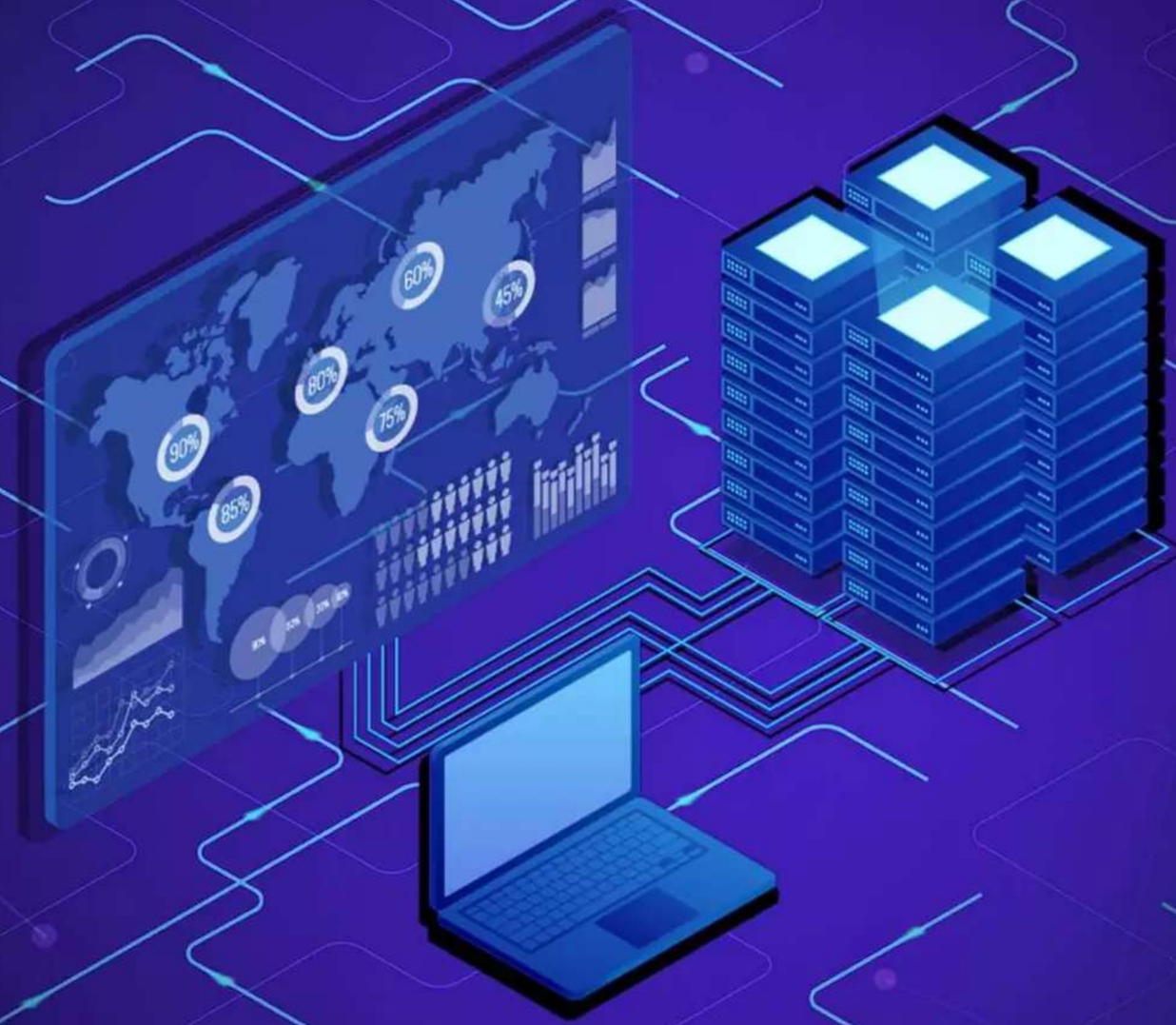
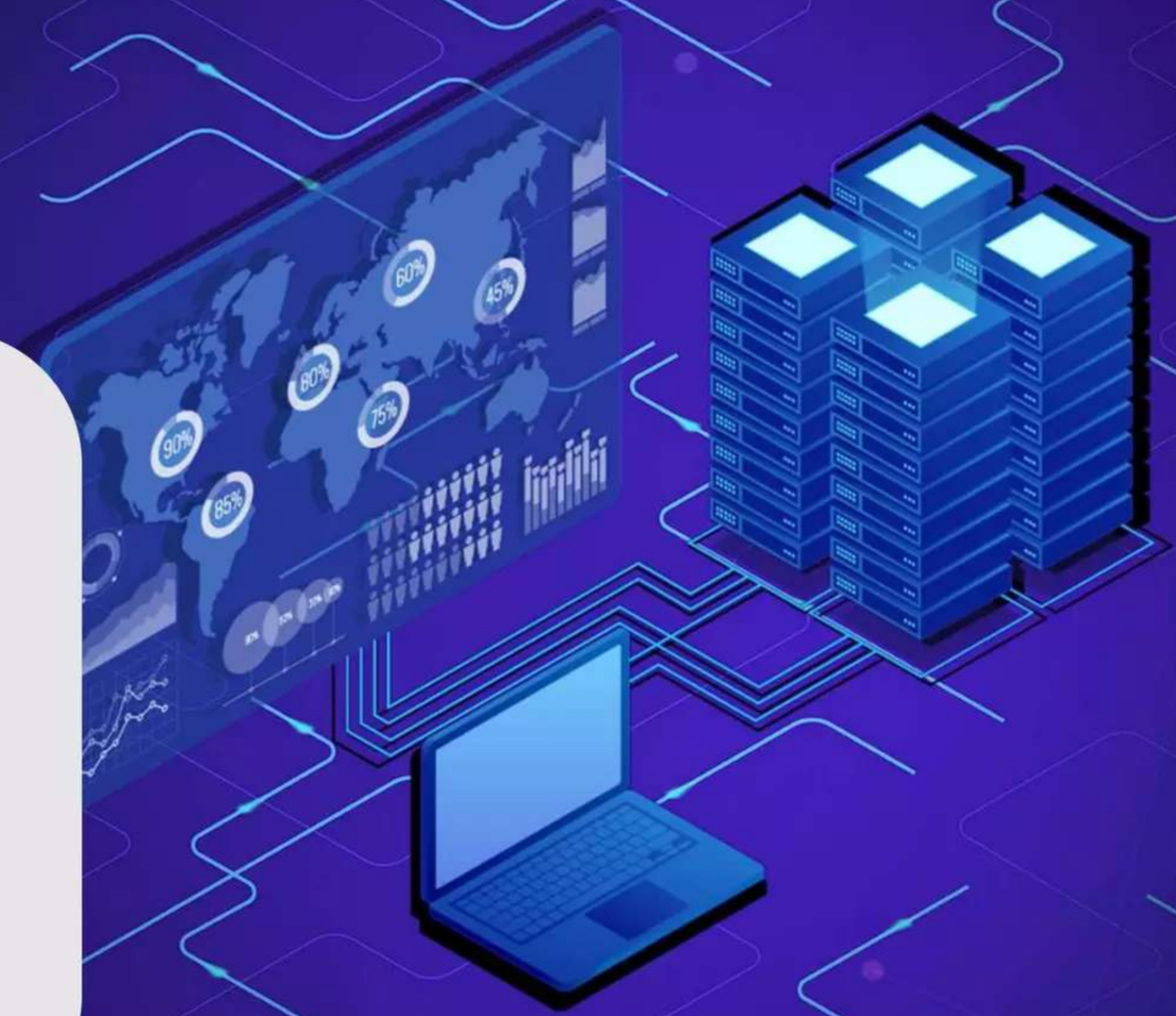
Apache Spark Architecture



simplilearn

What's in it for you?

1. What is Spark?
2. Components of Spark
    - Spark Core
    - Spark SQL
    - Spark Streaming
    - Spark MLlib
    - GraphX
3. Apache Spark Architecture
4. Running a Spark Application



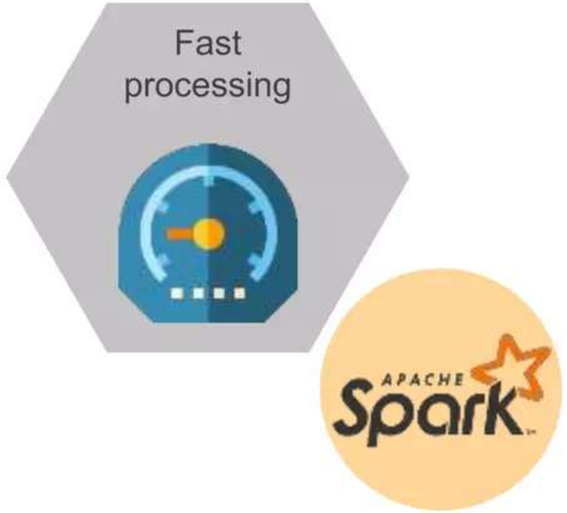
What is Apache Spark?

Apache Spark is a top-level open-source cluster computing framework used for real-time processing and analysis of a large amount of data

What is Apache Spark?

Apache Spark is a top-level open-source cluster computing framework used for real-time processing and analysis of a large amount of data

Spark processes data faster since it saves time in reading and writing operations



What is Apache Spark?

Apache Spark is a top-level open-source cluster computing framework used for real-time processing and analysis of a large amount of data

Spark processes data faster since it saves time in reading and writing operations

Fast processing

Real-time streaming

Spark allows real-time streaming and processing of data

What is Apache Spark?

Apache Spark is a top-level open-source cluster computing framework used for real-time processing and analysis of a large amount of data

Spark processes data faster since it saves time in reading and writing operations

Fast processing

Real-time streaming

Spark allows real-time streaming and processing of data

In-memory computation

Spark has DAG execution engine that provides in-memory computation

simplilearn

What is Apache Spark?

Apache Spark is a top-level open-source cluster computing framework used for real-time processing and analysis of a large amount of data

Spark processes data faster since it saves time in reading and writing operations

Fast processing

Real-time streaming

Spark allows real-time streaming and processing of data

Spark is fault tolerant through RDDs which are designed to handle the failure of any worker node in the cluster

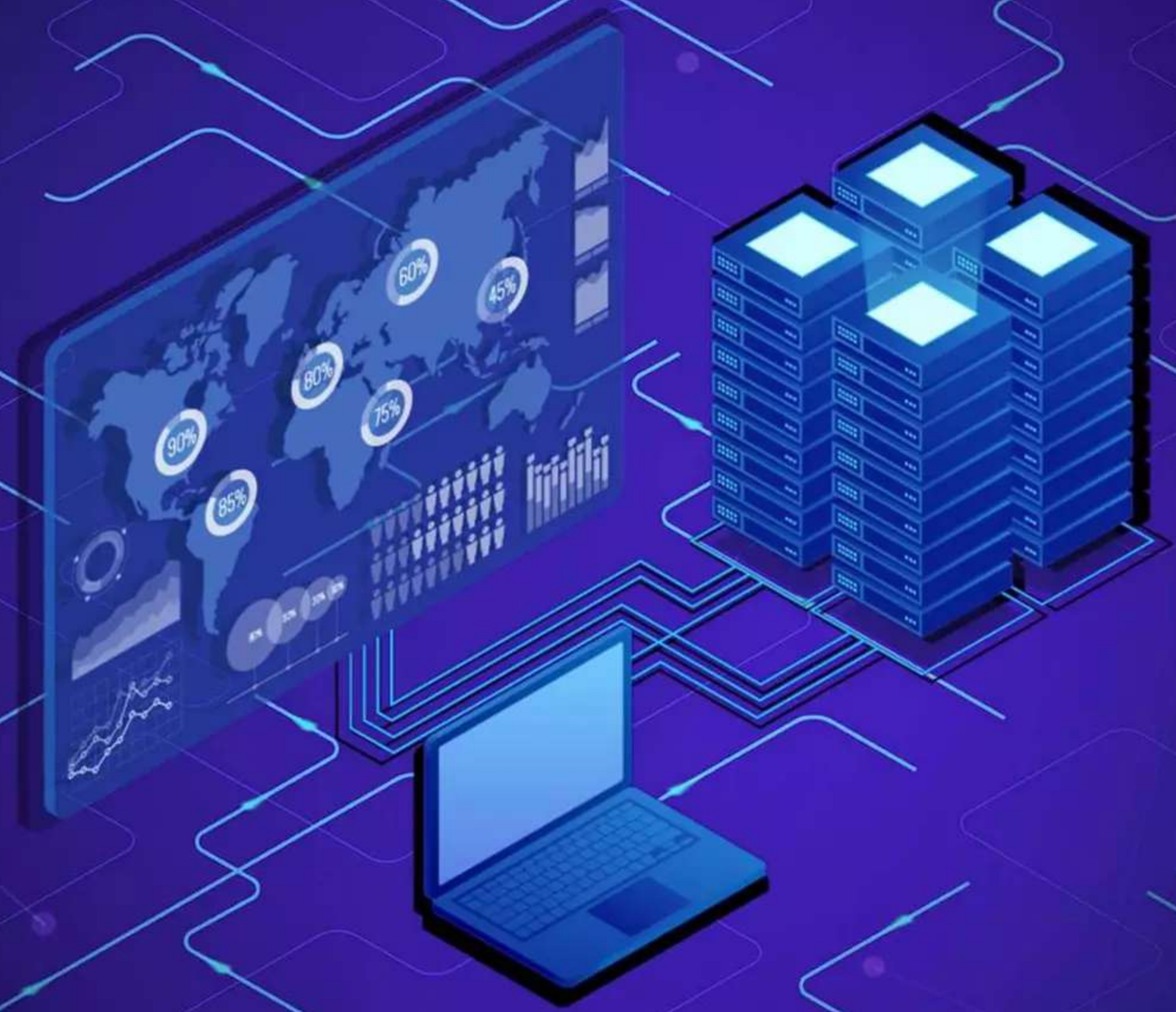
Fault tolerant

In-memory computation

Spark has DAG execution engine that provides in-memory computation

simplilearn

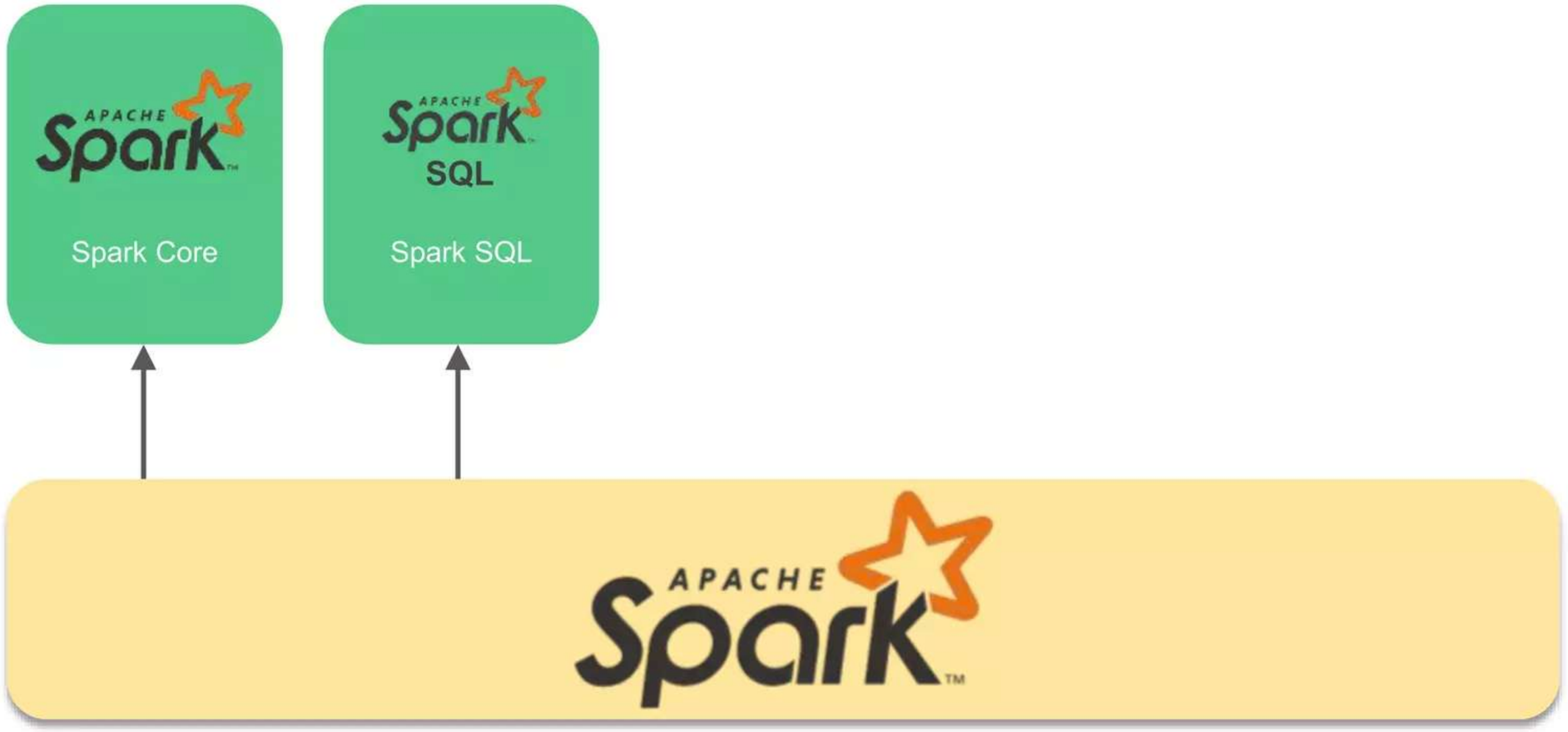
Spark Components



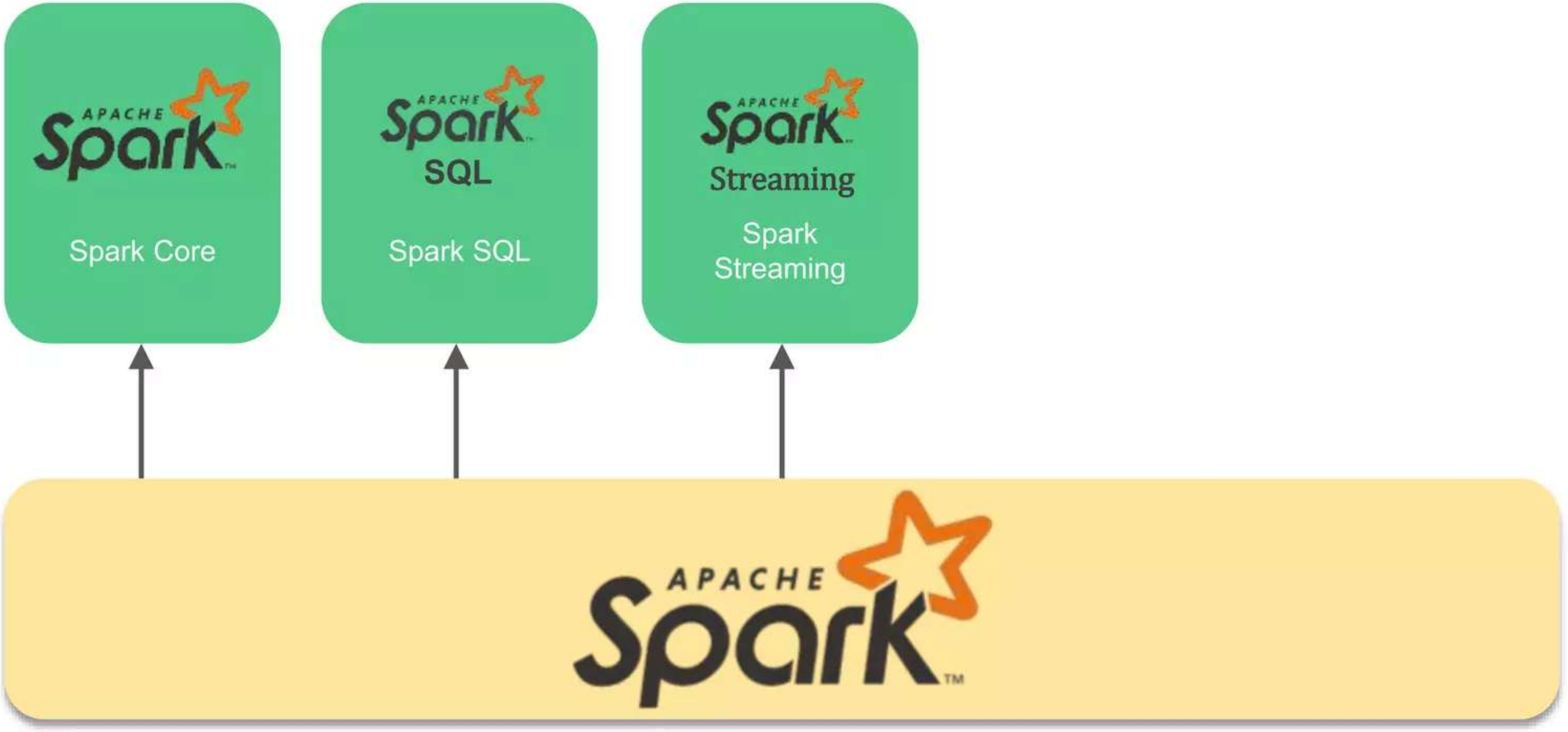
simplilearn

Apache Spark Components

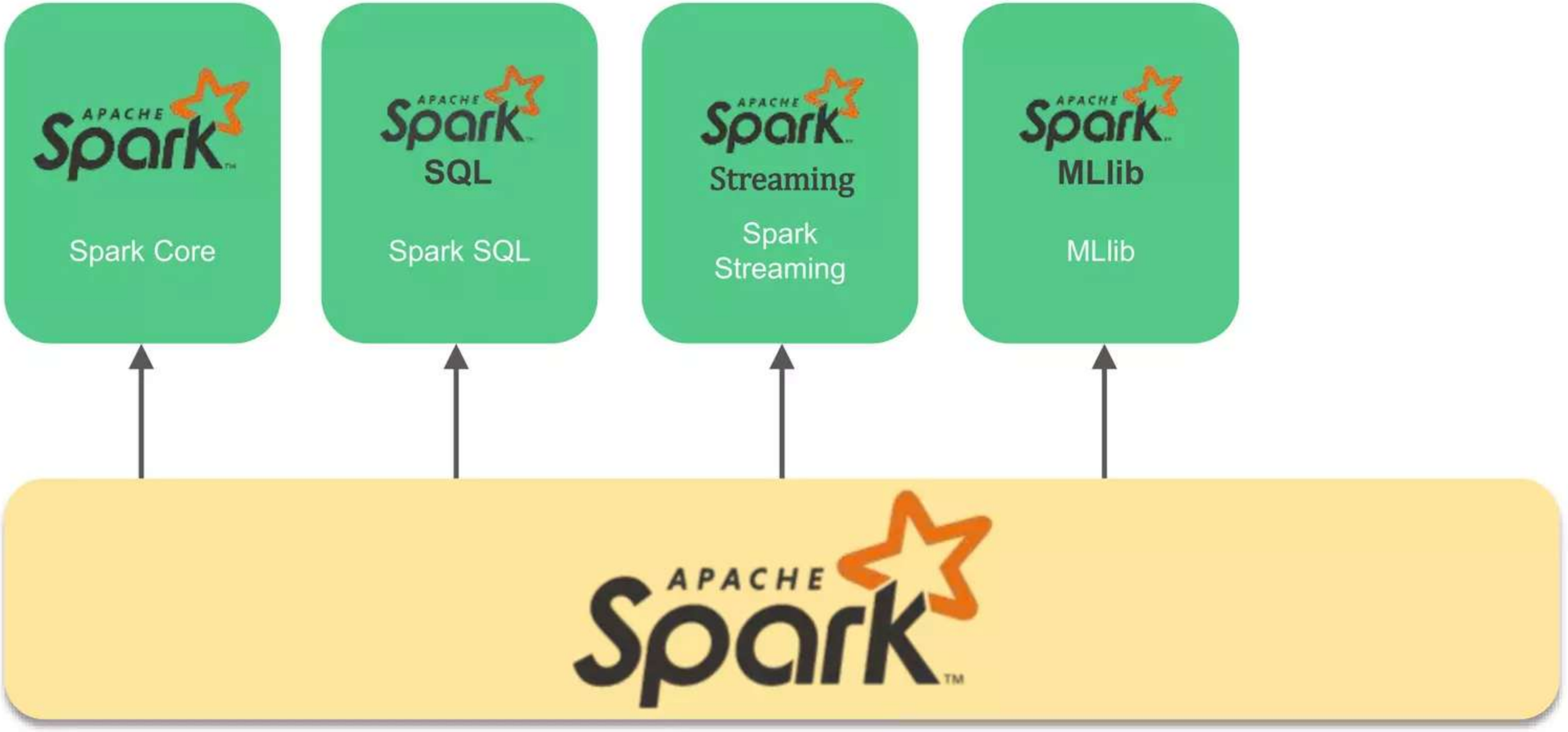
Apache Spark Components



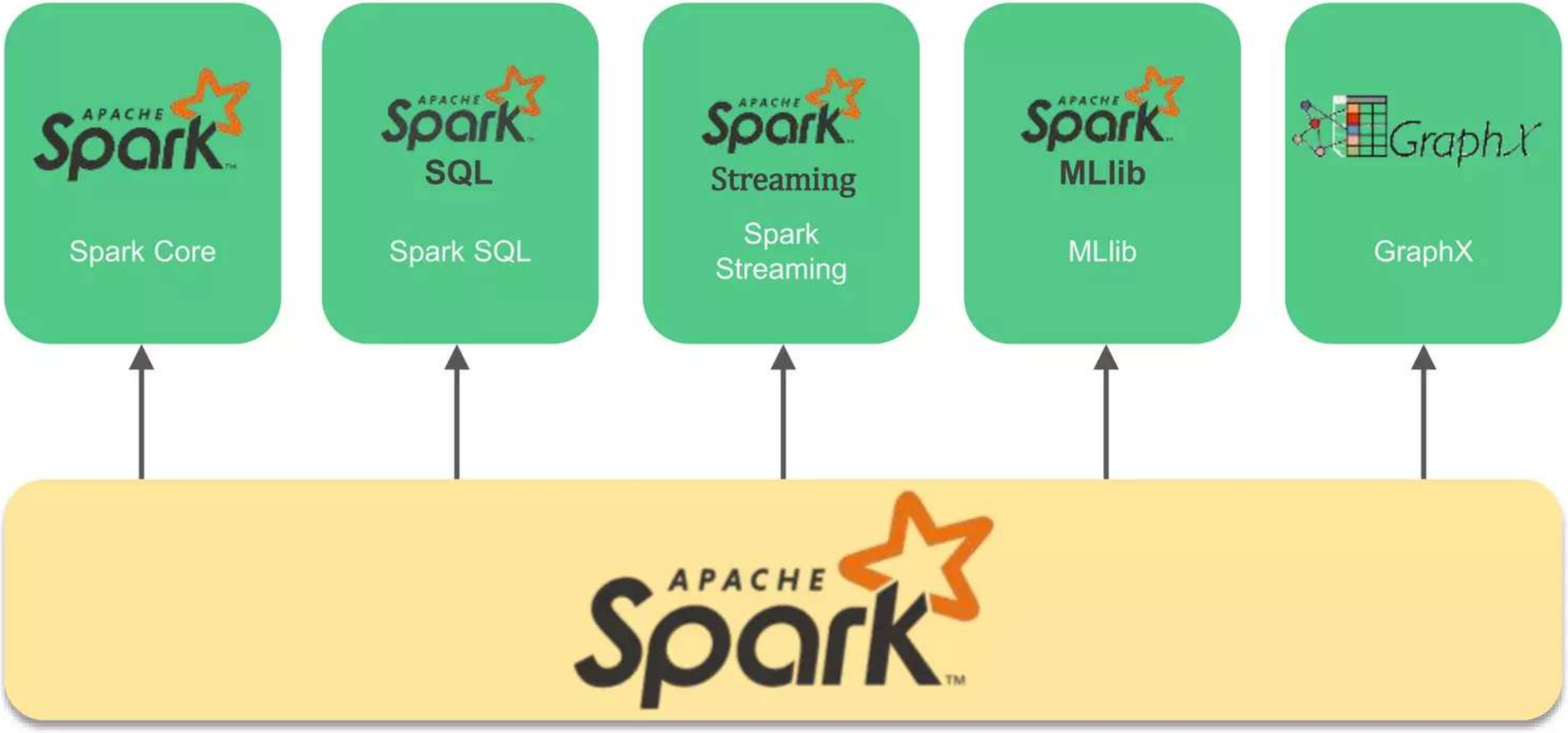
Apache Spark Components



Apache Spark Components



Apache Spark Components



Spark Core

Spark is the core engine for large-scale parallel and distributed data processing

Spark Core

Spark is the core engine for large-scale parallel and distributed data processing

Performs the following:

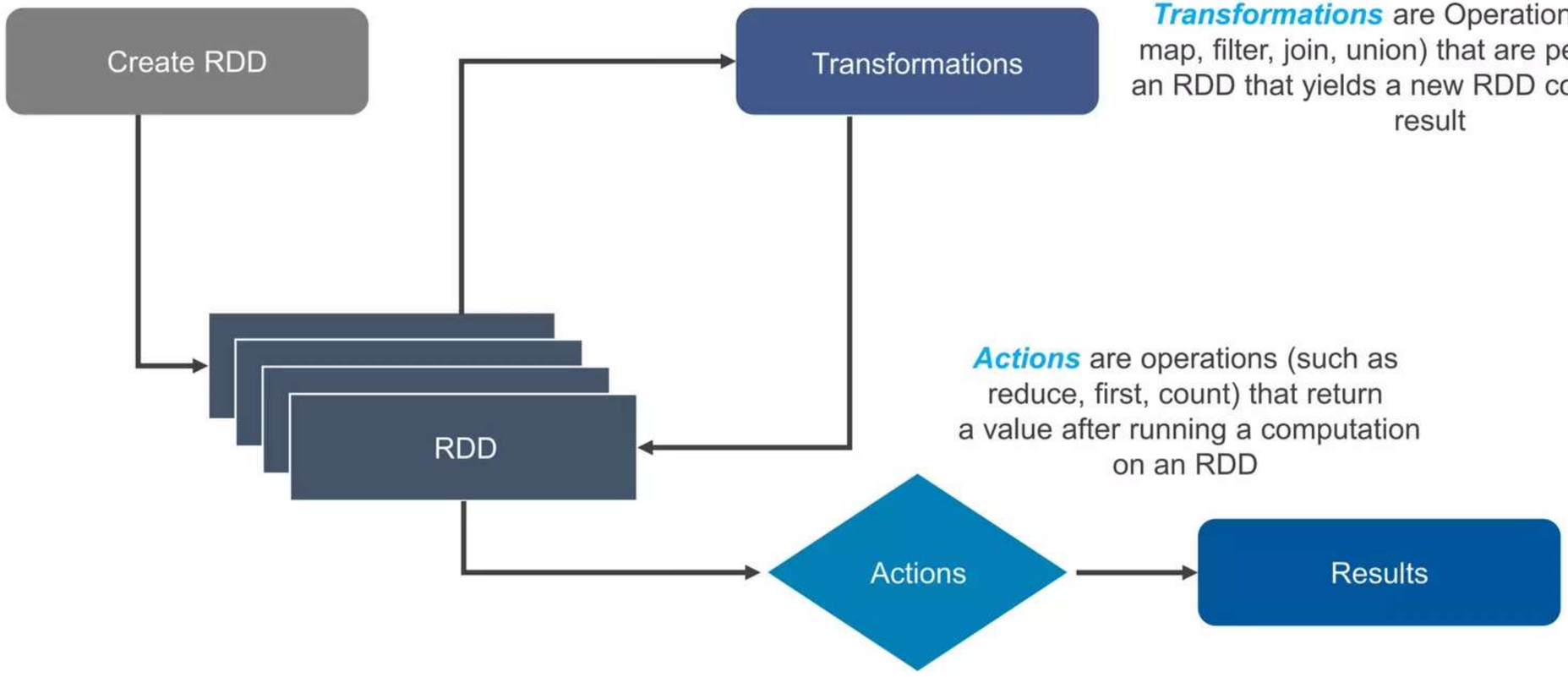
Memory management and fault recovery

Scheduling, distributing and monitoring jobs on a cluster

Interacting with storage system

Spark RDD

***Resilient Distributed Datasets* (RDDs) are the building blocks of any Spark application**



Transformations are Operations (such as map, filter, join, union) that are performed on an RDD that yields a new RDD containing the result

Actions are operations (such as reduce, first, count) that return a value after running a computation on an RDD

simplilearn

Spark SQL

Spark SQL is Apache Spark's module for working with structured data

Spark SQL

Spark SQL is Apache Spark's module for working with structured data

Spark SQL features

Integrated

You can integrate Spark SQL with Spark programs and query structured data inside Spark programs

Spark SQL

Spark SQL is Apache Spark's module for working with structured data

Spark SQL features

Integrated

You can integrate Spark SQL with Spark programs and query structured data inside Spark programs

High Compatibility

You can run unmodified Hive queries on existing warehouses in Spark SQL. With existing Hive data, queries and UDFs, Spark SQL offers full compatibility

simplilearn

Spark SQL

Spark SQL is Apache Spark's module for working with structured data

Spark SQL features

Integrated

You can integrate Spark SQL with Spark programs and query structured data inside Spark programs

High Compatibility

You can run unmodified Hive queries on existing warehouses in Spark SQL. With existing Hive data, queries and UDFs, Spark SQL offers full compatibility

Scalability

Spark SQL leverages RDD model as it supports large jobs and mid-query fault tolerance. Moreover, for both interactive and long queries, it uses the same engine

Spark SQL

Spark SQL is Apache Spark's module for working with structured data

Spark SQL features

Integrated

You can integrate Spark SQL with Spark programs and query structured data inside Spark programs

High Compatibility

You can run unmodified Hive queries on existing warehouses in Spark SQL. With existing Hive data, queries and UDFs, Spark SQL offers full compatibility

Scalability

Spark SQL leverages RDD model as it supports large jobs and mid-query fault tolerance. Moreover, for both interactive and long queries, it uses the same engine

Standard Connectivity

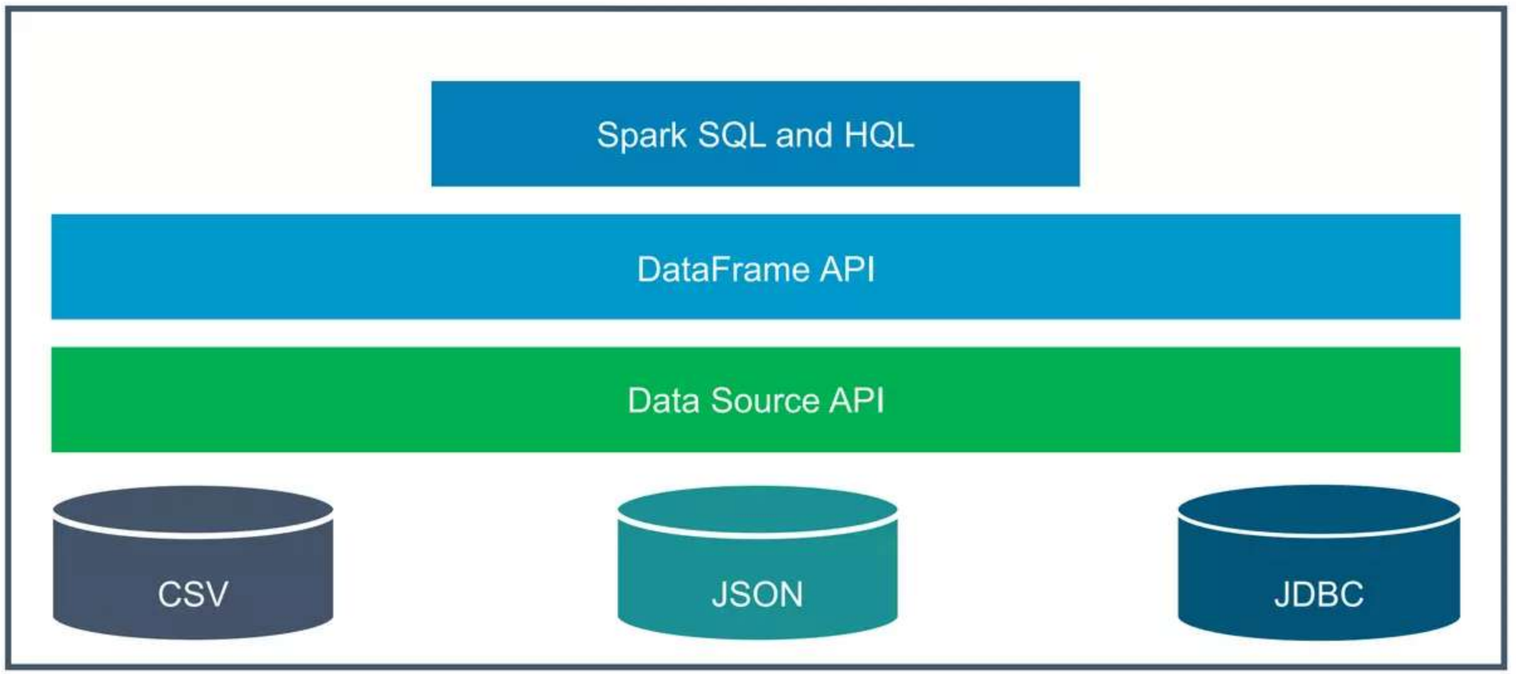
You can easily connect Spark SQL with JDBC or ODBC. For connectivity for business intelligence tools, both turned as industry norms

simplilearn

Spark SQL

Spark SQL is Apache Spark's module for working with structured data

SQL Architecture



Spark SQL

Spark SQL is Apache Spark's module for working with structured data

Spark SQL has three main layers

Language API	SchemaRDD	Data Sources
Spark is compatible and even supported by the languages like Python, HiveQL, Scala, and Java	As Spark SQL works on schema, tables, and records, you can use SchemaRDD or data frame as a temporary table	Data sources for Spark SQL are different like JSON document, HIVE tables, and Cassandra database

simplilearn

Spark SQL

Spark allows you to define custom SQL functions called **User Defined Functions** (UDFs)

UDF that removes all the whitespace and lowercases all the characters in a string

```
def lowerRemoveAllWhiteSpaces(s: String): String = {
    s.tolowerCase().replace("\\S", "")
}

val lowerRemoveAllWhiteSpacesUDF = udf[String, String]
(lowerRemoveAllWhiteSpaces)

val sourceDF = spark.createDF(
   List(
     (" WELCOME    ")
     (" SpaRk SqL    ")
  ), List(
  ("text", StringType, true)
  )
)

sourceDF.select(
  lowerRemoveAllWhiteSpacesUDF(col("text")).as("clean_text")
).show()
```

Output

clean_text
welcome
sparksql

Spark Streaming

Spark Streaming an extension of the core Spark API that enables scalable, high-throughput, fault-tolerant stream processing of live data streams

Spark Streaming

Spark Streaming an extension of the core Spark API that enables scalable, high-throughput, fault-tolerant stream processing of live data streams

Data can be ingested from many sources and the processed data can be pushed out to different filesystems

Spark Streaming

Spark Streaming an extension of the core Spark API that enables scalable, high-throughput, fault-tolerant stream processing of live data streams

Data can be ingested from many sources and the processed data can be pushed out to different filesystems

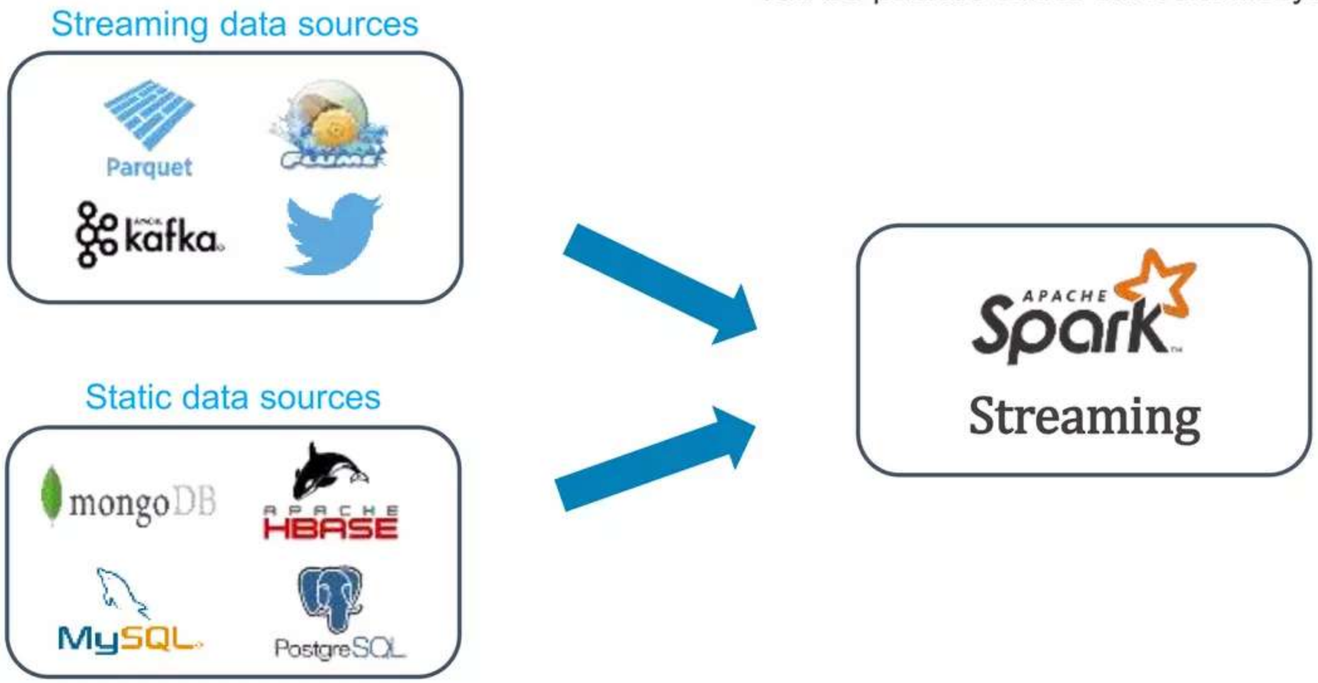
Streaming data sources

Static data sources

Spark Streaming

Spark Streaming an extension of the core Spark API that enables scalable, high-throughput, fault-tolerant stream processing of live data streams

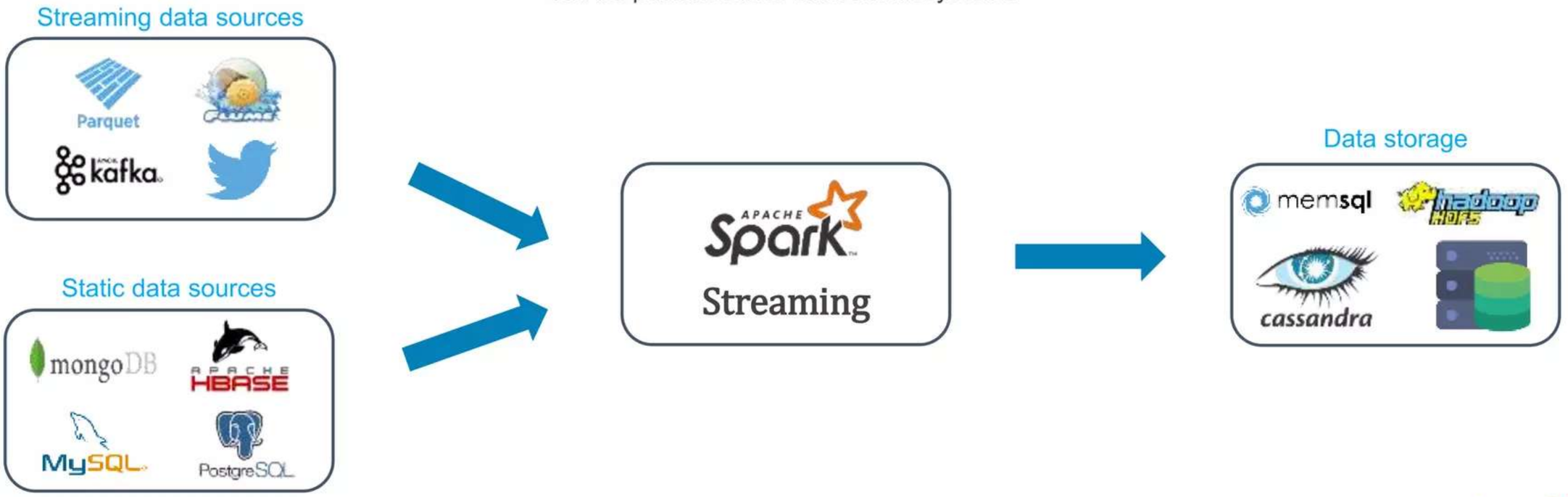
Data can be ingested from many sources and the processed data can be pushed out to different filesystems



Spark Streaming

Spark Streaming an extension of the core Spark API that enables scalable, high-throughput, fault-tolerant stream processing of live data streams

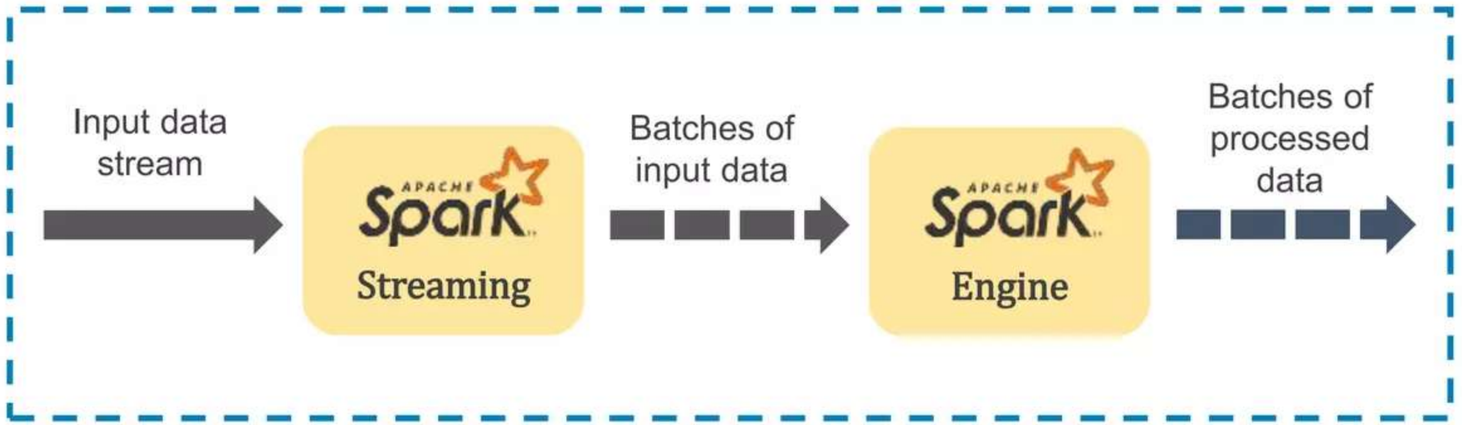
Data can be ingested from many sources and the processed data can be pushed out to different filesystems



Spark Streaming

Spark Streaming an extension of the core Spark API that enables scalable, high-throughput, fault-tolerant stream processing of live data streams

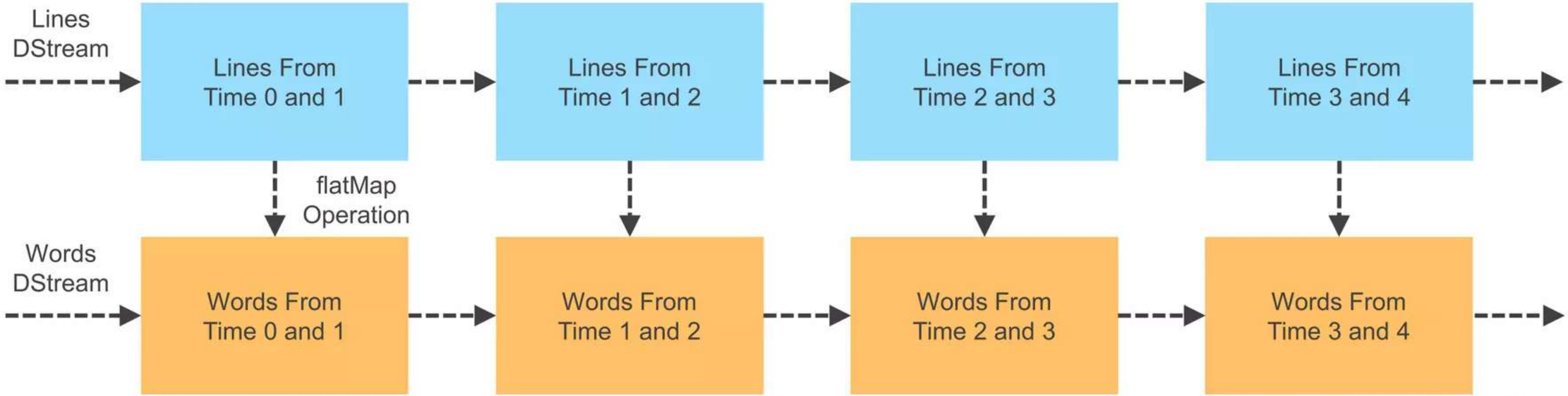
Spark Streaming receives live input data streams and divides the data into batches, which are then processed by the Spark engine to generate the final stream of results in batches



Spark Streaming

Spark Streaming an extension of the core Spark API that enables scalable, high-throughput, fault-tolerant stream processing of live data streams

Here is an example of a basic RDD operation to extract individual words from lines of text in an input data stream



Spark MLlib

MLlib

MLlib is Spark's machine learning library. Its goal is to make practical machine learning scalable and easy

Spark MLlib

MLlib

MLlib is Spark's machine learning library. Its goal is to make practical machine learning scalable and easy

At a high level, it provides the following:

ML Algorithms: classification, regression, clustering, and collaborative filtering

Spark MLlib

MLlib

MLlib is Spark's machine learning library. Its goal is to make practical machine learning scalable and easy

At a high level, it provides the following:

ML Algorithms: classification, regression, clustering, and collaborative filtering

Featurization: feature extraction, transformation, dimensionality reduction, and selection

Spark MLlib

MLlib

MLlib is Spark's machine learning library. Its goal is to make practical machine learning scalable and easy

At a high level, it provides the following:

- ML Algorithms: classification, regression, clustering, and collaborative filtering
- Featurization: feature extraction, transformation, dimensionality reduction, and selection
- Pipelines: tools for constructing, evaluating, and tuning ML pipelines

simplilearn

Spark MLlib

MLlib

MLlib is Spark's machine learning library. Its goal is to make practical machine learning scalable and easy

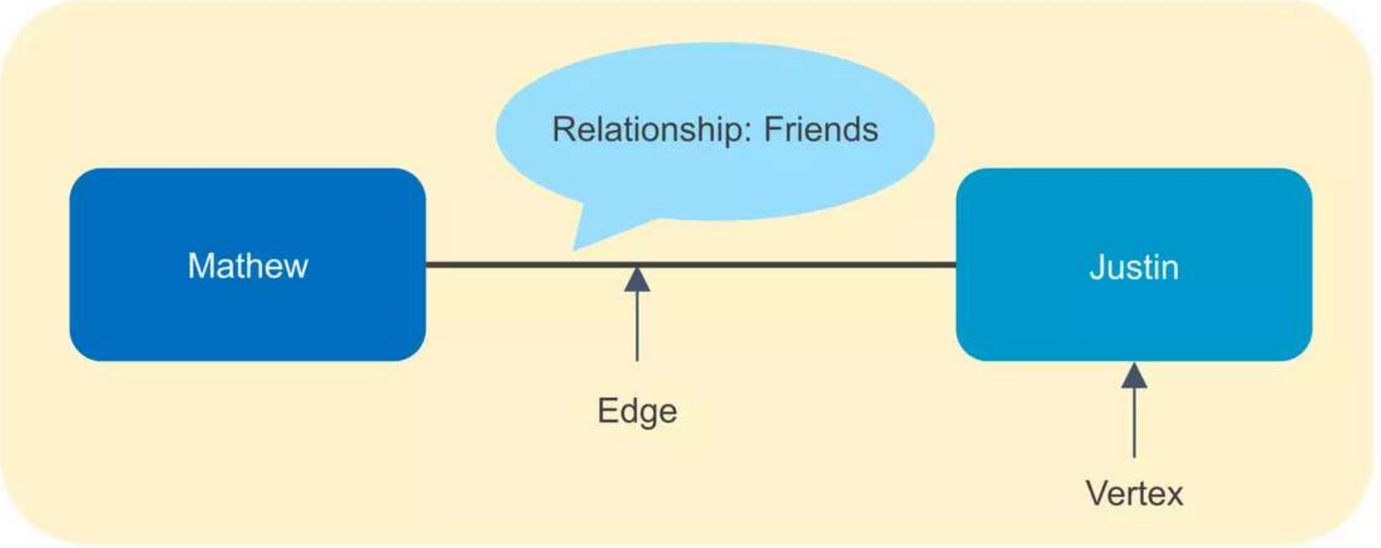
At a high level, it provides the following:

- ML Algorithms: classification, regression, clustering, and collaborative filtering
- Featurization: feature extraction, transformation, dimensionality reduction, and selection
- Pipelines: tools for constructing, evaluating, and tuning ML pipelines
- Utilities: linear algebra, statistics, data handling

GraphX

GraphX is a component in Spark for graphs and graph-parallel computation

GraphX is used to model relations between objects. A graph has vertices (objects) and edges (relationships).



GraphX

GraphX is a component in Spark for graphs and graph-parallel computation

Provides a uniform tool for ETL

Exploratory data analysis

Interactive graph computations

GraphX

GraphX is a component in Spark for graphs and graph-parallel computation

Following are the applications of GraphX

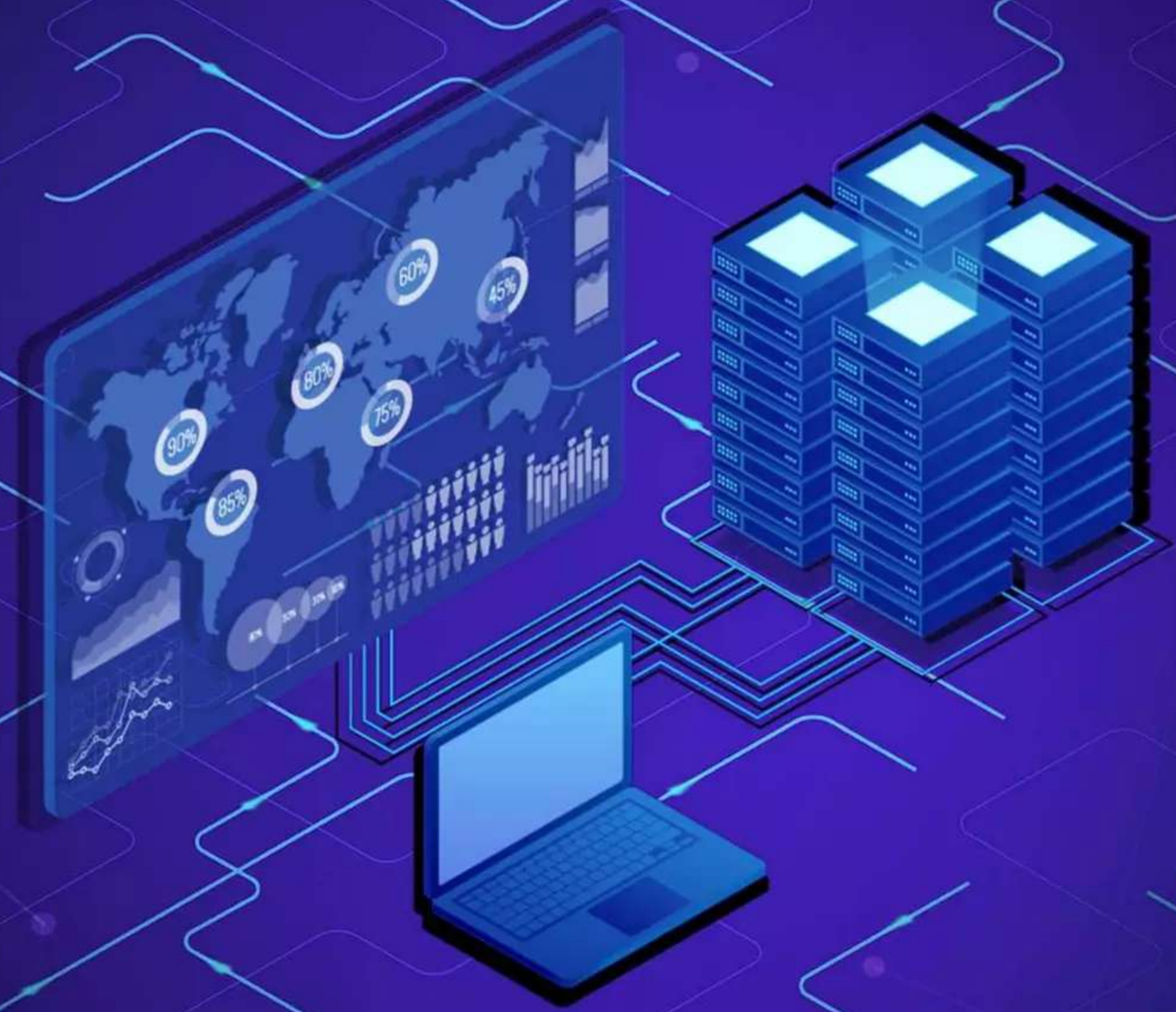
Page Rank

Fraud Detection

Geographic information system

Disaster management

Spark Architecture



simplilearn

Spark Architecture

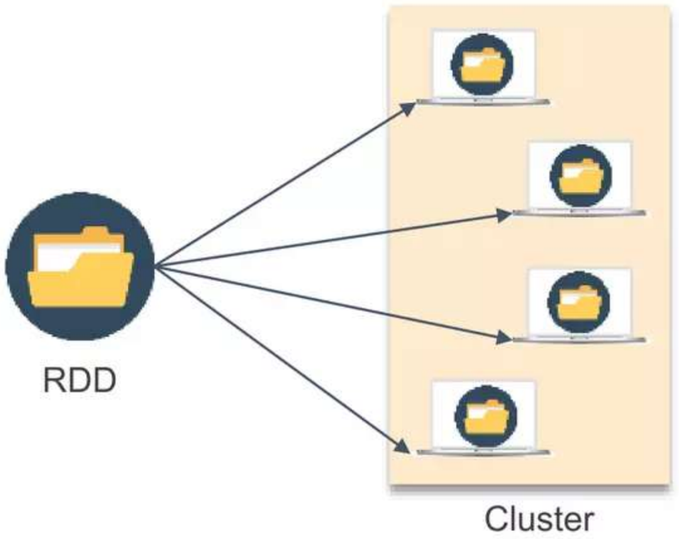
Spark Architecture is based on 2 important abstractions

Spark Architecture

Spark Architecture is based on 2 important abstractions

Resilient Distributed Dataset (RDD)

RDD's are the fundamental units of data in Apache Spark that are split into partitions and can be executed on different nodes of a cluster



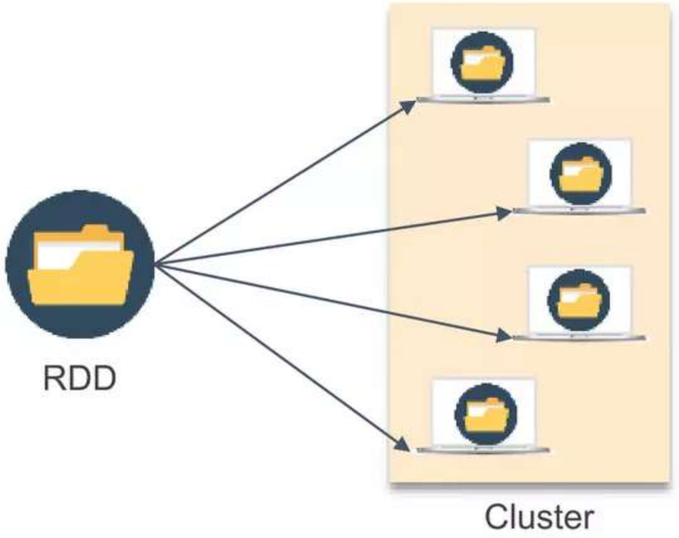
simplilearn

Spark Architecture

Spark Architecture is based on 2 important abstractions

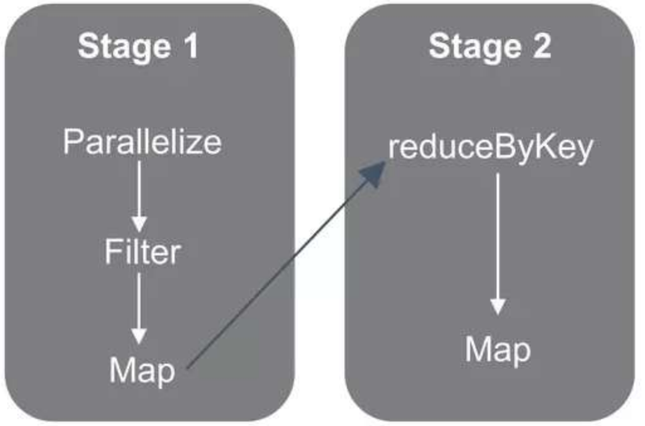
Resilient Distributed Dataset (RDD)

RDD's are the fundamental units of data in Apache Spark that are split into partitions and can be executed on different nodes of a cluster



Directed Acyclic Graph (DAG)

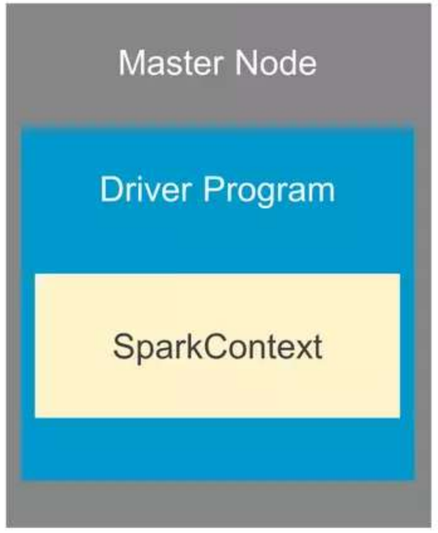
DAG is the scheduling layer of the Spark Architecture that implements stage-oriented scheduling and eliminates the Hadoop MapReduce multistage execution model



simplilearn

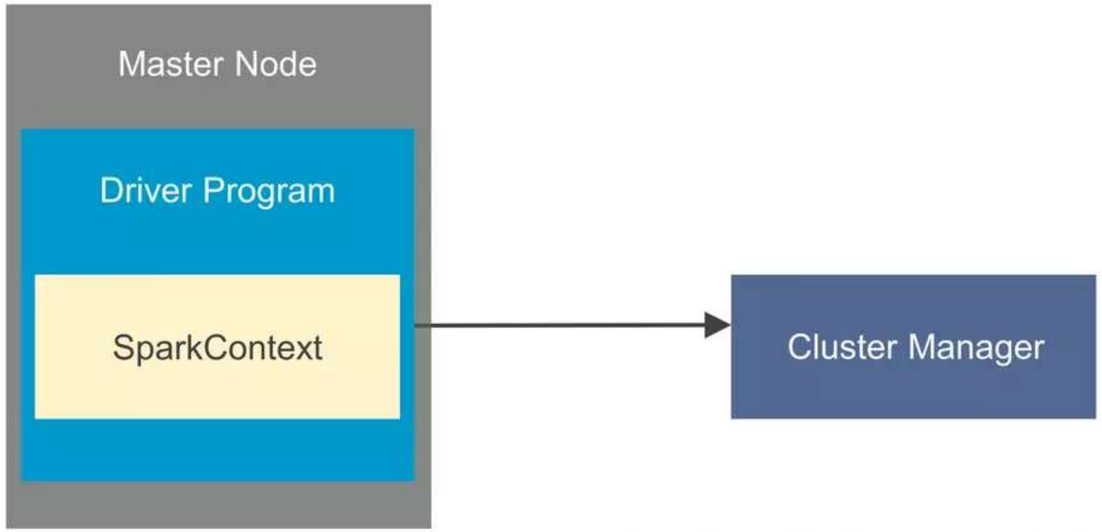
Spark Architecture

Apache Spark uses a master-slave architecture that consists of a driver, that runs on a master node, and multiple executors which run across the worker nodes in the cluster



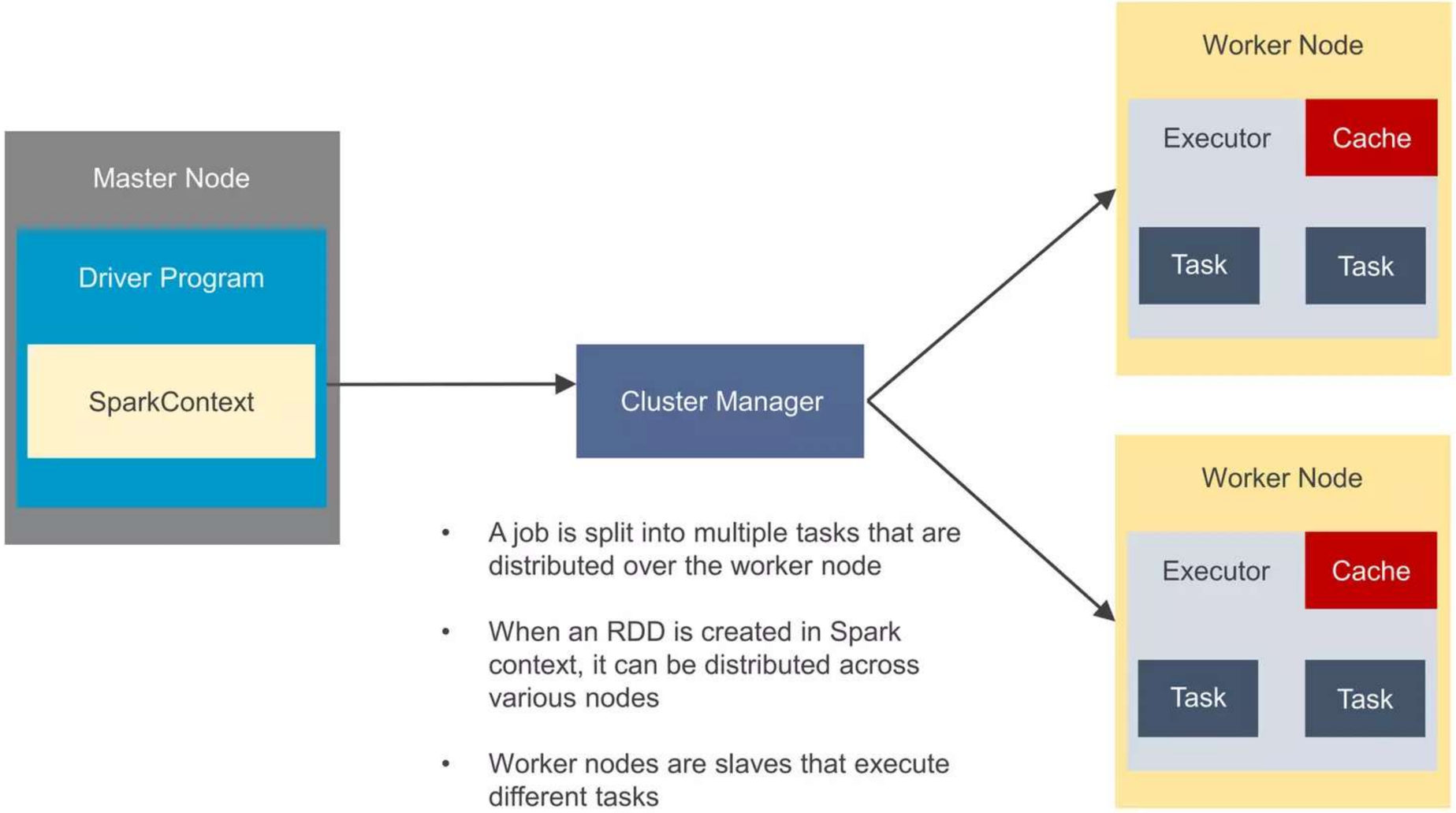
- Master Node has a Driver Program
- The Spark code behaves as a driver program and creates a SparkContext which is a gateway to all the Spark functionalities

Spark Architecture



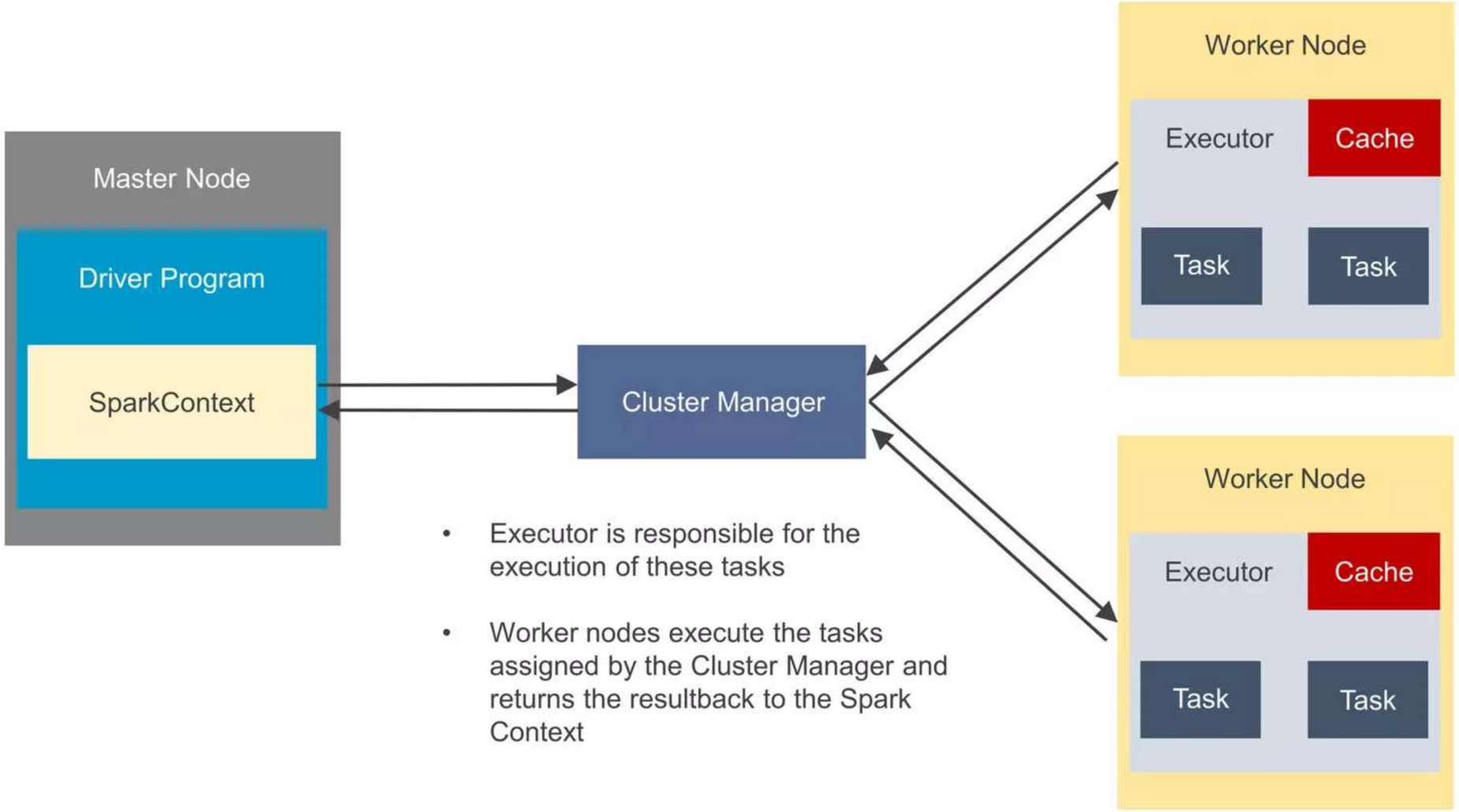
- Spark applications run as independent sets of processes on a cluster
- The driver program & Spark context takes care of the job execution within the cluster

Spark Architecture



- A job is split into multiple tasks that are distributed over the worker node
- When an RDD is created in Spark context, it can be distributed across various nodes
- Worker nodes are slaves that execute different tasks

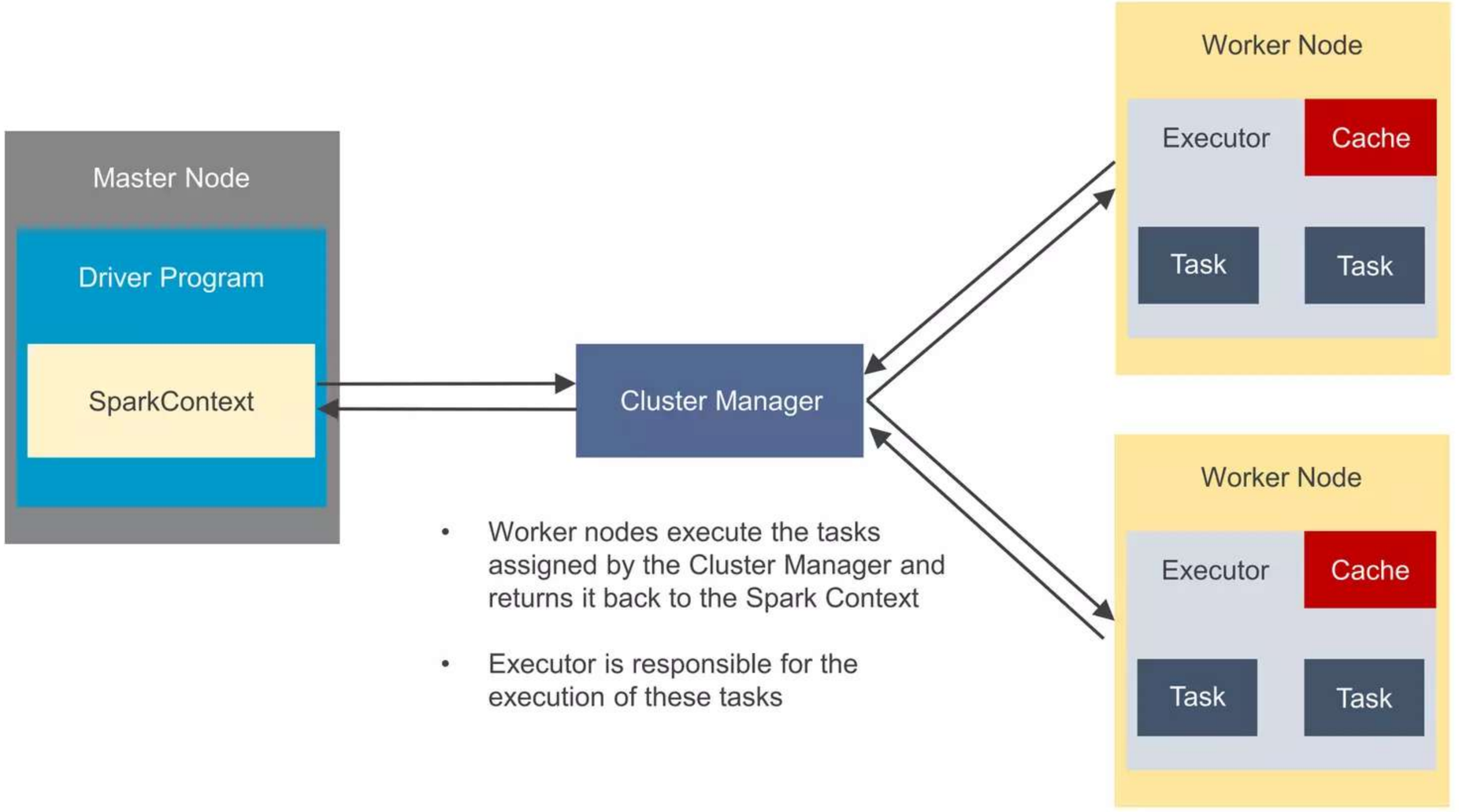
Spark Architecture



- Executor is responsible for the execution of these tasks
- Worker nodes execute the tasks assigned by the Cluster Manager and returns the resultback to the Spark Context

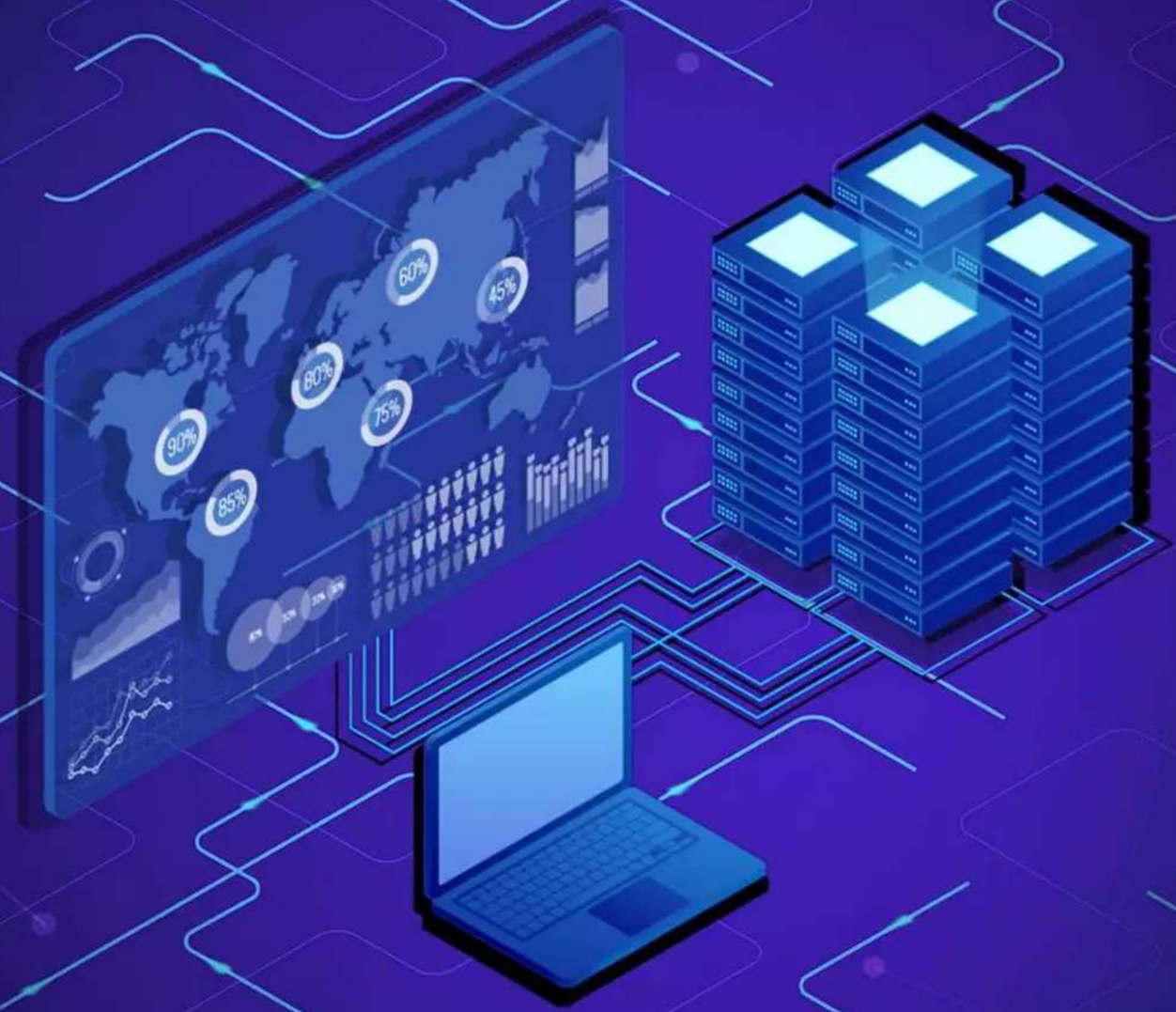
simplilearn

Spark Architecture



- Worker nodes execute the tasks assigned by the Cluster Manager and returns it back to the Spark Context
- Executor is responsible for the execution of these tasks

Running a Spark Application

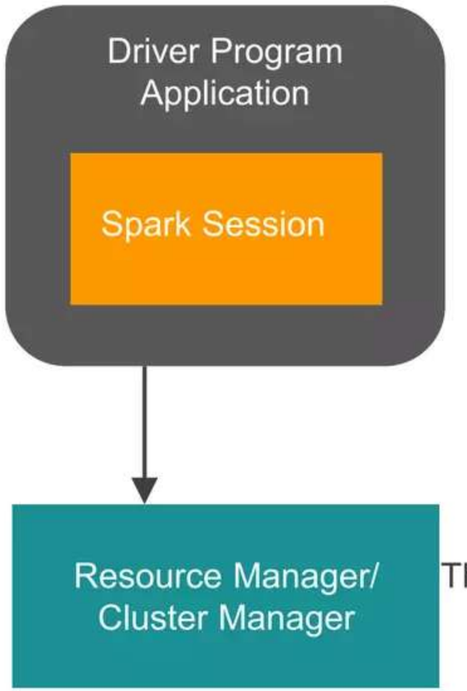


simplilearn

How a Spark application runs on a cluster?

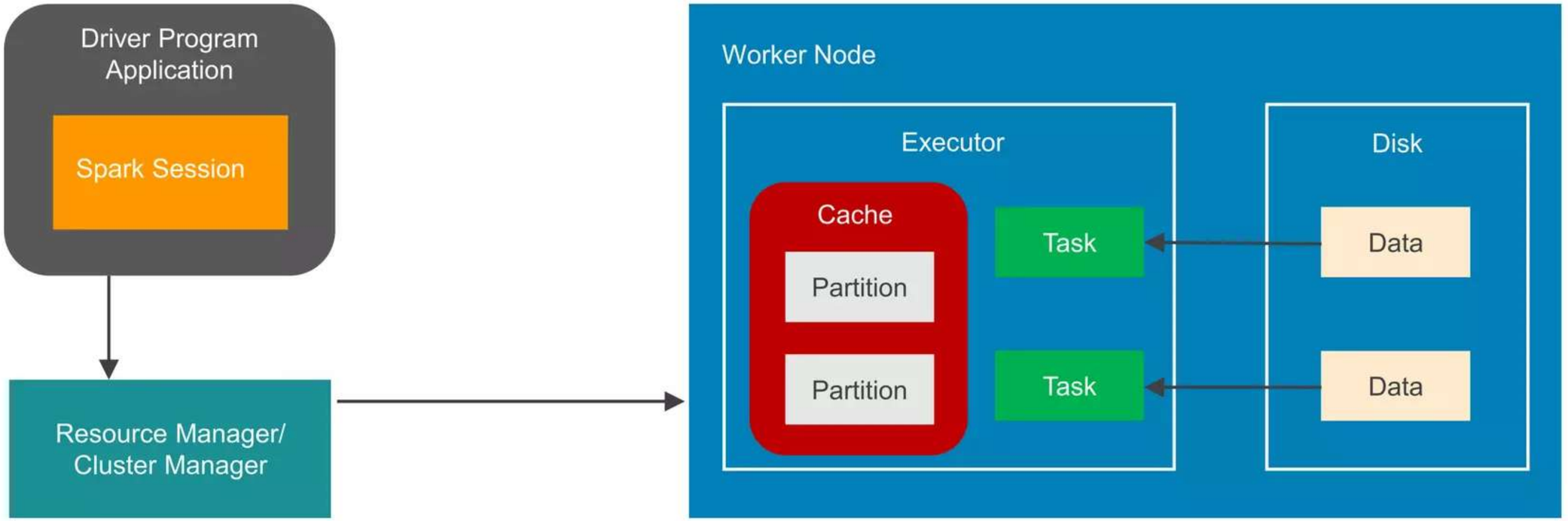
Spark applications run as independent processes, coordinated by the SparkSession object in the driver program

How a Spark application runs on a cluster?



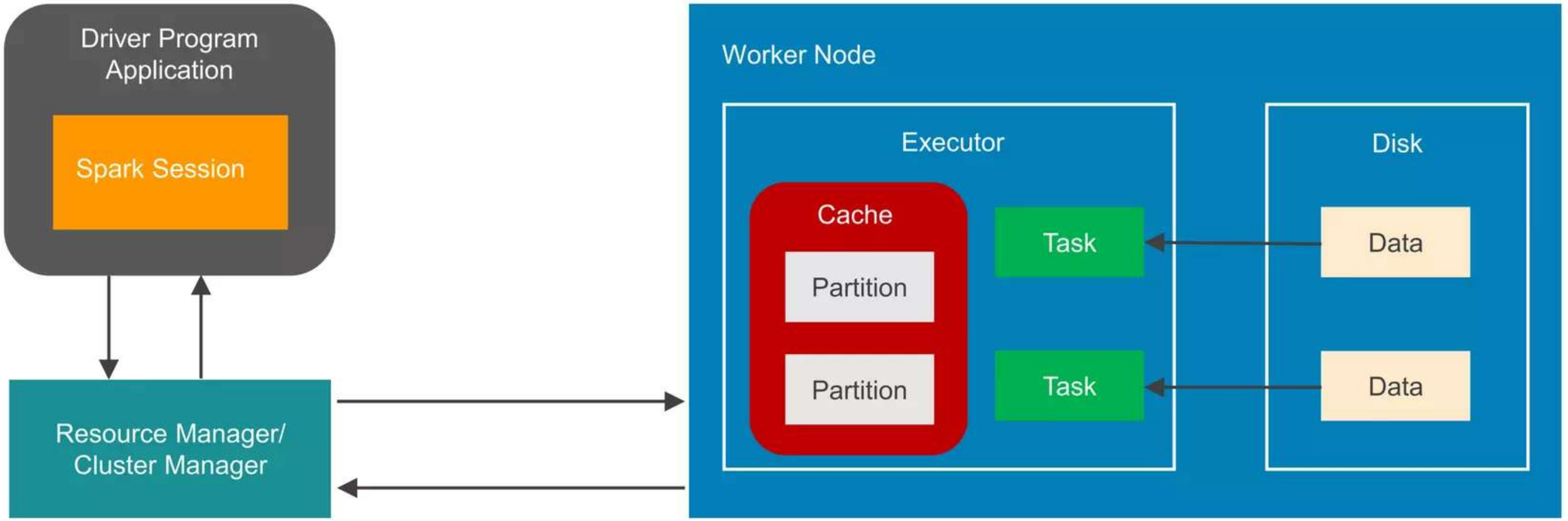
The resource or cluster manager assigns tasks to workers, one task per partition

How a Spark application runs on a cluster?



- A task applies its unit of work to the dataset in its partition and outputs a new partition dataset
- Because iterative algorithms apply operations repeatedly to data, they benefit from caching datasets across iterations

How a Spark application runs on a cluster?



Results are sent back to the driver application or can be saved to disk


THANK YOU
For more information, visit
www.simplilearn.com
simplilearn