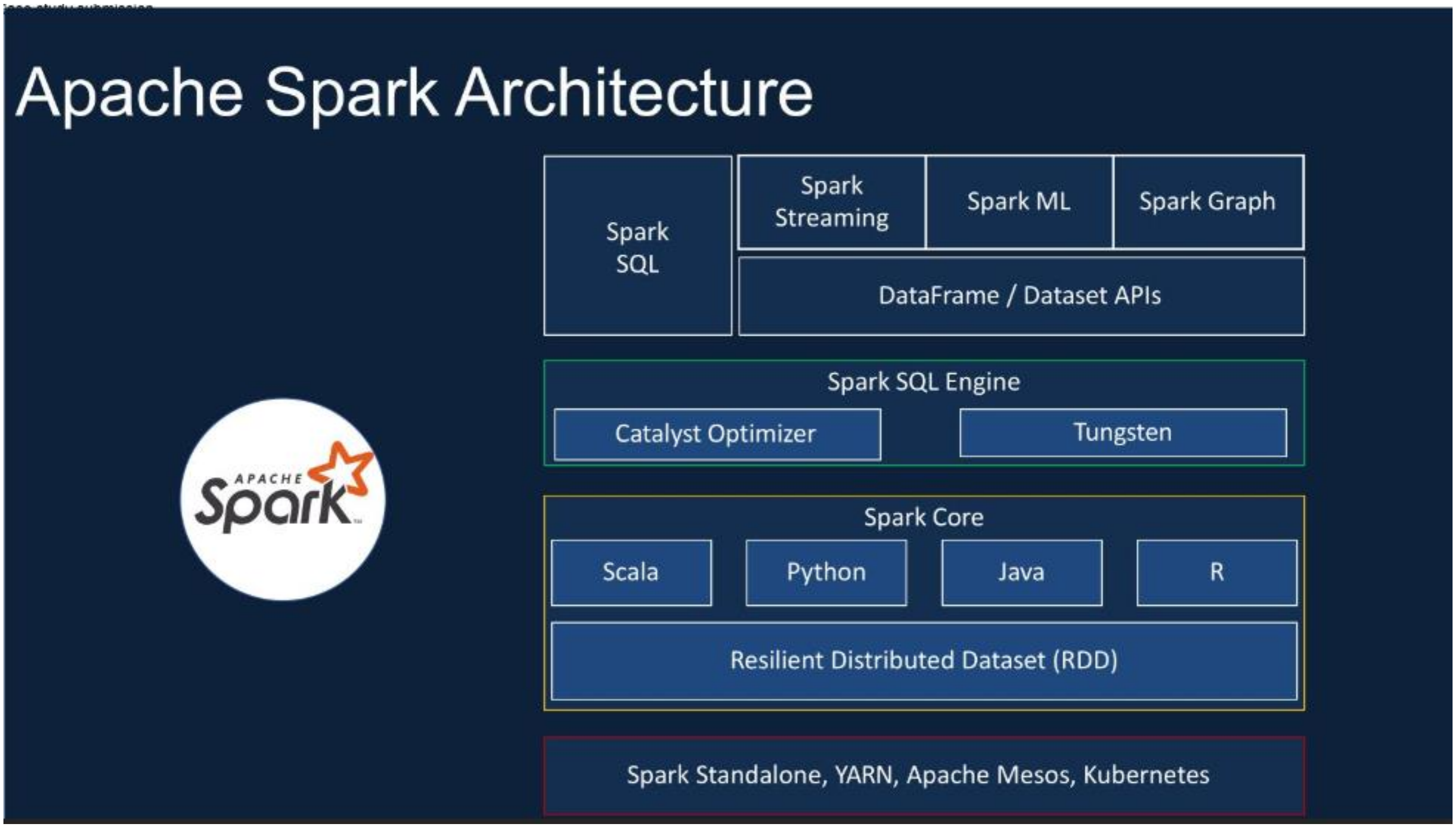
Spark Architecture

Name: Aathirainathan P

Date: 15-11-2024



Apache Spark Architecture

Spark SQL

Spark Streaming

Spark ML

Spark Graph

DataFrame / Dataset APIs

Spark SQL Engine

Catalyst Optimizer

Tungsten

Spark Core

Scala

Python

Java

R

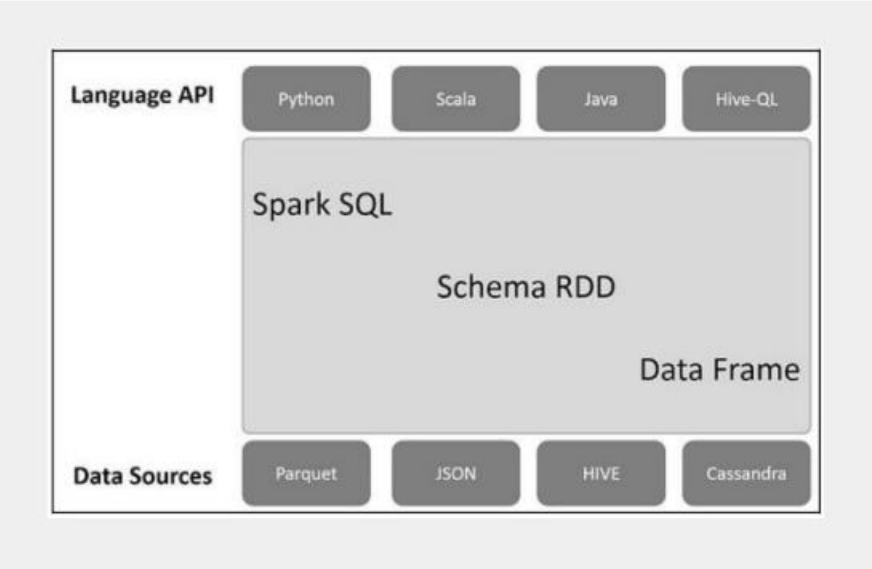
Resilient Distributed Dataset (RDD)

Spark Standalone, YARN, Apache Mesos, Kubernetes

Spark SQL Overview

- Spark SQL is a module in Apache Spark for **structured data processing**.
- Introduced in **Spark 1.0** (May 2014), initially developed by **Michael Armbrust and Reynold Xin** from Databricks.
- It adds a **programming abstraction** called DataFrame and serves as a **distributed SQL query engine**.

Spark SQL Architecture:



Language API

- Spark SQL supports **multiple programming languages**, including **Python, Scala, Java, and HiveQL** (SQL-like query language).
- This compatibility enables developers to work in their preferred language for structured data processing.

Schema RDD

- Spark Core introduces a specialized data structure called **RDD (Resilient Distributed Dataset)**.
- Spark SQL extends RDDs to work with **schemas, tables, and records**, making structured data processing easier.
- **SchemaRDD** can serve as a **temporary table** for executing SQL queries.
- Over time, SchemaRDD evolved into the more advanced **DataFrame** abstraction.

Data Sources

- **Spark Core** typically works with data sources like text files and Avro files.
- **Spark SQL**, however, supports structured data sources such as **Parquet files, JSON documents, Hive tables, and Cassandra databases**.

Key Features of Spark SQL

1. **Integrated**
   - Allows mixing of **SQL queries** with Spark programs seamlessly.
   - Queries structured data as distributed datasets (RDDs) using APIs for **Python, Scala, and Java**.
   - Enables running SQL queries alongside complex analytical algorithms.
2. **Unified Data Access**
   - Supports loading and querying data from diverse sources like **Hive tables, Parquet files, JSON files**, and more.
   - Provides a single interface (SchemaRDD/DataFrame) for structured data processing.
3. **Hive Compatibility**
   - Executes unmodified Hive queries on existing Hive warehouses.
   - Reuses the Hive frontend and MetaStore for compatibility.
   - Can be installed alongside Hive to support Hive queries and User-Defined Functions (UDFs).
4. **Standard Connectivity**
   - Offers industry-standard connectivity through **JDBC and ODBC**.
   - Includes a server mode for external applications to execute SQL queries.
5. **Scalability**
   - Supports interactive and long-running queries using the same engine.
   - Leverages RDD-based fault tolerance for **mid-query recovery** and scalability to handle large datasets.

Spark RDD (Resilient Distributed Dataset)

- Fundamental data structure in Spark, representing **immutable distributed collections** of objects.
- RDDs can store data in **memory or on disk** and are distributed across cluster nodes.
- Key Features:
  - **Partitioned Data**: Divides datasets into partitions for parallel computation.
  - **Parallel Transformations**: Supports operations like map, filter, and more.
  - **Fault Tolerance**: Automatically rebuilds partitions in case of failures.
- RDDs are created by:
  - **Parallelizing collections** in the driver program.
  - **Referencing external data** in storage systems like HDFS, HBase, or shared files.
- Enables faster and more efficient MapReduce operations.

DataFrame and Dataset

- **DataFrame**:
    - A distributed collection of data organized into **named columns**, similar to a relational table.
    - Sources: Hive tables, structured files, external databases, or RDDs.
- **Dataset**:
    - A distributed collection of strongly typed data.

Features of DataFrames

- Handles data sizes ranging from **KBs to PBs** across clusters.
- Supports various data formats (e.g., Avro, CSV, Cassandra) and storage systems (e.g., HDFS, MySQL).
- Uses the **Catalyst Optimizer** for advanced query optimization and code generation.
- Offers APIs in **Python, Scala, Java, and R**.
- Integrates with big data tools and frameworks via Spark Core.

SchemaRDD and Data Sources

- **SchemaRDD**:
    - A specialized RDD that Spark SQL uses for schema-based operations.
    - Also referred to as a **DataFrame**.
- **Data Sources**:
    - Spark Core sources: Text files, Avro files, etc.
    - Spark SQL sources: Parquet files, JSON documents, Hive tables, Cassandra databases, etc.