Name: Vishal Rajesh Mahajan AIDS EXP09

Class: TE IT A Roll No: 56

Extended Theory

Different Types of Big Data Analytics Tools

Big Data analytics tools are categorized based on the type of processing they perform:

1. Batch Processing Tools

These tools are designed to handle large volumes of data accumulated over a period. They process data in batches and are best suited for scenarios where immediate or real-time results are not necessary.

- **Apache Hadoop** – An open-source framework used for distributed data storage and batch processing based on the MapReduce programming model.
- **Apache Hive** – A data warehousing solution built on Hadoop that offers SQL-like query support for large datasets.
- **Apache Pig** – A high-level scripting language for developing MapReduce programs in Hadoop, simplifying complex data transformations.
- **Google BigQuery** – A fully-managed, cloud-based data warehouse that allows running SQL queries on massive datasets efficiently.

2. Streaming (Real-Time) Processing Tools

Streaming tools are capable of processing data continuously as it is generated. These tools are ideal for real-time analytics such as fraud detection, system monitoring, and IoT-based applications.

- **Apache Spark Streaming** – A module of Apache Spark that enables real-time processing of streaming data.
- **Apache Kafka** – A distributed platform for building real-time data pipelines and streaming applications.
- **Apache Flink** – A powerful framework for stateful stream processing with low latency and high throughput.
- **Amazon Kinesis** – A cloud-based platform for collecting, processing, and analyzing real-time streaming data.

3. Interactive Query Processing Tools
These tools enable users to perform queries on big data with minimal delay, avoiding the need for time-consuming batch jobs.

- **Presto** – A fast, distributed SQL query engine capable of querying data from various sources including Hadoop and cloud storage.

- **Druid** – A real-time analytics database optimized for low-latency queries on large datasets.

- **Google Data Studio** – A visualization tool for creating interactive and real-time dashboards and reports.

4. Machine Learning and Data Science Tools
These tools support big data analytics through advanced machine learning techniques and models to uncover patterns and predictions.

- **Apache Mahout** – A scalable library designed for implementing machine learning algorithms such as clustering, classification, and recommendations.
- **TensorFlow** – An open-source framework widely used for deep learning and artificial intelligence applications.
- **H2O.ai** – An AI platform offering open-source tools for building and deploying machine learning models at scale.

Apache Spark and Spark Framework

What is Apache Spark?

Apache Spark is a powerful, open-source distributed computing engine designed for big data analytics. Its in-memory processing capabilities provide faster performance compared to traditional batch-processing systems like Hadoop. Spark supports various processing workloads, including batch, streaming, machine learning, and graph processing.

Key Features of Apache Spark:

- **Speed** – Performs computations in memory, allowing for significantly faster processing.
- **Ease of Use** – Offers APIs in popular languages like Python, Java, Scala, and R.
- **Flexibility** – Compatible with different data sources such as HDFS, Cassandra, and Amazon S3.
- **Scalability** – Capable of running on large clusters with thousands of nodes.

Spark Framework Components:

- **Spark Core** – The base engine responsible for essential functions like task scheduling, memory management, and fault tolerance.
- **Spark SQL** – Supports structured data processing using SQL queries on large datasets.
- **Spark Streaming** – Allows for real-time data stream processing.
- **MLlib (Machine Learning Library)** – A library that provides scalable machine learning algorithms.
- **GraphX** – A tool for graph computation, used to analyze and process graph-structured data.

Spark Execution Process:

1. The user submits a job to Spark.
2. Spark generates a Directed Acyclic Graph (DAG) to plan and optimize task execution.
3. The job is broken down into smaller tasks and distributed across the cluster for parallel execution.
4. Results from these tasks are combined and returned to the user.