SPARK EXECUTION PLAN

Physical plan & tungsten optimizer

Spark execution plan levels

1. Logical Plan:

- **Definition**: The Logical Plan represents the high-level abstract syntax tree (AST) of the computation to be performed.
- **Characteristics**:
  - It describes the transformation operations to be applied to the input data.
  - It includes steps like Parsing, Analysis, and Optimization
  - It is a structured representation of the query or DataFrame operations.
- **Optimizations**:
  - Logical optimizations are applied to simplify and optimize the query plan before converting it into a Physical Plan.
- **Example**:
  - A logical plan for a DataFrame operation may include operations simple DataFrame operations and its logical plan like `select`, `filter`, `groupBy`, etc.

2. Physical Plan:

- **Definition**: The Physical Plan specifies the actual physical execution strategy to be used to perform the computation.
- **Characteristics**:
  - It defines the sequence of physical operators and their execution order.
  - It includes details like shuffle operations, join strategies, and data partitioning.
- **Optimizations**:
  - Physical optimizations are applied to optimize the execution plan for efficient data processing.
- **Example**:
  - A physical plan may include operations like `Exchange`, `Sort`, `HashAggregate`, `DataFrame.explain()` etc..

Note: Multiple physical plans may be generated → Spark picks the best cost-based one.

Spark execution plan levels

3. Catalyst Optimizer:

- **Definition**: Catalyst Optimizer is the query optimization framework in Apache Spark that optimizes logical plans and physical plans.
- **Functionality**:
  - It performs rule-based and cost-based optimizations on the logical plan to generate an optimized logical plan.
  - It transforms the logical plan into an optimized physical plan by applying various optimization rules.
  - It uses rules like **constant folding, predicate pushdown**
- **Key Features**:
  - Supports predicate pushdown, constant folding, and other logical optimizations.
  - Generates efficient query plans by analyzing and optimizing the query structure.
- **Role**:
  - Catalyst Optimizer plays a crucial role in optimizing Spark SQL queries by transforming logical plans into efficient physical plans.

4. Tungsten Execution Engine:

- **Definition**: Tungsten Execution Engine is the runtime component of Apache Spark that focuses on improving memory management and CPU efficiency.
- **Functionality**:
  - It includes features like whole-stage code generation, binary processing, and memory caching to enhance performance.
  - It optimizes the execution of generated physical plans by leveraging efficient memory management techniques.
- **Key Features**:
  - Whole-stage code generation compiles multiple operators into a single function for better performance.
  - Binary processing reduces the overhead of object serialization and deserialization.

Spark execution plan levels

Tungsten Optimizer:

•**Definition**: Tungsten Optimizer is a component within the Tungsten Execution Engine that focuses on optimizing the physical execution of Spark jobs.

•**Functionality**:

- It aims to improve memory management and CPU efficiency by generating optimized bytecode for the entire query plan.
- It enhances performance by reducing the overhead of interpreting and executing the code.

•**Key Features**:

- Efficient memory management for processing large datasets.
- Whole-stage code generation for compiling operators into optimized bytecode.

•**Role**:

- Tungsten Optimizer works within the Tungsten Execution Engine to optimize the physical execution of Spark jobs by generating efficient bytecode and managing memory effectively.

Note: Tungsten Execution Engine is responsible for executing the optimized physical plans efficiently by leveraging memory and CPU optimizations.

5. Adaptive Query Execution (AQE):

•**Definition**: Adaptive Query Execution is a feature in Spark that dynamically adjusts the query plan during runtime based on runtime statistics.

•**Characteristics**:

- It allows Spark to adapt the execution plan based on changing data characteristics.
- It can switch between different join algorithms, adjust the number of partitions, and optimize resource allocation.

•**Benefits**:

- Improved performance by adapting to changing data distributions and query patterns.
- Enhanced efficiency by avoiding unnecessary shuffles and optimizing resource utilization.

SUMMARY

1. The Spark Execution Plan starts with the Logical Plan, which is a high-level representation of the computation.
2. The Catalyst Optimizer then optimizes this Logical Plan and transforms it into an optimized Physical Plan.
3. The Tungsten Execution Engine, including the Tungsten Optimizer, focuses on efficient memory management and CPU usage to execute the Physical Plan.
4. Finally, Adaptive Query Execution (AQE) dynamically adjusts the execution plan at runtime based on real-time statistics for further optimization.