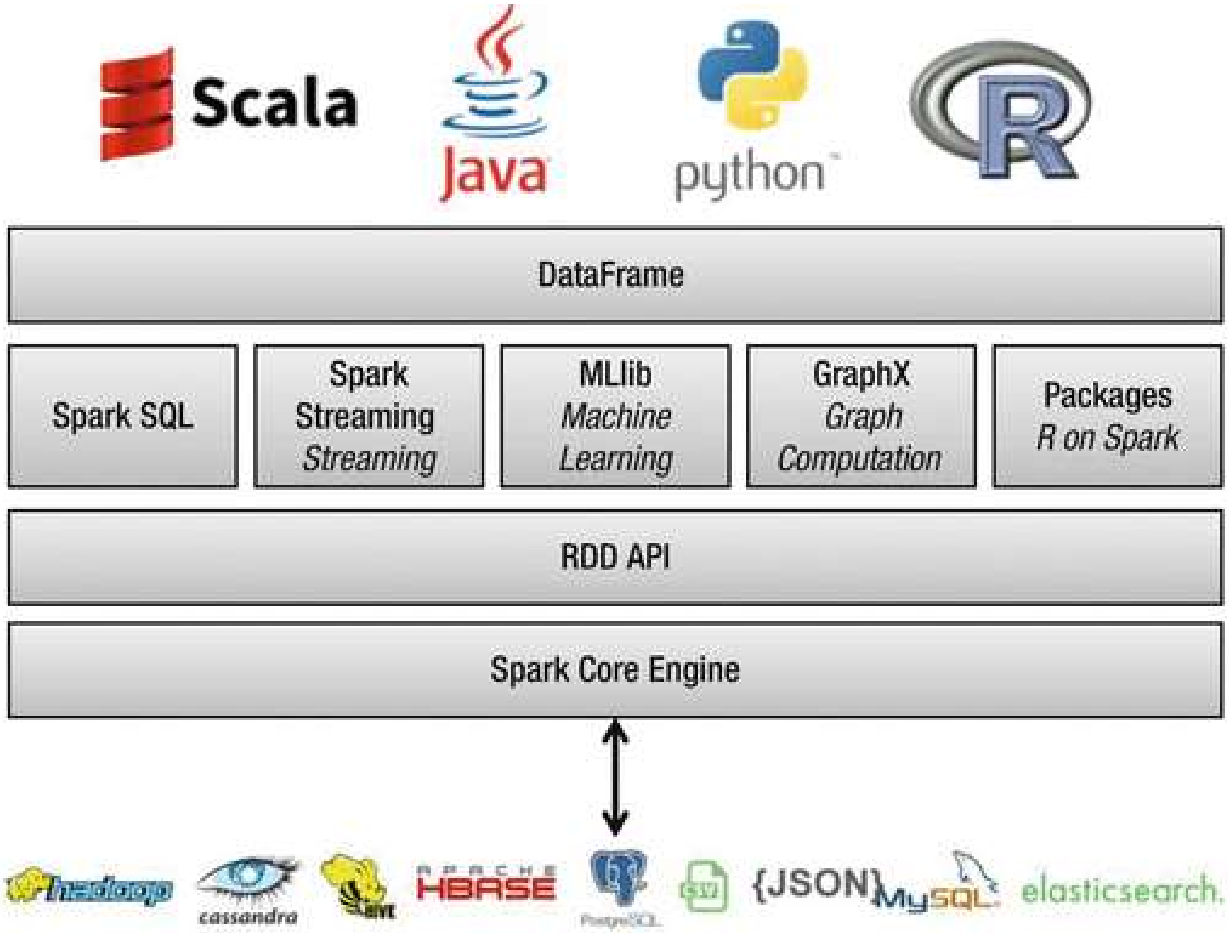
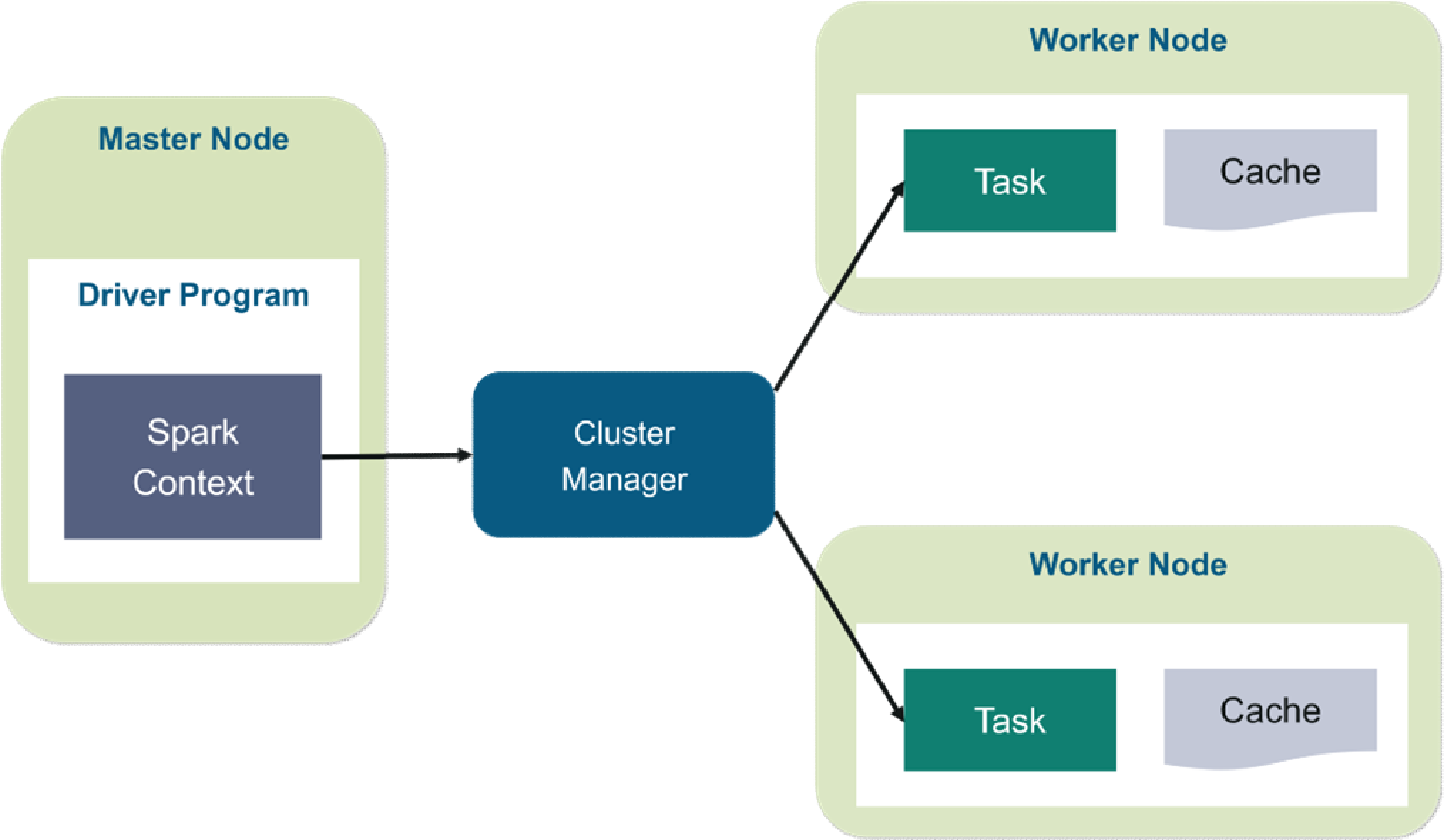
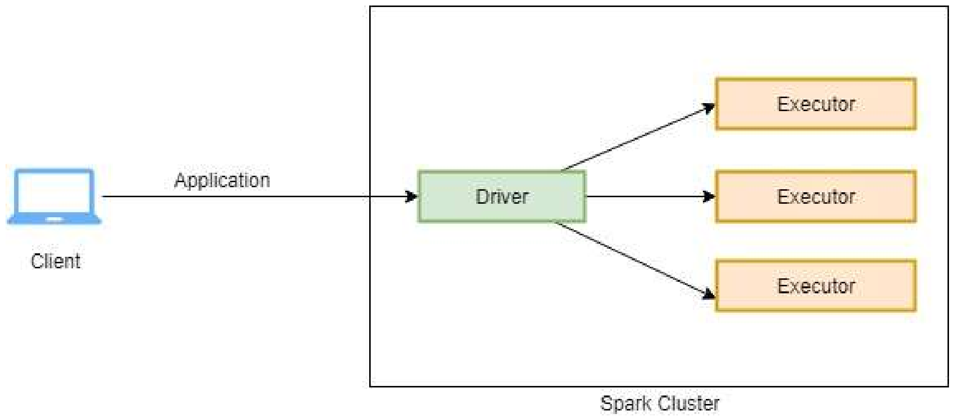
Apache Sparks Diagrams


Scala
Java
python
R
DataFrame
Spark SQL
Spark Streaming Streaming
MLlib Machine Learning
GraphX Graph Computation
Packages R on Spark
RDD API
Spark Core Engine
hadoop
cassandra
HIVE
APACHE HBASE
PostgreSQL
CSV
{JSON}
MySQL
elasticsearch.

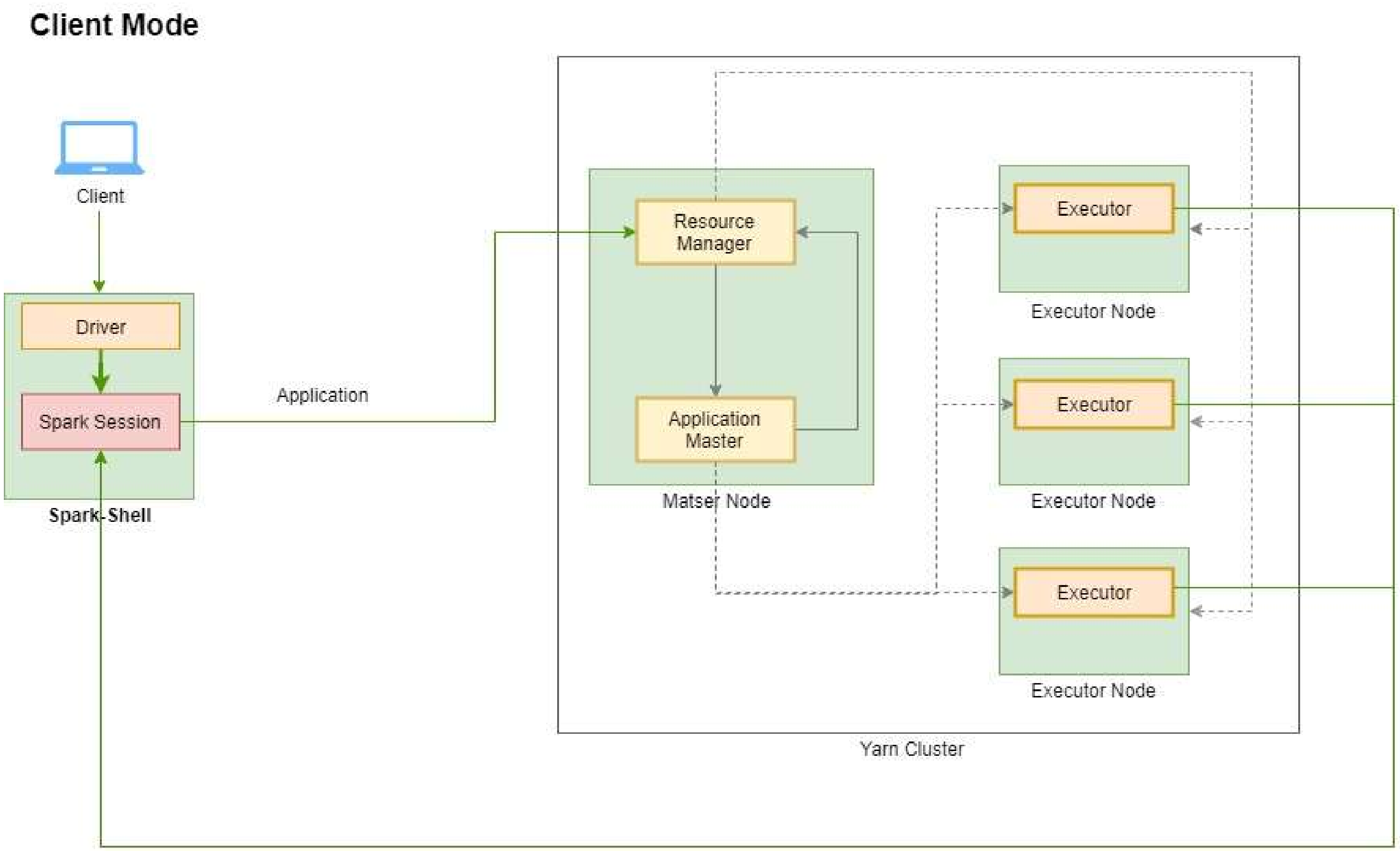

Master Node
Driver Program
Spark Context
Cluster Manager
Worker Node
Task
Cache
Worker Node
Task
Cache

Apache Spark Execution

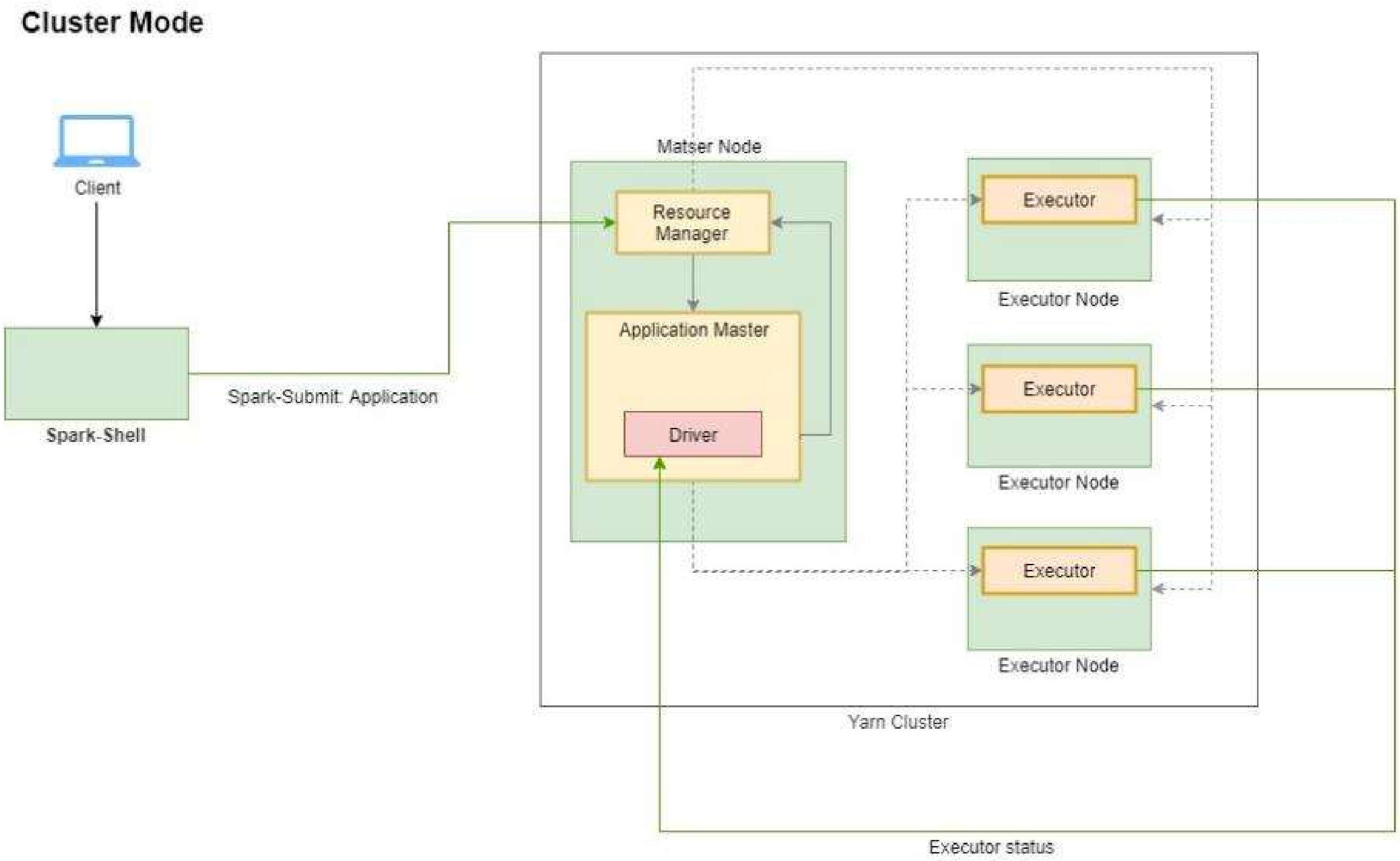
- For every application submitted on spark cluster spark creates a dedicated Driver process and bunch of Executor processes.
- Driver process is responsible for analyzing, distributing, scheduling and monitoring of executor processes.
- Whereas the executor process is only responsible for running the task they were assigned by drivers and reporting the status back to the driver.



Apache Spark Execution – Client Mode

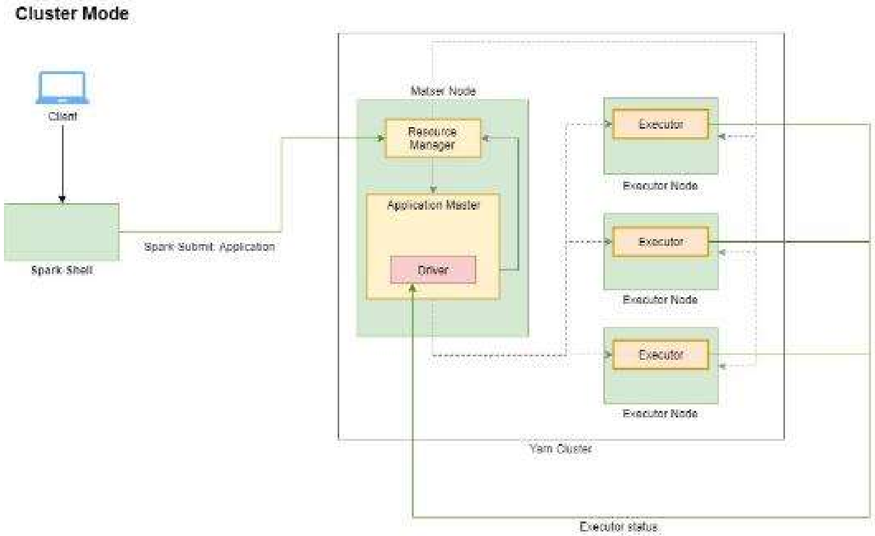


Apache Spark Execution – Cluster Mode



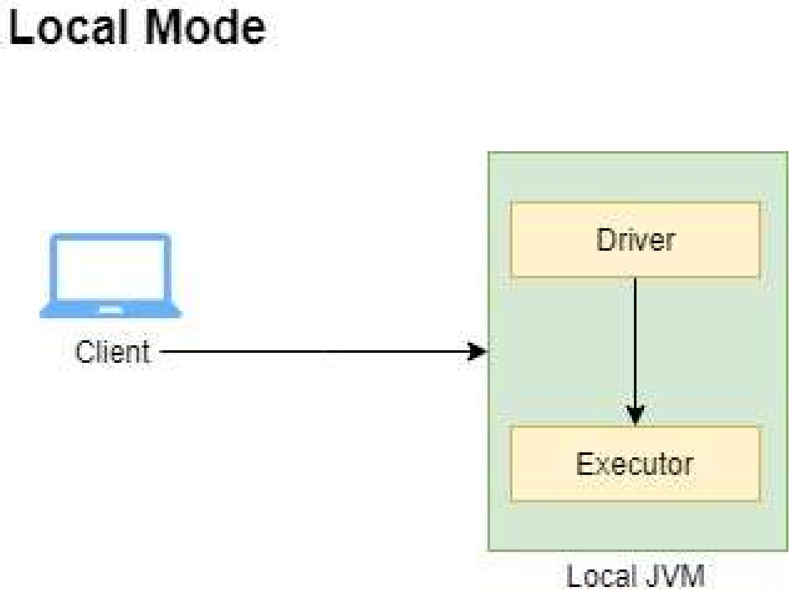
Apache Spark Execution – Cluster Mode

- Spark client submits the packed application to yarn request manager.
- Yarn request manager then creates the AM container. The Spark-Driver is also created in AM container.
- Rest of the flow is same as mentioned in the client mode.

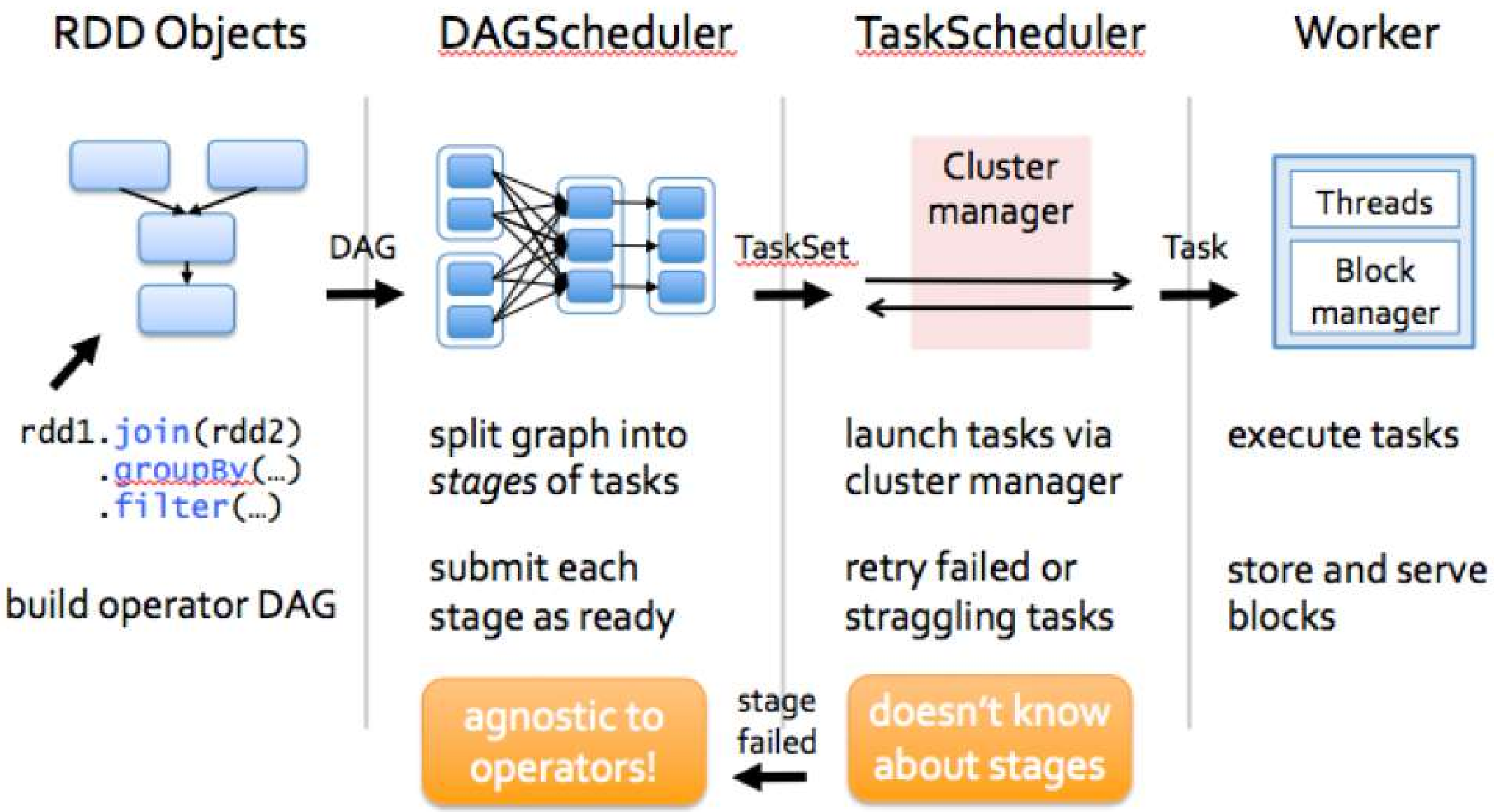


Apache Spark Execution – Local Mode

- There is another mode in which spark can be run locally without any cluster requirement.
- This mode is suitable for scenarios when we do not have enough resources to create cluster.
- But in this mode you get only one executor and both the Driver and Excuter runs in the same JVM.



How Sparks work?



```
rdd1.join(rdd2)
    .groupBy(…)
    .filter(…)
```

build operator DAG

split graph into *stages* of tasks

submit each stage as ready

agnostic to operators!

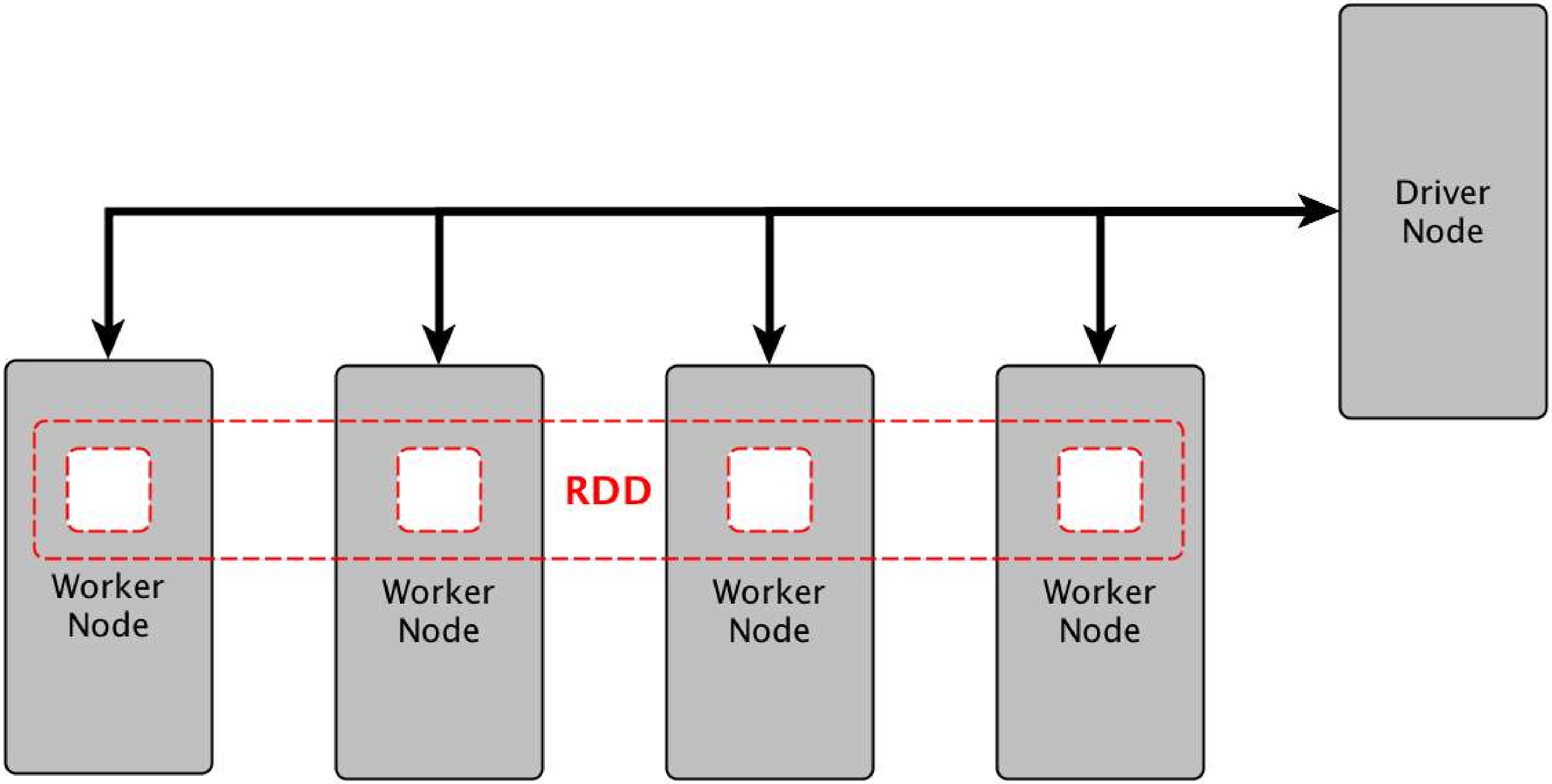
launch tasks via cluster manager

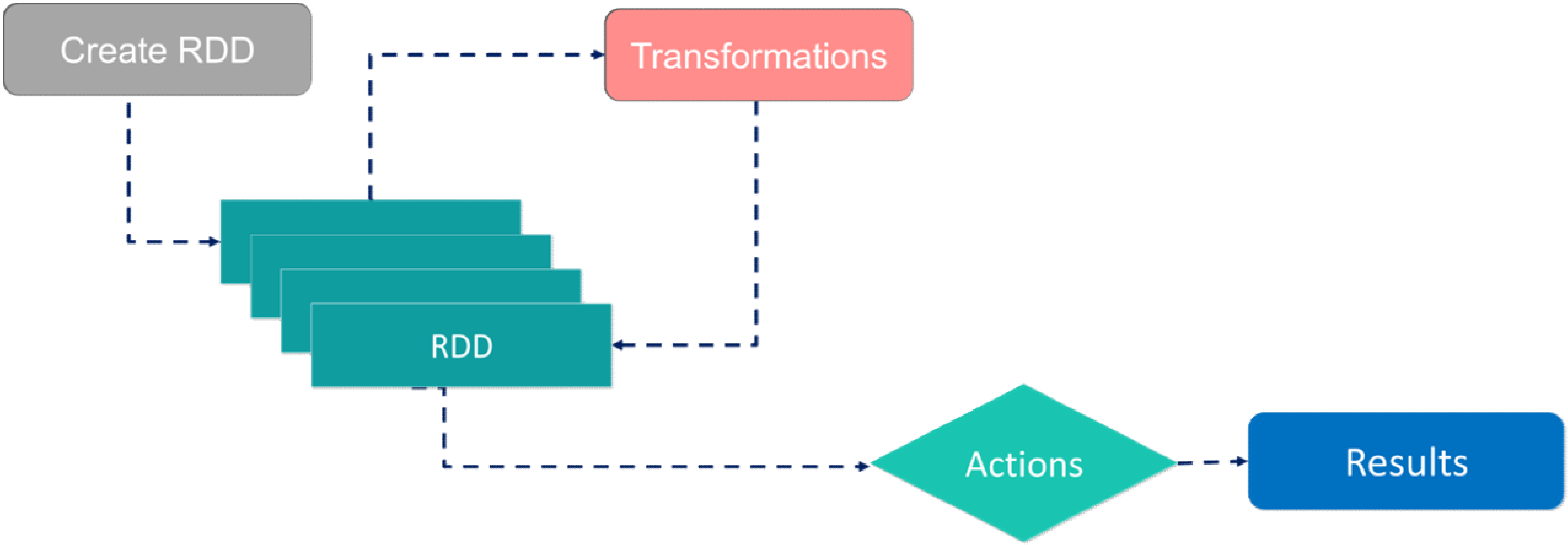
retry failed or straggling tasks

doesn't know about stages

execute tasks

store and serve blocks


Driver Node
RDD
Worker Node
Worker Node
Worker Node
Worker Node


Create RDD
Transformations
RDD
Actions
Results

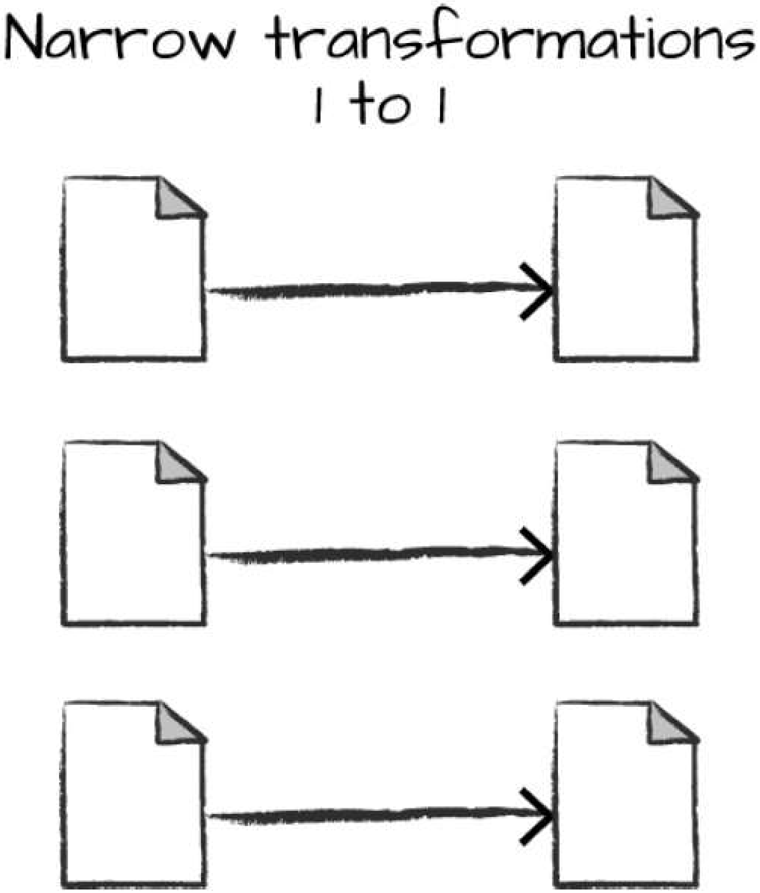
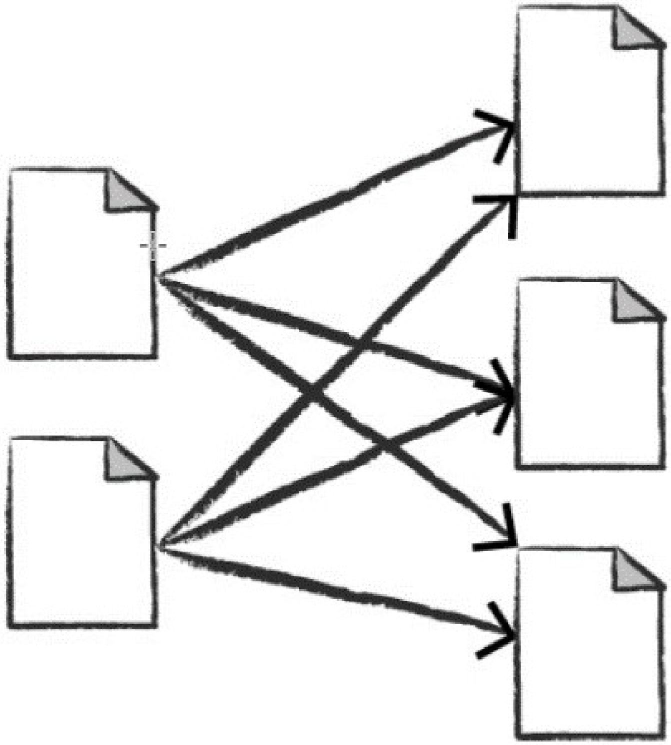


Figure 2-4. A narrow dependency

Wide transformations (shuffles) 1 to N



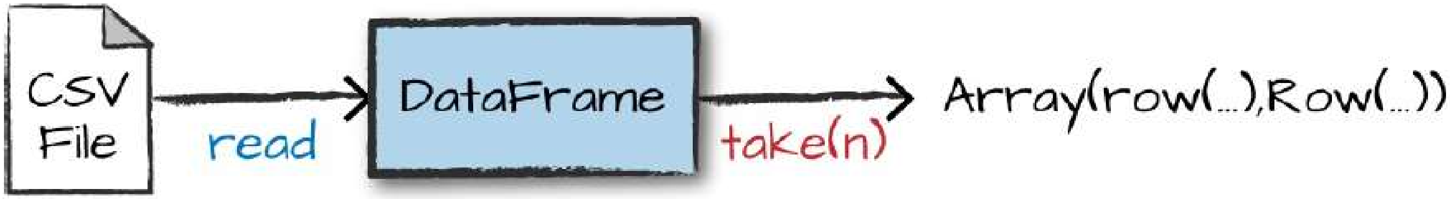


Figure 2-7. Reading a CSV file into a DataFrame and converting it to a local array or list of rows

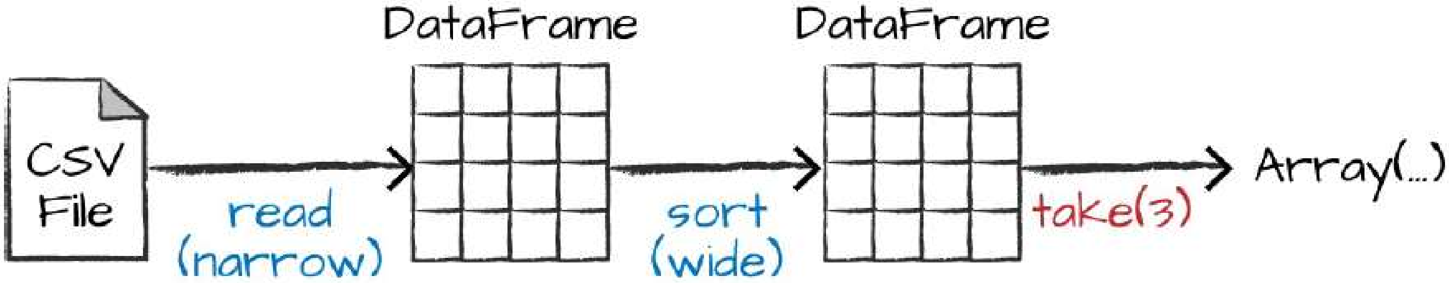


Figure 2-8. Reading, sorting, and collecting a DataFrame

Catalyst Optimizer

- Spark SQL uses an optimizer called catalyst to optimize all the queries
- This optimizer makes queries run much faster
- An optimizer automatically finds out the most efficient plan to execute data operations specified in the user’s program.
- logical plan — series of algebraic or language constructs, as for example: SELECT, GROUP BY or UNION keywords in SQL. It’s usually represented as a tree.
- physical plan — Concerns low level operations.

Catalyst Optimizer

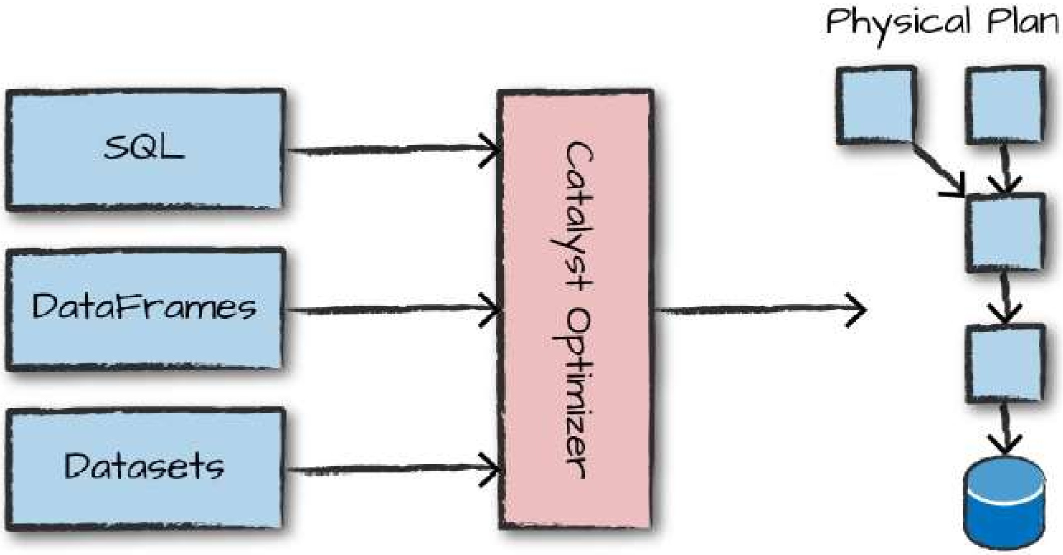


Figure 4-1. The Catalyst Optimizer

Catalyst Optimizer

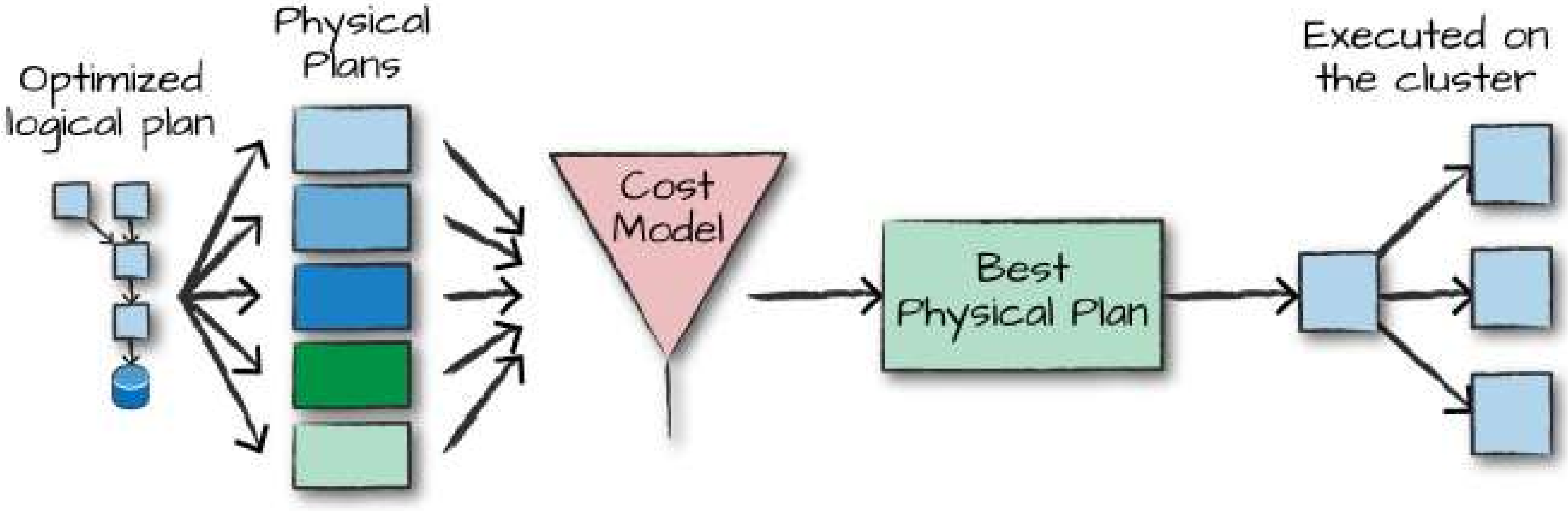


Figure 4-3. The physical planning process

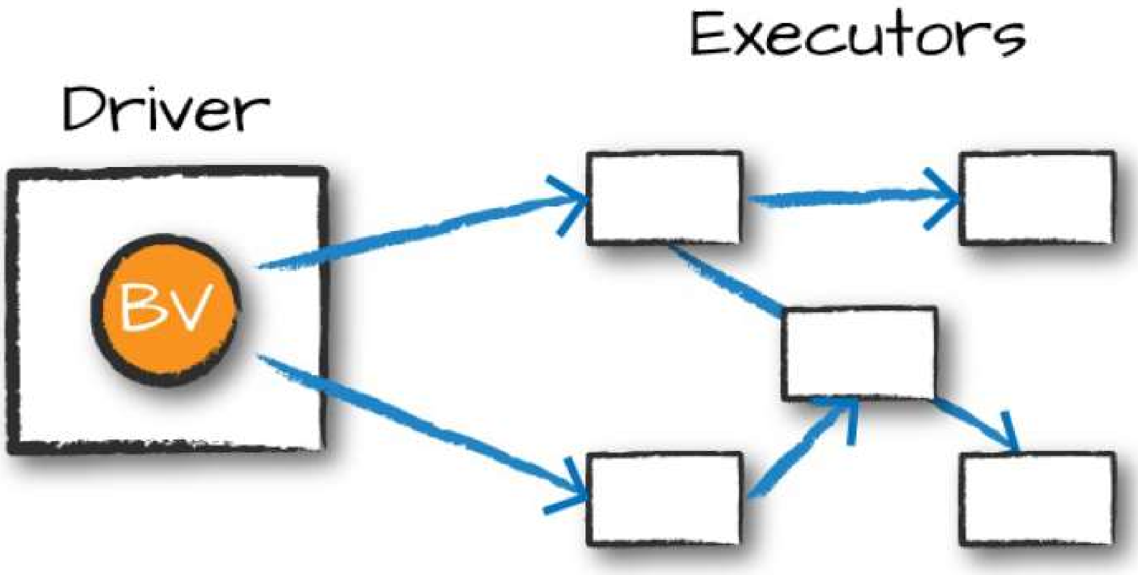


Figure 14-1. Broadcast variables

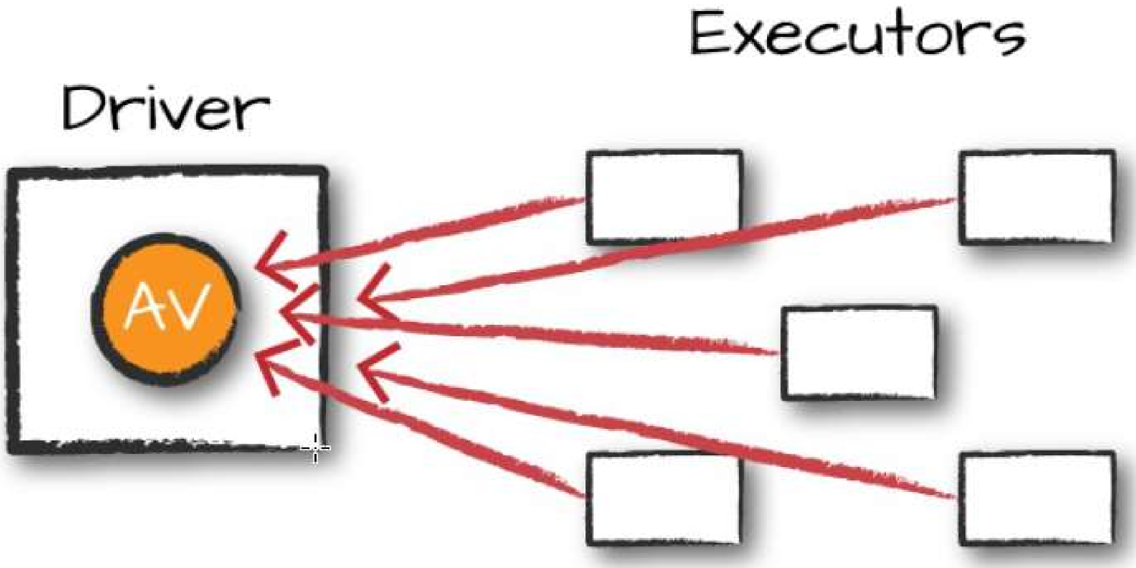


Figure 14-2. Accumulator variable

Thanks