**Day 9 Notes**

**Name : Podutur Lahari**

**Date : 18-11-2024**

**Big Data**

#### **1. Introduction to Big Data**

* Big Data refers to extremely large datasets that exceed traditional data processing capabilities.
* Companies like Google, Facebook, and LinkedIn were pioneers in utilizing Big Data.
* It offers benefits like cost reduction, faster computation, and improved services.

#### **2. Characteristics of Big Data (3Vs)**

* **Volume**: Vast amounts of data generated (e.g., GB, TB, PB).
* **Velocity**: Speed at which data is generated and processed (e.g., real-time data from sensors).
* **Variety**: Different forms of data such as structured, semi-structured, and unstructured (e.g., text, audio, video).

#### **3. Sources of Big Data**

* Social media, mobile devices, sensors, cameras, software programs, and science facilities.

#### **4. Applications of Big Data**

* Homeland security, healthcare, sales, telecom, manufacturing, traffic control, trading analytics, and search quality.

#### **5. Big Data Storage**

* Techniques include selecting data sources, eliminating redundant data, and using NoSQL databases.
* Tools: Hadoop Distributed File System (HDFS), HBase, Hive.

#### **6. Big Data Processing**

* Involves integrating data, mapping it to programming frameworks, and using Hadoop MapReduce for distributed processing.

#### **7. Hadoop Framework**

* Open-source framework for distributed storage and processing.
* Key components: HDFS and MapReduce.
* History: Originated from Google’s GFS and MapReduce whitepapers and implemented by Yahoo.

#### **8. Why Big Data?**

* Vast and rapidly growing data sources:
  + Facebook: 10 TB of data daily.
  + Twitter: 7 TB of data daily.
* IBM claims 90% of today's data was generated in the last two years.

#### **9. Big Data Tools**

* **Storage**: Distributed systems like Amazon S3.
* **Processing**: Hadoop MapReduce, MongoDB, and schema-free databases.
* **Hosting**: Cloud platforms like Amazon EC2.

#### **10. Benefits of Big Data**

* Enables real-time decision-making and meaningful actions.
* Technologies like MapReduce and Hive allow querying without altering data structures.
* Plays a significant role in the $64 billion database and analytics market.

**Spark SQL**

#### **1. Introduction**

* Spark SQL is a module of Apache Spark for structured data processing.
* Introduced in Spark 1.0 (May 2014), it enables SQL-like querying and data processing through DataFrames.

#### **2. Key Concepts**

* RDD (Resilient Distributed Datasets): Fundamental Spark data structure, fault-tolerant, and distributed.
* DataFrame: Distributed collection of data organized into named columns, similar to relational tables.
* Dataset: A strongly-typed version of DataFrames introduced in Spark 1.6.

#### **3. Architecture**

* Built on Spark Core with schema-based operations using SchemaRDD.
* Supports multiple languages: Python, Scala, Java, HiveQL.
* Data sources: Parquet files, JSON documents, Hive tables, Cassandra, and more.

#### **4. Features**

1. Integration:
   * Mixes SQL queries with Spark programs for combined operations.
   * APIs support multiple languages like Python, Scala, Java, and R.
2. Unified Data Access:
   * Access data from Hive tables, Parquet files, JSON, and external systems.
3. Hive Compatibility:
   * Runs Hive queries and uses the Hive frontend and metastore.
4. Standard Connectivity:
   * JDBC and ODBC support for external connectivity.
5. Scalability:
   * Handles both interactive and long queries with mid-query fault tolerance.

#### **5. Catalyst Optimizer**

* Core engine for query optimization in Spark SQL.
* Performs logical and physical plan optimizations for efficient execution.
* Simplifies transformations using tree manipulation rules.

#### **6. Challenges and Solutions**

* Challenges: ETL for diverse data sources, advanced analytics like ML or graph processing.
* Solutions:
  + DataFrame API for relational operations.
  + Catalyst optimizer for rule-based query planning.

#### **7. Comparison of RDD, DataFrame, and Dataset**

* RDD: Low-level, unstructured data; requires more code for operations.
* DataFrame: High-level abstraction with named columns, optimized for performance.
* Dataset: Adds type-safety and compile-time checks to DataFrames.

#### **8. Optimized Execution**

* Efficient query execution by optimizing logical and physical plans.
* Techniques like filter push-down and projection combination improve performance.

#### **9. Use Cases**

* ETL processing, machine learning workflows, graph analytics, and complex data aggregation.

#### **10. Practical Examples**

* Example operations include filtering, grouping, joining data, and writing queries with APIs.
* Efficient data storage formats like Parquet are used to optimize processing.

#### **11. Integration with Hive**

* Leverages HiveQL and existing Hive data warehouses for compatibility and enhanced querying.

#### **12. Benefits**

* Unified data processing engine.
* Scales seamlessly from kilobytes to petabytes.
* Easily integrates with Big Data tools like HDFS, MySQL, and Elasticsearch.