PySpark

Oscar Mike Claure Cabrera

anwb & DIGITAL POWER

Content

Overview of relevant concepts:

- Introduction
- Core components
- Unified framework
- RDDs
- Lazy vs eager evaluation
- Catalyst optimizer
- Shuffling
- Partitioning

Performance evaluation

- Explore query plans
- Spark UI
- Type of joins

anwb & DIGITAL POWER

Introduction

What is Apache Spark?

A distributed data processing engine designed for big data and large-scale computation

Spark's key features

- Speed: In-memory processing for faster computation
- Scalability: Handles petabytes of data across large clusters
- Versatility: Supports multiple programming languages (API, Java, Scala, R, SQL)

Common use cases:

- ETL pipelines
- Data Analytics
- Machine Learning workflows

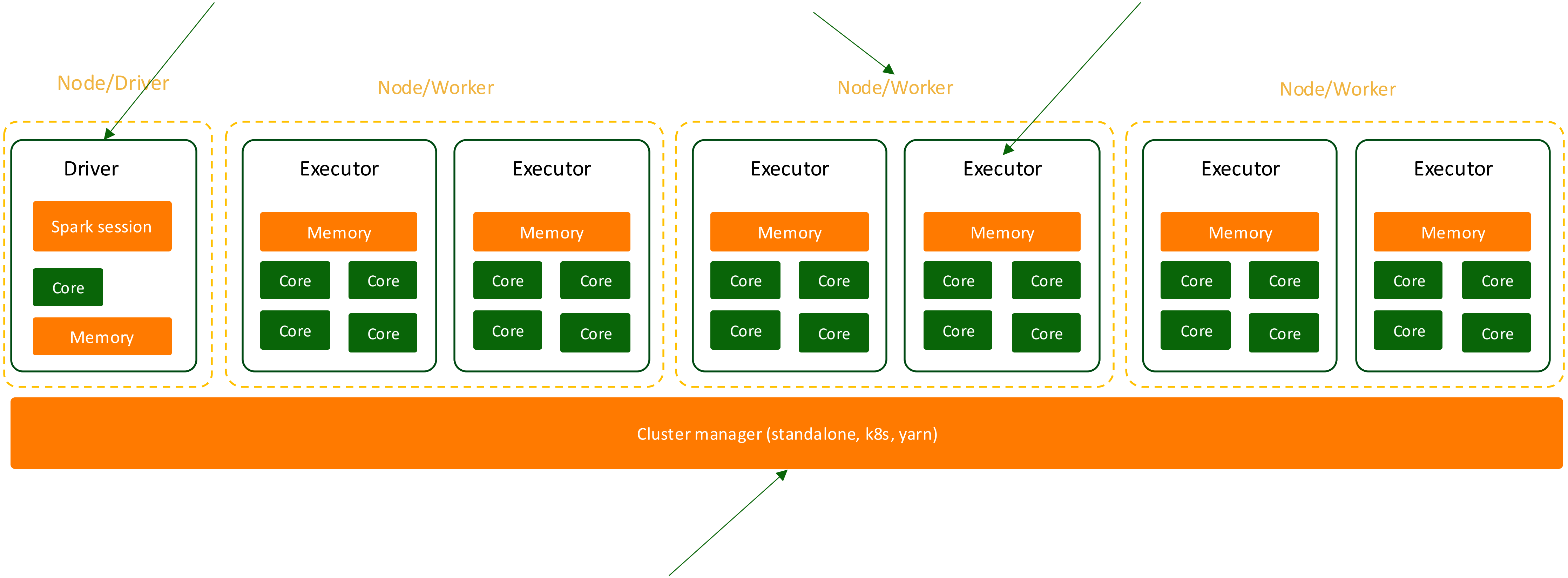
&

Spark’s core components

Central coordinator for the spark application. Converts user code into tasks, schedules and monitors their execution.

Machines hosting the executors

Processes running on workers, responsible for executing tasks and storing data



Node/Driver

Driver

Spark session

Core

Memory

Node/Worker

Executor

Memory

Core Core

Core Core

Executor

Memory

Core Core

Core Core

Node/Worker

Executor

Memory

Core Core

Core Core

Executor

Memory

Core Core

Core Core

Node/Worker

Executor

Memory

Core Core

Core Core

Executor

Memory

Core Core

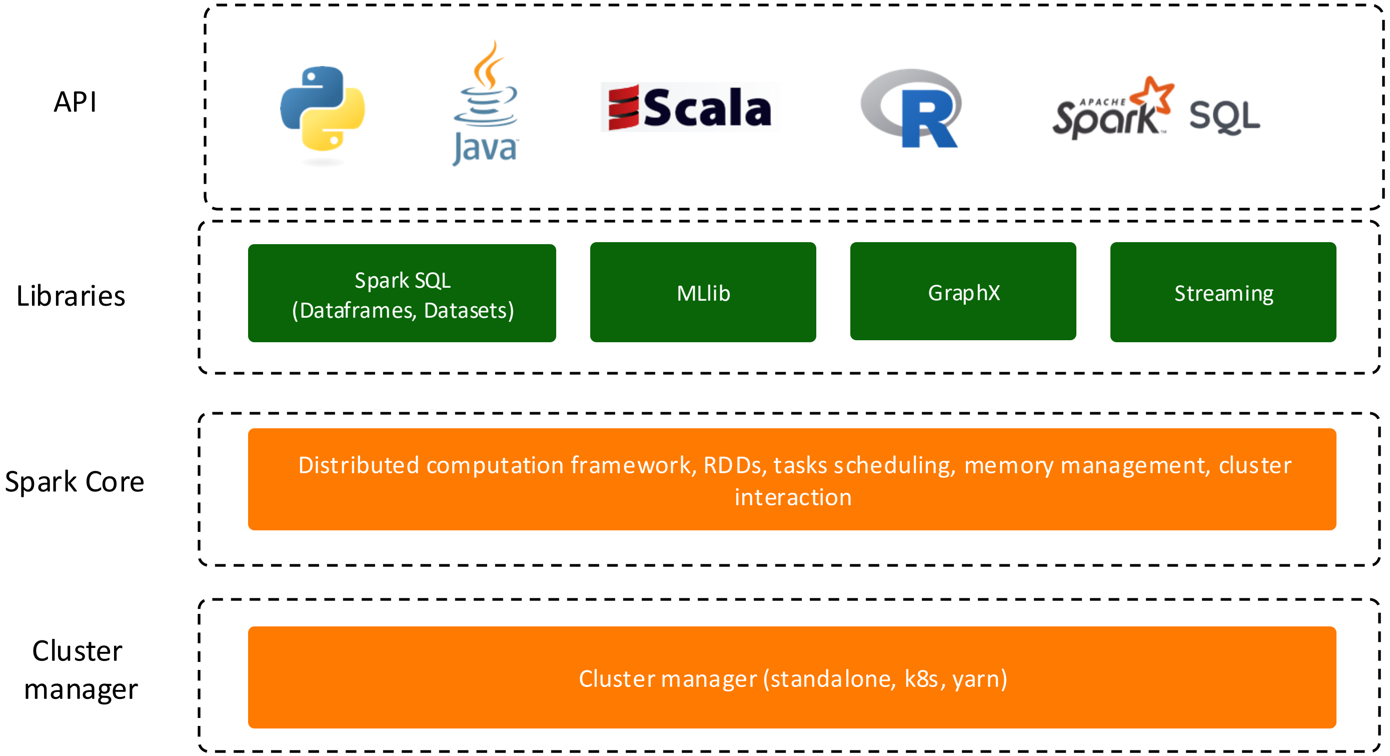
Core Core

Cluster manager (standalone, k8s, yarn)

Orchestrates resources across the cluster

&

Spark’s unified framework



API

Libraries

Spark SQL (Dataframes, Datasets)

MLlib

GraphX

Streaming

Spark Core

Distributed computation framework, RDDs, tasks scheduling, memory management, cluster interaction

Cluster manager

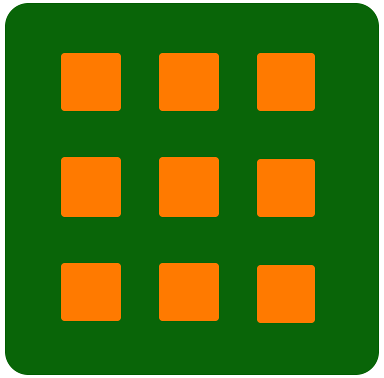
Cluster manager (standalone, k8s, yarn)

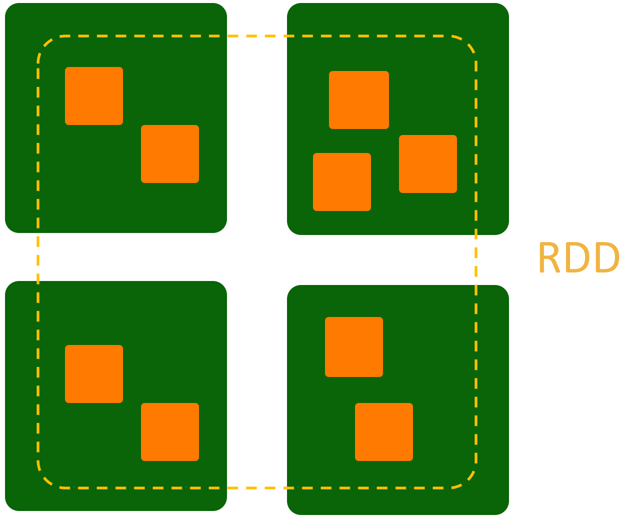
RDDs

Resilient distributed dataset (RDD), is a fault-tolerant, immutable collection of elements that can be operated in parallel across a cluster of machines.

```
my_var = [1, 2, 3, 4, 5, …, N]
```

```
my_rdd = sc.parallelize([1, 2, 3, 4, 5, …, N])
```





🟩 Node

🟧 Data point

Lazy vs eager evaluation

Lazy

Evaluation of expressions is delayed until their results are needed

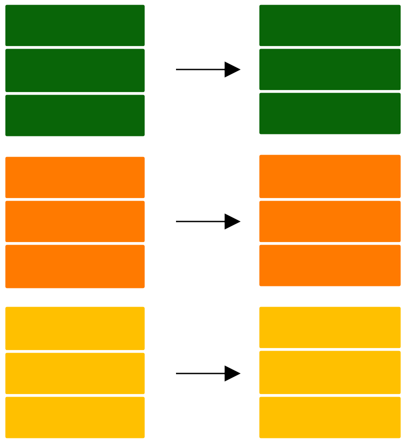
Eager

Evaluation of expressions occur every time a new expression is declared.

Transformations and actions

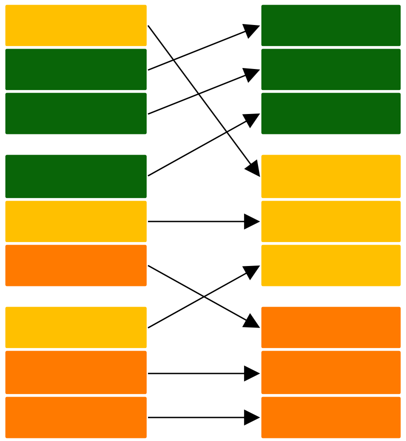
Transformations

Narrow transformations



Filter
FlatMap
Map
WithColumn
Sample

Wide transformations



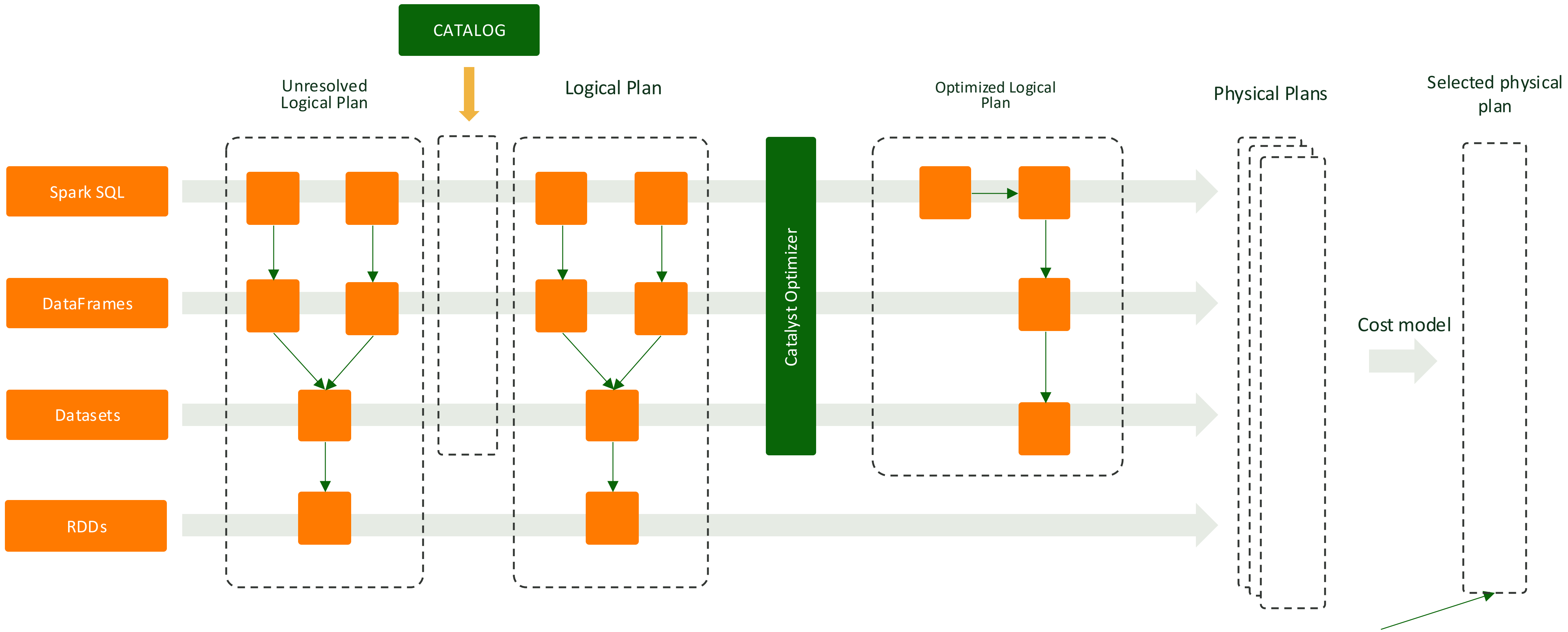
GroupBy
Join
Distinct
Repartition
Coalesce
SortBy
Distinct

Actions

Count
Take
Collect
Save
Show
ToPandas

&

Catalyst optimizer and Tungsten engine

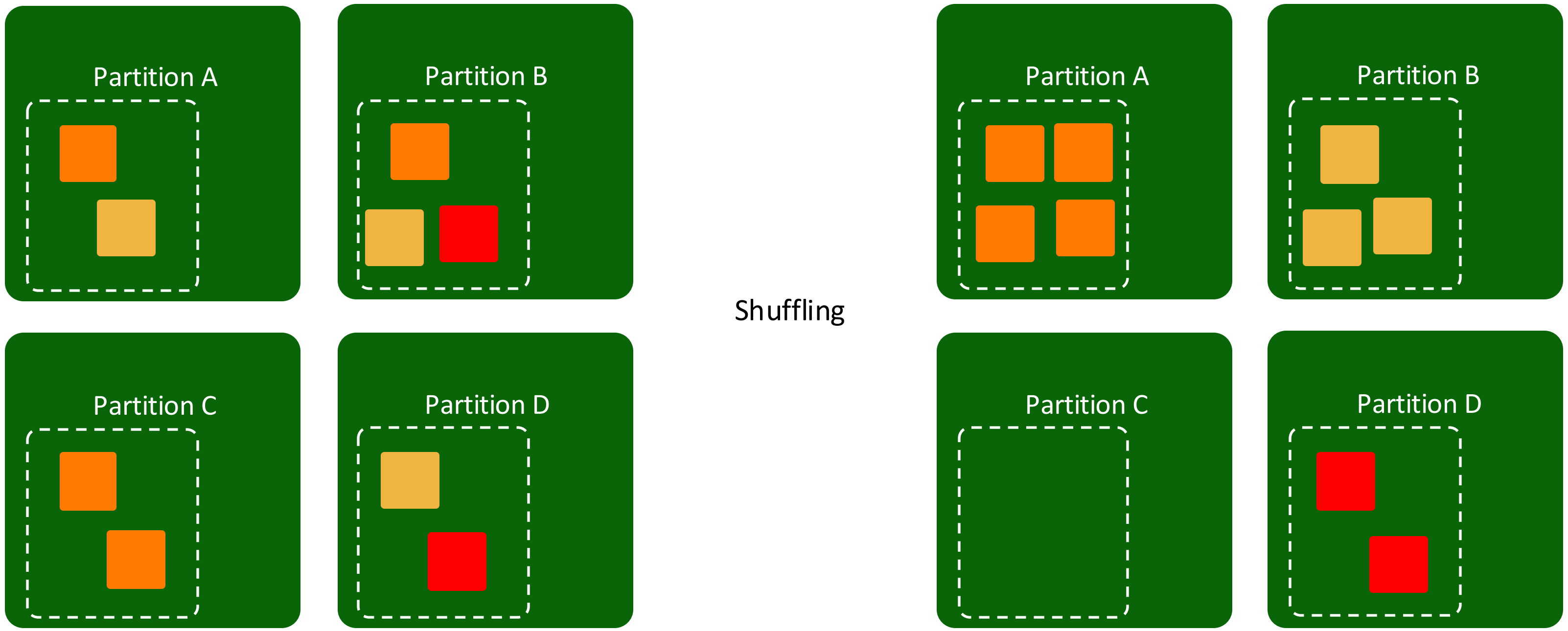


How the queries will be executed into the cluster.
Execution order, data partitioning, join strategies.

Source: Key topics in Apache Spark, Catalyst Optimizer

anwb & DIGITAL POWER

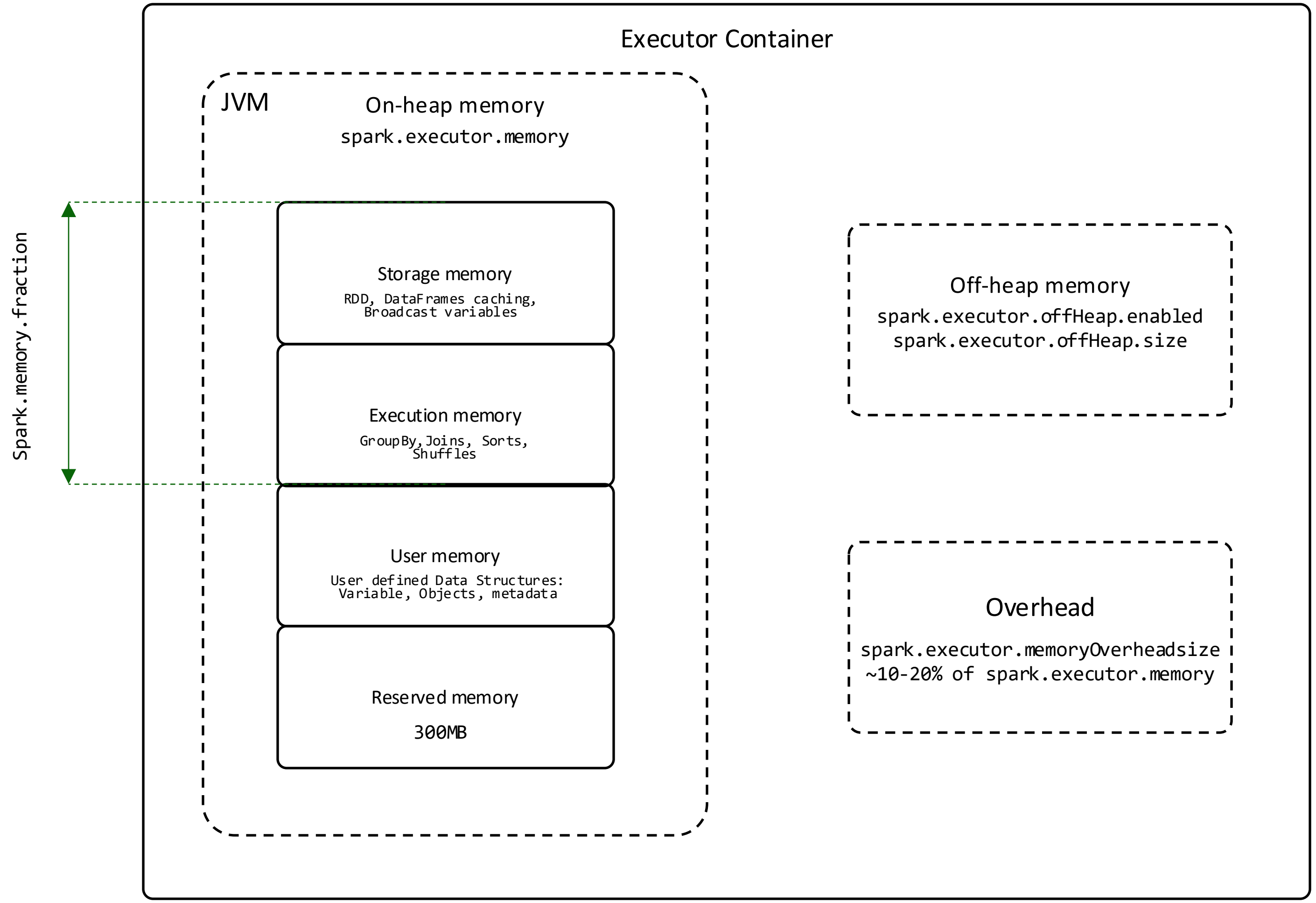
Shuffling and partitioning



🟩 Node

🟧 Data point

Memory management



Executor Container

JVM

On-heap memory

spark.executor.memory

Spark.memory.fraction

Storage memory

RDD, DataFrames caching, Broadcast variables

Execution memory

GroupBy, Joins, Sorts, Shuffles

User memory

User defined Data Structures: Variable, Objects, metadata

Reserved memory

300MB

Off-heap memory

spark.executor.offHeap.enabled
spark.executor.offHeap.size

Overhead

spark.executor.memoryOverheadsize
~10-20% of spark.executor.memory

Source: Spark’s memory management overview

&