POLITECNICO
MILANO 1863

# Apache Spark

Andrea Tocchetti
andrea.tocchetti@polimi.it

# Spark - Introduction


**Apache Spark** is a multi-language engine for executing data engineering, data science, and machine learning on single-node machines or clusters.

# Spark - Introduction

**Batch/Streaming Data**

Apache Spark unifies data processing in batches and real-time streaming.

**SQL Analytics**

Apache Spark executes fast, distributed SQL queries for dashboarding and ad-hoc reporting.

**Data Science at Scale**

Apache Spark performs Exploratory Data Analysis (EDA) on petabyte-scale data without having to resort to downsampling.

**SQL Analytics**

Apache Spark trains machine learning algorithms on standalone machines and uses the same code to scale to fault-tolerant clusters of thousands of machines.

**Language Support**

Apache Spark supports using your preferred language (e.g., Python, SQL, Scala, Java, or R)

# Spark - Integrations

# Spark - Architecture

**SparkSQL**

SparkSQL is a module for structured data processing that manages Datasets and DataFrames (i.e., optimised Spark data structures)

**Spark Streaming**

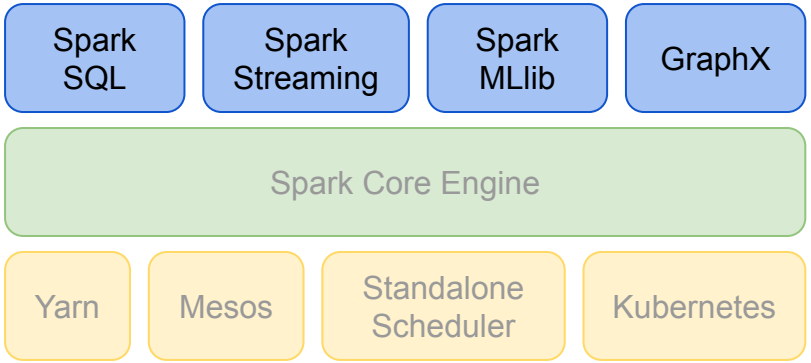
Spark Streaming enables scalable, high-throughput, fault-tolerant stream processing of live data streams.

**Spark MLlib**

Spark MLlib is a high-performance, scalable library for performing Machine Learning tasks.

**GraphX**

GraphX manages graphs and graph-parallel computation.



# Spark - Architecture

**Yarn**

Yarn splits up the functionalities of resource management and job scheduling/monitoring through a single *global Resource Manager* (RM) and multiple *per-application Application Masters* (AM).

**(Apache) Mesos**

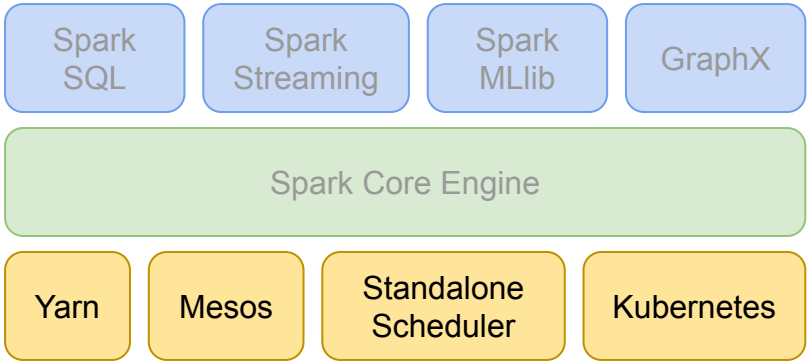
Mesos handles the workload in distributed environment by dynamic resource sharing and isolation while merging multiple physical resources into a single virtual one.

**Standalone Scheduler**

The easiest way to create and manage a cluster on Spark. It works on a single, standalone node/machine.

**Kubernetes**

Kubernetes is an open-source container orchestration system for automating software deployment, scaling, and management. It deploy, maintain, and scale applications based on CPU or memory availability.



# Spark - RDD VS DataFrame

**RDD**

An RDD is a fundamental structure and a fixed collection of data that calculates on a cluster's different nodes. It enables a developer to process computations on large clusters inside the memory in a resilient and efficient manner.

**DataFrame**

A Dataframe is an RDD arranged into named columns. It is a fixed distributed data collection that enables Spark developers to implement a structure on distributed data, allowing high-level abstraction.

**RDD VS DataFrame**

- RDD refers to the ***distributed data elements collection*** across various nodes in the cluster.
- Dataframe refers to the distributed collection of ***organized data in named columns***.


ANY
Questions?