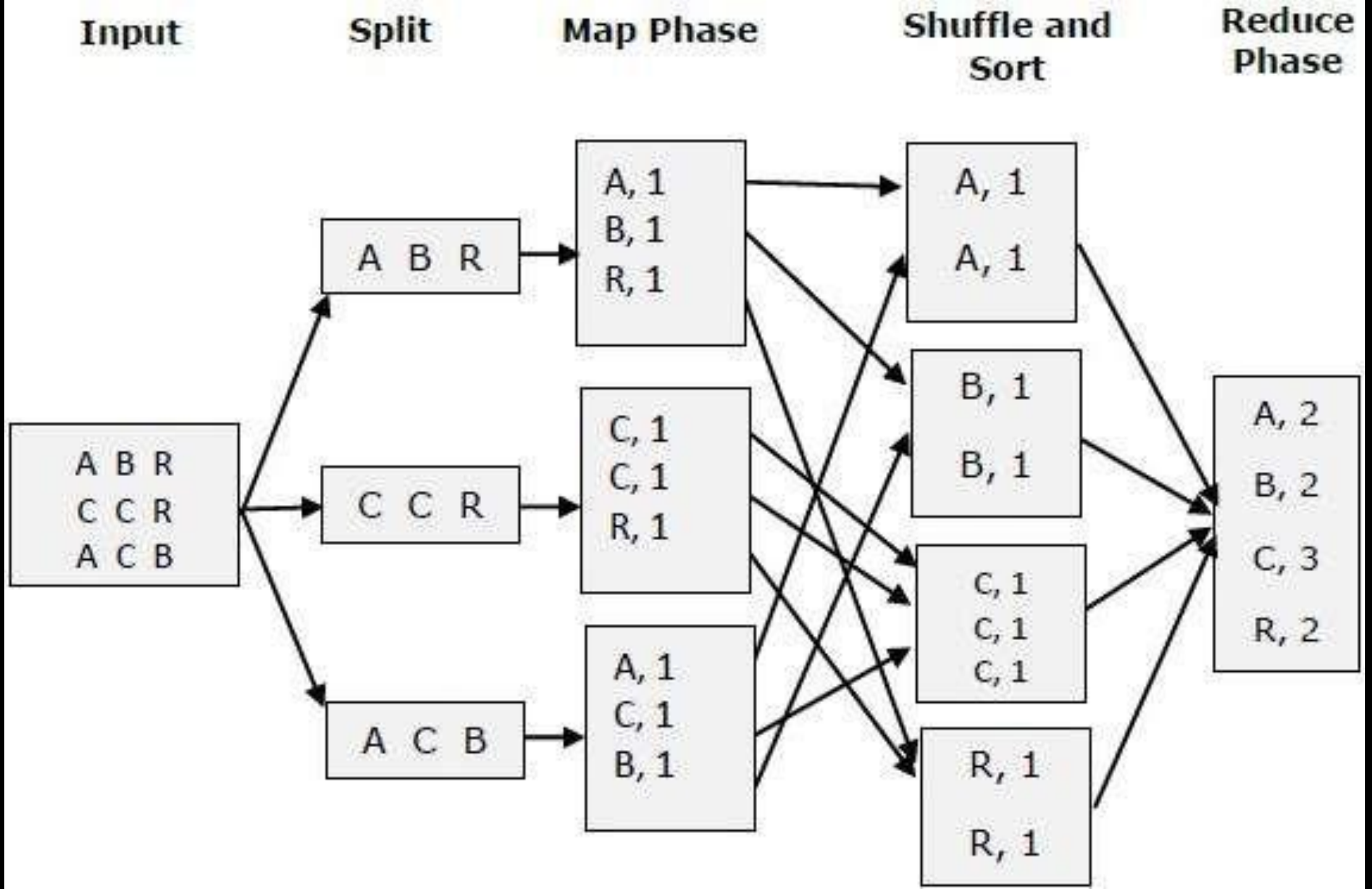
Databricks

Apache spark

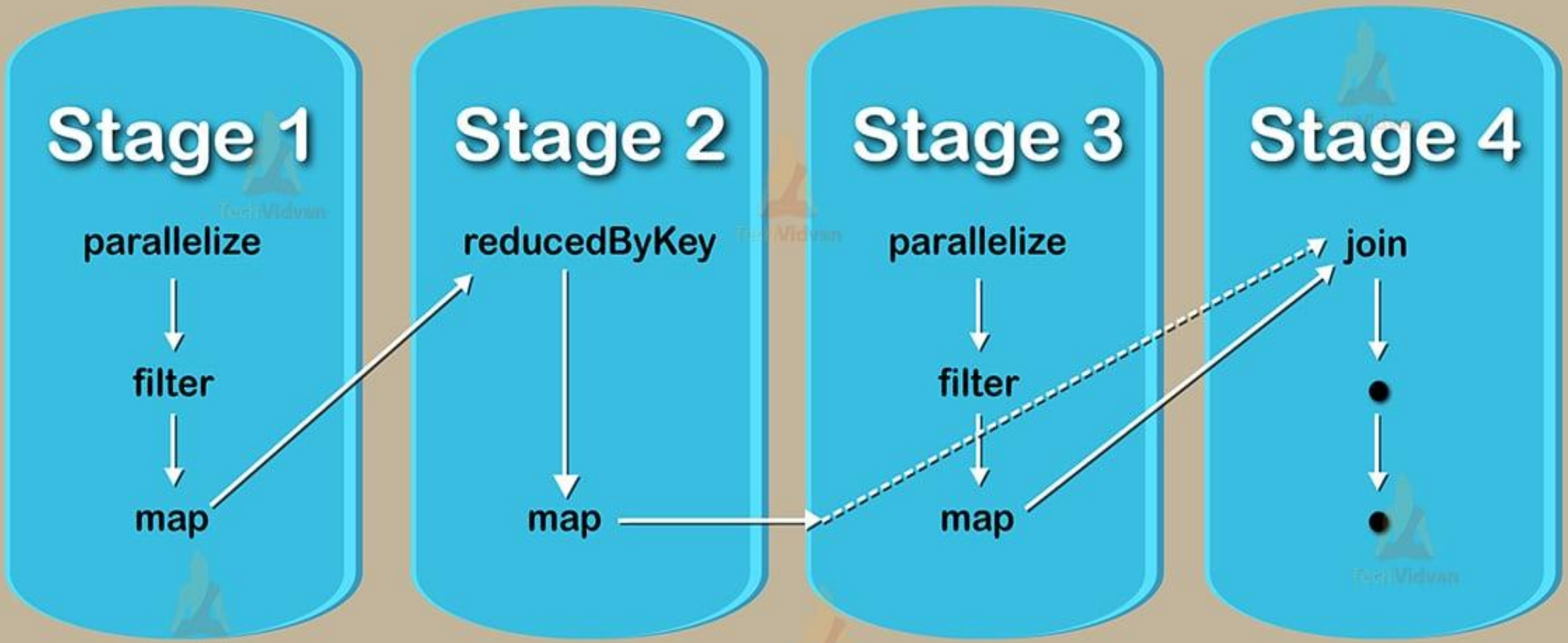
- Tool to analyze data at large scale.
- **Distributed in-memory processing system** for fast big data analytics at any scale
- **Multi-language support** - Java, Scala, Python, and R APIs available.
- **Unified platform** - handles batch processing, real-time analytics, ML, and graph processing with code reuse.
- **High performance** - 100x faster than MapReduce using in-memory caching and optimized execution.

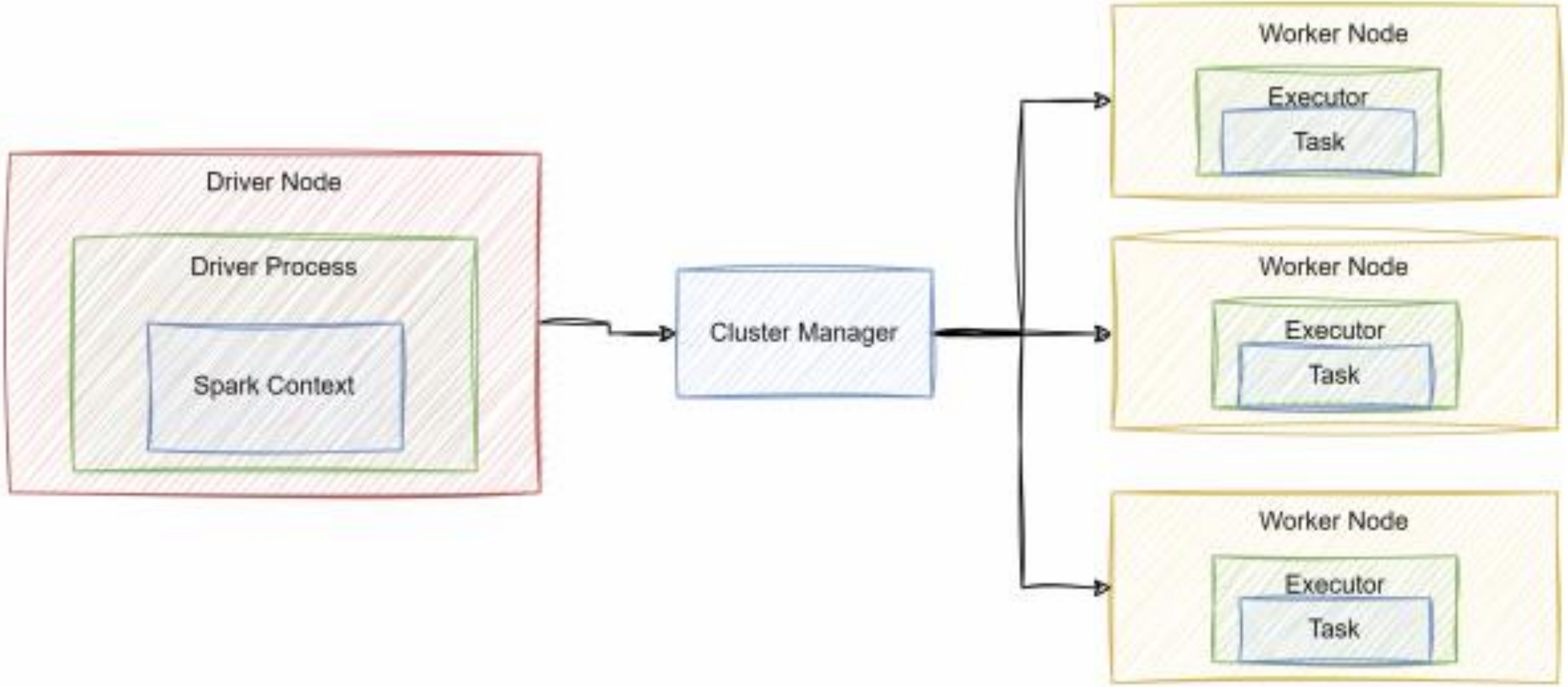
Why sparks over MapReduce?

- In-memory processing is 100x faster
  - "DAG engine" for efficiency
  - Less disk I/O, lower latency
- Supports batch, stream, ML, graph
- Simple APIs, optimized performance


Input
Split
Map Phase
Shuffle and Sort
Reduce Phase
A B R
C C R
A C B
A B R
C C R
A C B
A, 1
B, 1
R, 1
C, 1
C, 1
R, 1
A, 1
C, 1
B, 1
A, 1
A, 1
B, 1
B, 1
C, 1
C, 1
C, 1
R, 1
R, 1
A, 2
B, 2
C, 3
R, 2

DAG Visualisation





Spark Application Architecture

Source: [Medium]

Prefer data bricks for :

Unified platform for data engineering, analytics, and machine learning — all in one workspace.

Built on Apache Spark, offering **high-speed, scalable, distributed processing**.

Serverless and managed clusters remove the burden of infrastructure setup and tuning.

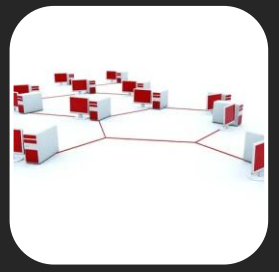
Collaborative notebooks let data engineers, analysts, and scientists work together seamlessly.

Integration with all major clouds (AWS, Azure, GCP) for flexibility and scalability.

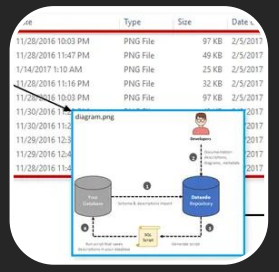
How Data Bricks work?

Workspace

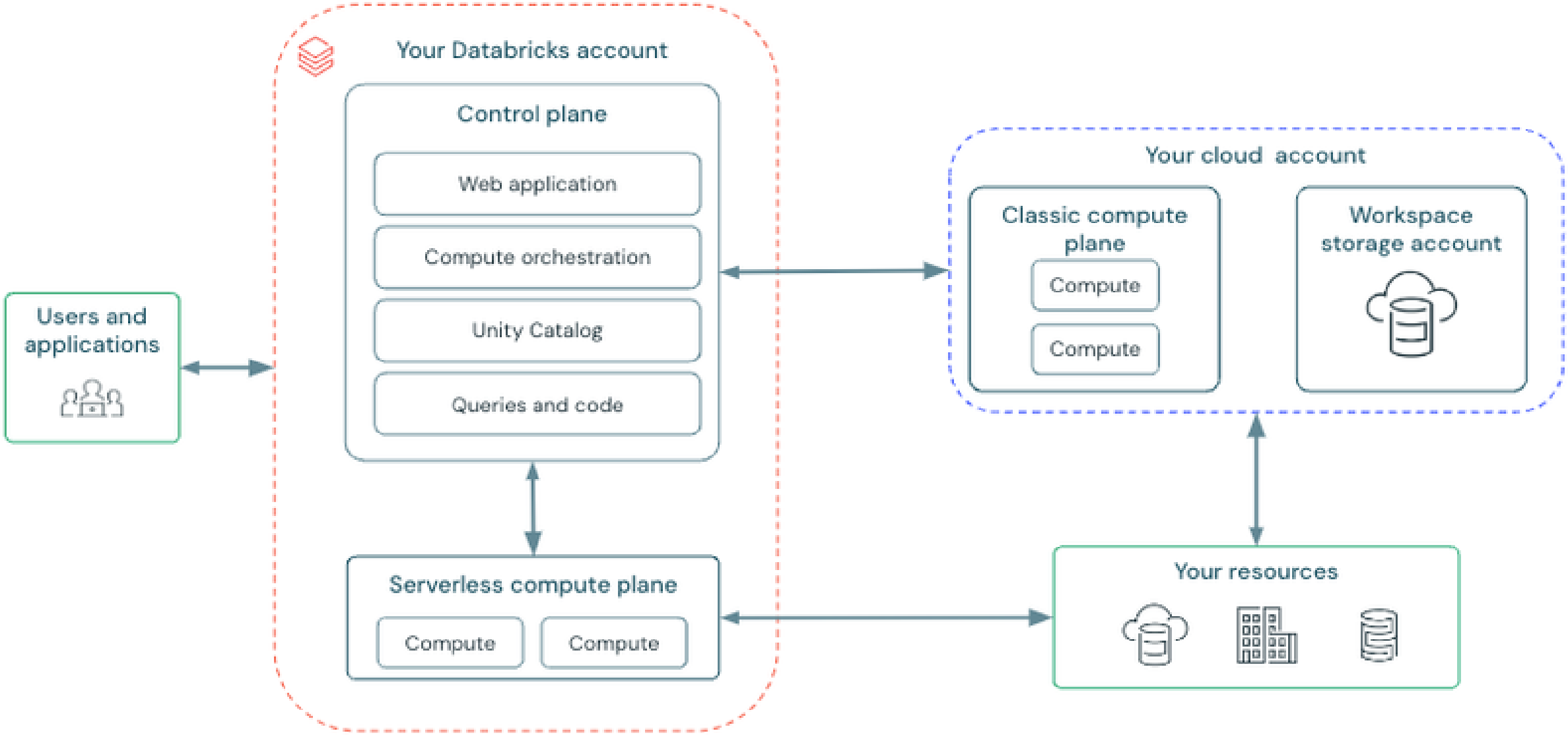
Notebooks



Cluster



metastore



Databricks Architecture Diagram

Source: [Microsoft]

Example with python

- Create data frame using python and export it to excel/csv format
- Work with spark using python
  - check spark in notebook
  - create sample data
  - use spark functions
  - visualize data using spark

THANK YOU