Introduction to Apache Spark using Scala

Business Overview

Apache Spark is a distributed processing solution for large data workloads that is open-source. For quick analytic queries against any quantity of data, it uses in-memory caching and efficient query execution. It offers code reuse across many workloads—batch processing, interactive queries, real-time analytics, machine learning, and graph processing—and provides development APIs in Java, Scala, Python, and R.

Hadoop MapReduce is a programming technique that uses a parallel, distributed method to handle extensive data collections. Developers do not have to worry about job distribution or fault tolerance when writing massively parallelized operators. The sequential multi-step procedure required to perform a task, however, is a difficulty for MapReduce. MapReduce gets data from the cluster, conducts operations, and publishes the results to HDFS at the end of each phase. Due to the latency of disk I/O, MapReduce tasks are slower since each step involves a disk read and write. By doing processing in memory, lowering the number of steps in a job, and reusing data across several concurrent processes, Spark was built to solve the constraints of MapReduce. With Spark, data is read into memory in a single step, operations are executed, and the results are written back, resulting in significantly quicker execution. Spark additionally reuses data by employing an in-memory cache to substantially accelerate machine learning algorithms that execute the same function on the same dataset several times.

Tech Stack

➔ Language: Scala, SQL

➔ Services: Apache Spark, IntelliJ

Approach

· Using Docker

    o Implementing RDD Transformation and Action functions

· Using IntelliJ

    o Using IntelliJ to setup SBT for Scala-Spark project

    o Spark Analysis for the given dataset

Dataset Description

Fitness Tracker data is used to perform transformations and gain insights. Few parameters included in this data are:

- Platform
- Activity
- Heartrate
- Calories
- Time_stamp

Key Takeaways

- Understanding project overview
- Installing Spark using Docker
- Introduction to Apache Spark architecture
- Understanding Resilient Distributed Dataset (RDD)
- Understanding RDD Transformations
- Understanding RDD Actions
- Implementing RDD Shuffle Operation
- Understanding dataset and its scope
- Creating Spark Session for Data formatting
- Deriving valuable insights from the data
- Exploring Spark Web UI
- Understanding Spark Configuration properties