Name : Divesh Lulla **D15_C** **Roll No: 31**

Experiment No: 10

Aim: To perform Batch and Streamed Data Analysis using Apache Spark.

Theory:

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

Streaming is a method of data processing where data is continuously generated and processed in real-time or near real-time. It allows immediate analysis and response to incoming data, making it ideal for applications like live video feeds, financial transactions, or social media updates. Streaming helps in quick decision-making as the data is analyzed the moment it arrives, without waiting for it to accumulate.

Batch data refers to data that is collected, stored, and processed in bulk at a scheduled time. This method is suitable for scenarios where immediate processing is not required, such as generating monthly sales reports, processing payroll, or analyzing archived logs. Batch processing is simple and efficient for handling large volumes of data but lacks real-time capabilities.

On the other hand, stream data is generated and processed continuously in small chunks, enabling real-time insights and immediate actions. It is used in applications like real-time analytics, fraud detection, and system monitoring. Stream processing handles data as it comes in, which requires more complex infrastructure but offers the advantage of low-latency responses.

2. How does Data Streaming Takes Place Using Apache Spark?

Apache Spark Streaming is a powerful extension of Apache Spark that enables scalable and fault-tolerant stream processing. It works by dividing incoming data into small batches, which are then processed using the Spark engine.

Working of Spark Streaming:

- Data from live sources like Kafka, Flume, HDFS, or socket connections is ingested as mini-batches.
- Each mini-batch is represented internally as an RDD (Resilient Distributed Dataset).
- These RDDs are processed using standard Spark transformations and actions.

- After processing, results can be pushed to external systems such as databases, file systems, or visual dashboards.

Key Components of Spark Streaming:

- Dstreams (Discretized Streams):
  The core abstraction in Spark Streaming. A DStream is a series of RDDs that represent a continuous stream of data.

- Input Sources:
  Real-time data can be ingested from Kafka, socket streams, text files, HDFS, or custom receivers.

- Transformations:
  DStreams support several transformations such as `map()`, `filter()`, `reduceByKey()`, and `window()` operations.

- Windowed Computations:
  Enables processing over a sliding time window, useful for aggregating data over the past few minutes or hours.

Use Cases:

- Real-time Fraud Detection: Analyzing transactions to identify suspicious behavior instantly.

- Social Media Analytics: Monitoring and analyzing user sentiments from platforms like Twitter or Facebook in real time.

- IoT Monitoring: Processing data from sensors and devices continuously.

- System Log Processing: Capturing and analyzing logs from servers or applications to identify failures or unusual behavior.

Conclusion:

Apache Spark enables unified data processing through both batch and stream processing models. While batch processing is best suited for offline, historical data analysis, streaming offers real-time processing for immediate insights. Spark Streaming combines these two paradigms under one framework, offering developers a flexible and robust toolset for building big data applications. Its ability to process massive volumes of data efficiently and with low latency makes it a top choice for industries relying on real-time analytics.