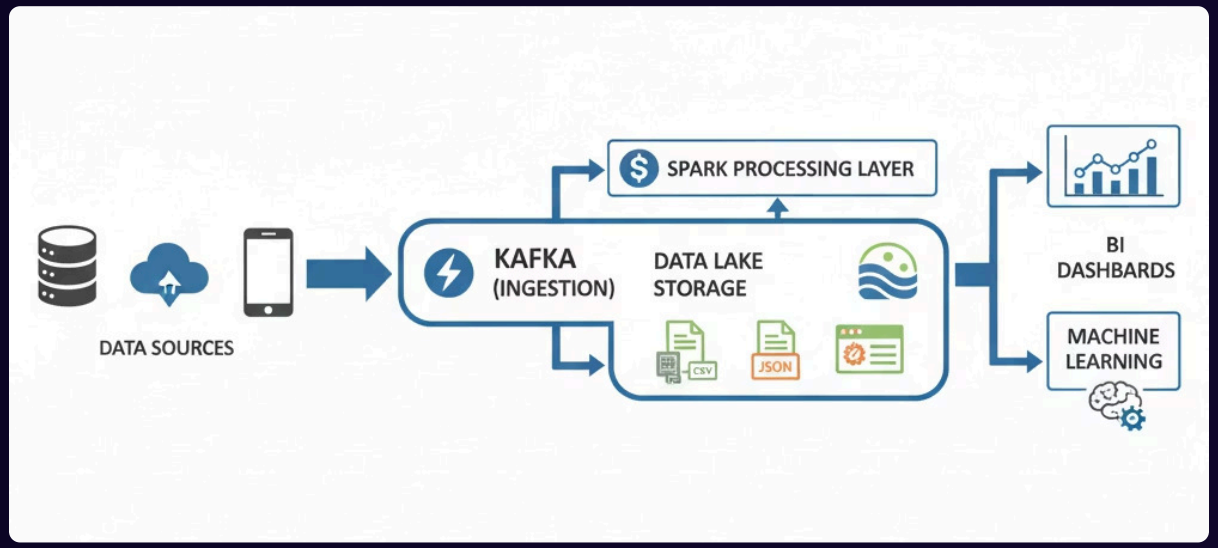
Where Spark is Used

Use Cases

- ETL pipelines
- Data lakes
- Data warehouses
- Streaming applications
- ML feature engineering

Spark integrates with:

- Kafka
- Hive
- Cloud storage
- Kubernetes


SPARK PROCESSING LAYER
KAFKA
(INGESTION)
DATA LAKE
STORAGE
CSV
JSON
BI
DASHBARDS
MACHINE
LEARNING
DATA SOURCES

Hadoop vs Spark Stack

Hadoop Stack

01

HDFS

Storage

02

YARN

Resource Manager

03

MapReduce

Processing

Characteristics:

- Disk-based
- Batch-oriented
- Slower for iterative tasks

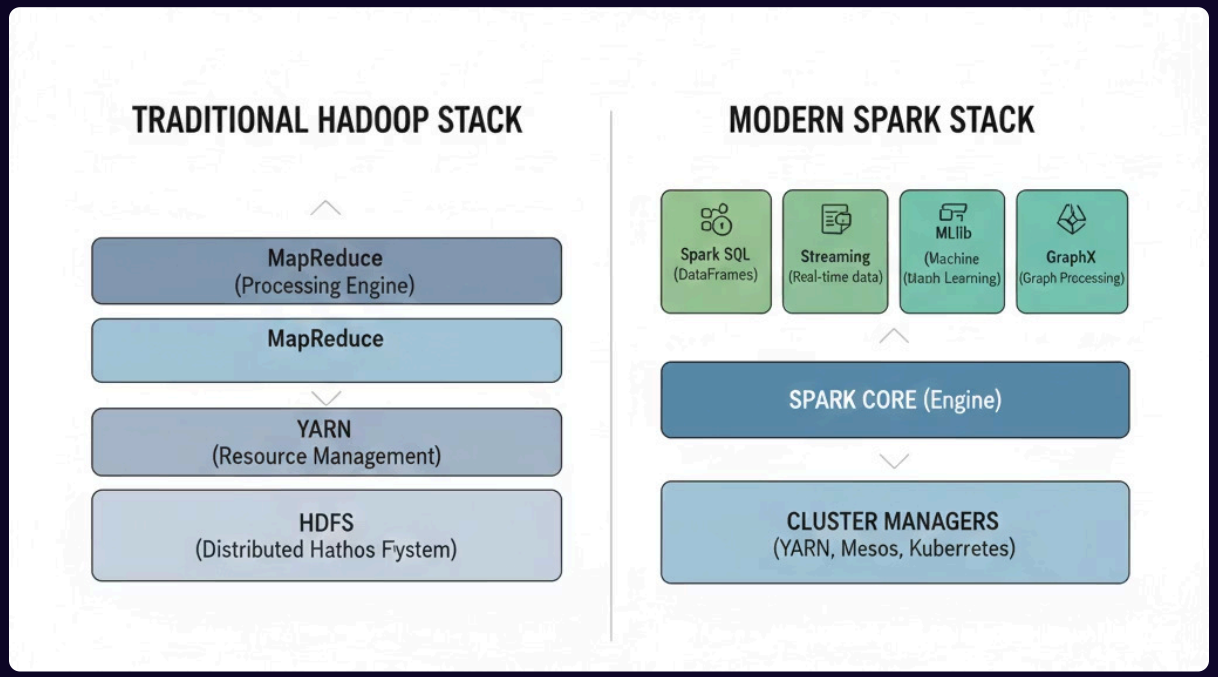
Spark Stack

Spark Components

- Spark Core
- Spark SQL
- Structured Streaming
- MLlib
- GraphX

Runs on:

- HDFS
- Cloud Storage
- YARN
- Kubernetes


TRADITIONAL HADOOP STACK
MapReduce
(Processing Engine)
MapReduce
YARN
(Resource Management)
HDFS
(Distributed Hathos Fystem)
MODERN SPARK STACK
Spark SQL
(DataFrames)
Streaming
(Real-time data)
MLlib
(Machine
GraphX
(Graph Processing)
SPARK CORE (Engine)
CLUSTER MANAGERS
(YARN, Mesos, Kuberretes)

Key Differences

Feature	Hadoop MapReduce	Spark
Processing	Disk-based	Memory-first
Speed	Slower	Faster
Iterative Jobs	Inefficient	Efficient
Streaming	Separate system	Built-in
Ease of Use	Complex	Developer-friendly

When to Use Hadoop vs Spark

Use Hadoop When:

- You need distributed storage (HDFS)
- Pure batch workloads
- Legacy ecosystem compatibility

Use Spark When:

- You need fast analytics
- You need streaming + batch
- You build ML pipelines
- You need interactive SQL

Summary

Distributed Analytics Engine

Spark is a distributed analytics engine.

Performance Improvement

It improves performance over Hadoop MapReduce.

Unified Workloads

It supports batch, streaming, SQL, and ML.

Modern Data Architectures

It is widely used in modern data architectures.