G. Rupa Manasa 22-12-2023

Day 17 Assessment

**Introduction to Apache Spark:**

Apache Spark is an open-source, distributed computing system that is designed for big data processing and analytics.

It provides an interface for programming entire clusters with implicit data parallelism and fault tolerance.

**Features of Apache Spark:**

Speed

Ease of use

Versatility

Supports multiple languages

Advanced Analytics

**Components:**

Spark Core, Spark SQL, Spark Streaming, MLlib, GraphX, SparkR.

**Spark Core:** Spark Core is the foundational and distributed data processing engine of Apache Spark.

It provides the basic functionality of Spark, including task scheduling, memory management, and fault recovery.

**Spark SQL:** Spark SQL is a module for structured data processing, enabling the integration of SQL queries with Spark programs.

It supports querying data using SQL syntax as well as the DataFrame API, allowing users to work with structured data.

**Spark Streaming:** Spark Streaming is a real-time processing module that allows the processing of live data streams.

It breaks down the data stream into small batches and processes them using Spark's core engine.

**MLlib (Machine Learning Library):** MLlib is a scalable machine learning library for Spark, providing a set of high-level APIs for machine learning algorithms.

It includes tools for data preparation, feature extraction, model training, and evaluation.

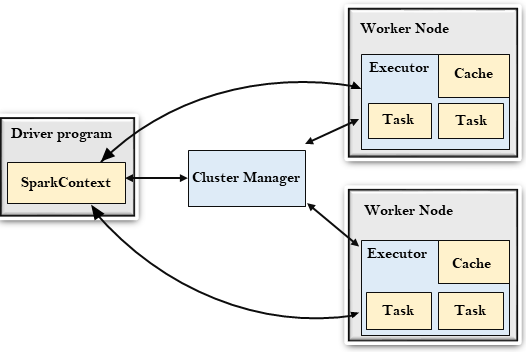
**GraphX:** GraphX is a graph processing library built on top of Spark, allowing for the processing of graph-structured data.

It provides a set of graph computation operations and a distributed graph computation framework.

**SparkR:** SparkR is an R package that provides an R interface for Apache Spark.

It allows R users to leverage the capabilities of Spark, enabling distributed data processing using familiar R syntax.

**Architecture of Apache Spark Cluster:**

****

**Driver Program:** The driver program is the main entry point for Spark applications.

It contains the SparkContext, which is responsible for coordinating the execution of tasks across the cluster.

The driver program is responsible for creating the SparkContext, setting up configuration parameters, and defining the application logic.

**Cluster Manager:** Spark can run on different cluster managers, such as Apache Mesos, Apache Hadoop YARN, or its built-in standalone cluster manager.

The cluster manager is responsible for managing resources (CPU, memory) across the cluster, scheduling tasks, and ensuring fault tolerance.

**Cluster Nodes/Workers:** A Spark cluster consists of multiple worker nodes, also known as executors, distributed across the cluster.

Each worker node is responsible for executing tasks assigned to it by the driver program.

Executors run in separate Java processes and communicate with the driver program for task execution.