Sairam Konar D15C 27 2024-2025

Experiment 10

Aim:

To perform Batch and Streamed Data Analysis using Apache Spark.

Theory:

1. What is Streaming? Explain Batch and Stream Data.

In data processing, we generally deal with two major types: Batch processing and Stream processing.

Batch Data Processing refers to collecting data over a period of time and then processing it all at once. Think of it like baking cookies: you prepare a whole batch and then put it in the oven. It's ideal when real-time insights aren't necessary.

Stream Data Processing (also known as real-time processing) handles continuously flowing data, such as sensor feeds, social media updates, or payment transactions. Instead of waiting for a complete dataset, stream processing processes each piece of data as it arrives.

Streaming is the process of analyzing data in motion. It's especially useful when immediate insights are crucial, such as in fraud detection, server monitoring, or stock market analysis.

2. How Does Data Streaming Take Place Using Apache Spark?

Apache Spark provides a module called Spark Streaming and its newer, more advanced version, Structured Streaming, for processing real-time data streams.

Working:

- **Data Source:** Data is continuously received from sources like Kafka, socket connections, or files being updated in real time.
- **Spark Streaming Engine**: Spark divides incoming data into small batches (micro-batches). Despite being a streaming method, it operates by processing these small batches at regular intervals.
- **Transformations and Actions:** Similar to batch mode, filtering, grouping, or aggregation logic can be applied to each micro-batch.
- **Output Sink**: Results are pushed to dashboards, databases, or alert systems for near-instant insights.

  The power of Spark lies in its unified programming model for both batch and stream processing, making it both flexible and efficient.

Steps for Batch Data Analysis using Apache Spark

- **Start Apache Spark Environment:** Launch Apache Spark locally, on the cloud, or using a notebook interface like Jupyter or Databricks.

- **Read a Batch Dataset:** Load a static dataset (e.g., CSV or JSON) with a fixed structure.
- **Explore the Dataset:** Inspect columns, data types, and sample records.
- **Clean and Prepare the Data:** Handle missing values, rename columns, fix data types, and remove duplicates.
- **Perform Transformations and Aggregations:** Apply filtering, grouping, and compute metrics like averages or totals.
- **Store or Display Output:** Save results to files, visualize them, or print to console.

Steps for Streamed Data Analysis using Apache Spark

- **Initialize Spark with Structured Streaming:** Start a Spark session with structured streaming configuration.
- **Connect to a Streaming Data Source:** Connect to sources like Kafka, socket, or folders receiving new files.
- **Define the Schema for Streaming Data:** Manually specify schema fields like timestamp, value, etc.
- **Apply Streaming Operations:** Perform real-time filtering, windowed grouping, and aggregation.
- **Write Stream Output to a Sink:** Continuously write results to console, files, databases, or dashboards.
- **Monitor the Streaming Pipeline:** Track job status, data throughput, and resource usage during streaming.

Conclusion

This experiment demonstrates the power of Apache Spark in handling both batch and streamed data processing. Batch analysis is ideal for static historical data, while streamed analysis provides real-time insights for live data scenarios. Apache Spark’s unified architecture makes it an efficient and scalable solution for diverse data processing needs.