ExamsIndex
Practice Materials & Tests

DEMO VERSION

Databricks

Developer-for-Apache-Spark-Scala Exam

Certified Associate Developer for Apache Spark - Scala

Exam Latest Version: 6.0

Question 1. (Multi Select)

Which of the following scenarios would benefit the most from using Adaptive Query Execution (AQE) in Spark?

A: A query that processes a fixed dataset with a well-defined schema and predictable data distribution.
B: A query that performs simple aggregations on a small dataset, where performance is not critical.
C: A query that involves complex joins, aggregations, and data transformations on a large and potentially skewed dataset.
D: A query that reads data from a real-time data stream, where data characteristics are highly dynamic.
E: A query that performs batch processing of data with static characteristics.

Correct Answer: C, D

Explanation:

AQE is most beneficial when dealing with complex queries on large and potentially skewed datasets, as well as queries on dynamic data streams. In these scenarios, AQE can effectively adapt to data characteristics and optimize the execution plan for improved performance. AQE is less impactful on simple queries with fixed datasets, where pre-optimization might already be sufficient.

Question 2. (Single Select)

Which of the following statements accurately describes the role of the Spark Driver in a Spark application?

A: The Driver is responsible for distributing tasks to executors and managing data partitioning.
B: The Driver is responsible for storing the final results of the computation.
C: The Driver is the main process that coordinates the execution of the Spark application,

including submitting tasks to executors, tracking their progress, and aggregating results.
D: The Driver is responsible for reading and writing data to external data sources.
E: The Driver is responsible for managing the memory allocation for executors.

Correct Answer: C

Explanation:

The Spark Driver is the main process that coordinates the execution of the Spark application. It is responsible for submitting tasks to executors, tracking their progress, and aggregating results. The Driver is not responsible for data partitioning, storing final results, reading/writing to external sources, or managing executor memory.

Question 3. (Single Select)

What is the role of the Spark Shuffle Manager in Adaptive Query Execution (AQE)?

A: The Shuffle Manager is responsible for managing the data partitioning and shuffling process, allowing AQE to dynamically adjust these operations based on runtime data characteristics.
B: The Shuffle Manager is responsible for caching frequently accessed data in memory, which can improve AQE's performance.
C: The Shuffle Manager is responsible for validating data transformations during execution, ensuring data integrity for AQE.
D: The Shuffle Manager is responsible for scheduling tasks across different worker nodes, which can be dynamically adjusted by AQE.
E: The Shuffle Manager is responsible for managing the communication between the Driver and the worker nodes, which can be optimized by AQ

Correct Answer: A

Explanation:

The Spark Shuffle Manager plays a crucial role in AQE by managing the data partitioning and shuffling process. This allows AQE to dynamically adjust these operations based on runtime data characteristics. While the Shuffle Manager is important for data management, its role in

AQE is primarily focused on data partitioning and shuffling, not caching, data validation, task scheduling, or communication management.

Question 4. (Single Select)

How does Spark's AQE address data skewness during query execution?

A: AQE identifies skewed data partitions and dynamically adds more executors to handle the increased workload.
B: AQE dynamically adjusts the shuffling strategy to redistribute skewed data across multiple partitions, improving load balancing.
C: AQE automatically filters out skewed data points, reducing the overall data volume and improving performance.
D: AQE applies a pre-processing step to normalize skewed data before the query execution.
E: AQE dynamically increases the memory allocation for executors handling skewed data partitions.

Correct Answer: B

Explanation:

AQE tackles data skewness by dynamically adjusting the shuffling strategy. It identifies skewed partitions and redistributes the skewed data across multiple partitions, improving load balancing and reducing the impact of skewness on query performance. AQE does not automatically add executors, filter out data, apply pre-processing steps, or increase memory allocation as primary methods to address data skewness.

Question 5. (Single Select)

In a scenario where your Spark application is experiencing performance issues due to data skewness, how could you leverage AQE to improve query execution?

A: Enable AQE and use a custom partitioner that distributes data based on the skewness characteristics.
B: Disable AQE and manually optimize the query plan by explicitly partitioning and shuffling the data based on the skewed attributes.
C: Increase the number of executors to handle the skewed data partitions more efficiently.
D: Use a different data serialization format to optimize data exchange between executors.
E: Reduce the number of tasks assigned to executors to prevent overloading them with skewed data.

Correct Answer: A

Explanation:

In a scenario where data skewness is causing performance issues, enabling AQE with a custom partitioner can be a valuable approach. The custom partitioner allows you to distribute data based on the skewness characteristics, guiding AQE to efficiently handle skewed data during execution. Disabling AQE and manual optimization, increasing executors, altering serialization format, or reducing tasks are not the most effective strategies for addressing data skewness in this context.

ExamsIndex

Demo PDF Complete

Your Developer-for-Apache-Spark-Scala Demo (5 Questions)

Get the Complete Version

Full Questions with Detailed Explanations

Interactive Web-Based Exams Available

To get 30% off, use Coupon Code: OFF30

https://examsindex.com/exam/developer-for-apache-spark-scala