Introduction to Big Data with Spark and Hadoop

Module 3 Glossary: Apache Spark

Welcome! This alphabetized glossary contains many of the terms in this course. This comprehensive glossary also includes additional industry-recognized terms not used in course videos. These terms are essential for you to recognize when working in the industry, participating in user groups, and in other professional certificate programs.

Estimated reading time: 10 minutes

Term	Definition
Amazon Simple Storage Service (Amazon S3)	An object store interface protocol that Amazon invented. It is a Hadoop component that understands the S3 protocol. S3 provides an interface for Hadoop services, such as IBM Db2 Big SQL, to consume S3-hosted data.
Apache Spark	An in-memory and open-source application framework for distributed data processing and iterative analysis of enormous data volumes.
Application programming interface (API)	Set of well-defined rules that help applications communicate with each other. It functions as an intermediary layer for processing data transfer between systems, allowing companies to open their application data and functionality to business partners, third-party developers, and other internal departments.
Big data	Data sets whose type or size supersedes the ability of traditional relational databases to manage, capture, and process the data with low latency. Big data characteristics include high volume, velocity, and variety.
Classification algorithms	A type of machine learning algorithm that helps computers learn how to categorize things into different groups based on patterns they find in data.
Cluster management framework	It handles the distributed computing aspects of Spark. It can exist as a stand-alone server, Apache Mesos, or Yet Another Resource Network (YARN). A cluster management framework is essential for scaling big data.
Commodity hardware	Consists of low-cost workstations or desktop computers that are IBM-compatible and run multiple operating systems such as Microsoft Windows, Linux, and DOS without additional adaptations or software.
Compute interface	A shared boundary in computing against which two or more different computer system components exchange information.
Data engineering	A prominent practice that entails designing and building systems for collecting, storing, and analyzing data at scale. It is a discipline with applications in different industries. Data engineers use Spark tools, including the core Spark engine, clusters, executors and their management, Spark SQL, and DataFrames.
Data science	Discipline that combines math and statistics, specialized programming, advanced analytics, artificial intelligence (AI), and machine learning with specific subject matter expertise to unveil actionable insights hidden in the organization's data. These insights can be used in decision-making and strategic planning.
DataFrames	Data collection categorically organized into named columns. DataFrames are conceptually equivalent to a table in a relational database and similar to a dataframe in R or Python, but with greater optimizations. They are built on top of the Spark SQL RDD API. They use RDDs to perform relational queries. Also, they are highly scalable and support many data formats and storage systems. They are developer-friendly, offering integration with most big data tools via Spark and APIs for Python, Java, Scala, and R.
Declarative programming	A programming paradigm that a programmer uses to define the program's accomplishment without defining how it needs to be implemented. The approach primarily focuses on what needs to be achieved, rather than advocating how to achieve it.
Distributed computing	A group of computers or processors working together behind the scenes. It is often used interchangeably with parallel computing. Each processor accesses its own memory.
Fault tolerance	A system is fault-tolerant if it can continue performing despite parts failing. Fault tolerance helps to make your remote-boot infrastructure more robust. In the case of OS deployment servers, the whole system is fault-tolerant if the OS deployment servers back up each other.
For-loop	Extends from a FOR statement to an END FOR statement and executes for a specified number of iterations, defined in the FOR statement.
Functional programming (FP)	A style of programming that follows the mathematical function format. Declarative implies that the emphasis of the code or program is on the "what" of the solution as opposed to the "how to" of the solution. Declarative syntax abstracts out the implementation details and only emphasizes the final output, restating "the what." We use expressions in functional programming, such as the expression f of x, as mentioned earlier.
Hadoop	An open-source software framework offering reliable distributed processing of large data sets by using simplified programming models.
Hadoop Common	Fundamental part of the Apache Hadoop framework. It refers to a collection of primary utilities and libraries that support other Hadoop modules.
Hadoop Distributed File System (HDFS)	A file system distributed on multiple file servers, allowing programmers to access or store files from any network or computer. It is the storage layer of Hadoop. It works by splitting the files into blocks, creating replicas of the blocks, and storing them on different machines. It is built to access streaming data seamlessly. It uses a command-line interface to interact with Hadoop.
HBase	A column-oriented, non-relational database system that runs on top of Hadoop Distributed File System (HDFS). It provides real-time wrangling access to the Hadoop file system. It uses hash tables to store data in indexes, allowing for random data access and making lookups faster.
Immutable	This type of object storage allows users to set indefinite retention on the object if they are unsure of the final duration of the retention period or want to use event-based retention. Once set to indefinite, user applications can change the object retention to a finite value.
Imperative programming paradigm	In this software development paradigm, functions are implicitly coded in every step used in solving a problem. Every operation is coded, specifying how the problem will be solved. This implies that pre-coded models are not called on.

Term	Definition
In-memory processing	The practice of storing and manipulating data directly in a computer's main memory (RAM), allowing for faster and more efficient data operations compared to traditional disk-based storage.
Iterative process	An approach to continuously improving a concept, design, or product. Creators produce a prototype, test it, tweak it, and repeat the cycle to get closer to the solution.
Java	A technology equipped with a programming language and a software platform.
Java virtual machines (JVMs)	The platform-specific component that runs a Java program. At runtime, the VM interprets the Java bytecode compiled by the Java compiler. The VM is a translator between the language and the underlying operating system and hardware.
JavaScript Object Notation (JSON)	A simplified data-interchange format based on a subset of the JavaScript programming language. IBM Integration Bus provides support for a JSON domain. The JSON parser and serializer process messages in the JSON domain.
Lambda calculus	A mathematical concept that implies every computation can be expressed as an anonymous function that is applied to a data set.
Lambda functions	Calculus functions, or operators. These are anonymous functions that enable functional programming. They are used to write functional programming code.
List processing language (Lisp)	The functional programming language that was initially used in the 1950s. Today, there are many functional programming language options, including Scala, Python, R, and Java.
Machine learning	A full-service cloud offering that allows developers and data scientists to collaborate and integrate predictive capabilities with their applications.
MapReduce	A program model and processing technique used in distributed computing based on Java. It splits the data into smaller units and processes big data. It is the first method used to query data stored in HDFS. It allows massive scalability across hundreds or thousands of servers in a Hadoop cluster.
Modular development	Techniques used in job designs to maximize the reuse of parallel jobs and components and save user time.
Parallel computing	A computing architecture in which multiple processors execute different small calculations fragmented from a large, complex problem simultaneously.
Parallel programming	It resembles distributed programming. It is the simultaneous use of multiple compute resources to solve a computational task. Parallel programming parses tasks into discrete parts solved concurrently using multiple processors. The processors access a shared pool of memory, which has control and coordination mechanisms in place.
Parallelization	Parallel regions of program code executed by multiple threads, possibly running on multiple processors. Environment variables determine the number of threads created and calls to library functions.
Persistent cache	Information is stored in "permanent" memory. Therefore, data is not lost after a system crash or restart, as if it were stored in cache memory.
Python	Easy-to-learn, high-level, interpreted, and general-purpose dynamic programming language focusing on code readability. It provides a robust framework for building fast and scalable applications for z/OS, with a rich ecosystem of modules to develop new applications like any other platform.
R	An open-source, optimized programming language for statistical analysis and data visualization. Developed in 1992, it has a rich ecosystem with complex data models and elegant tools for data reporting.
Redundancy	Duplication of data across multiple partitions or nodes in a cluster. This duplication is implemented to enhance fault tolerance and reliability. If one partition or node fails, the duplicated data on other partitions or nodes can still be used to ensure that the computation continues without interruption. Redundancy is critical in maintaining data availability and preventing data loss in distributed computing environments like Spark clusters.
Resilient Distributed Datasets (RDDs)	A fundamental abstraction in Apache Spark that represents distributed collections of data. RDDs allow you to perform parallel and fault-tolerant data processing across a cluster of computers. RDDs can be created from existing data in storage systems (like HDFS), and they can undergo various transformations and actions to perform operations like filtering, mapping, and aggregating. The "resilient" aspect refers to resilient distributed datasets (RDDs) ability to recover from node failures, and the "distributed" aspect highlights their distribution across multiple machines in a cluster, enabling parallel processing.
Scala	A general-purpose programming language that supports both object-oriented and functional programming. The most recent representative in the family of programming languages. Apache Spark is written mainly in Scala, which treats functions as first-class citizens. Functions in Scala can be passed as arguments to other functions, returned by other functions, and used as variables.
Scalability	The ability of a system to take advantage of additional resources, such as database servers, processors, memory, or disk space. It aims at minimizing the impact on maintenance. It is the ability to maintain all servers efficiently and quickly with minimal impact on user applications.
Spark applications	Include a driver program and executors that run the user's multiple primary functions and different parallel operations in a cluster.
Spark Core	Often popularly referred to as "Spark." The fault-tolerant Spark Core is the base engine for large-scale parallel and distributed data processing. It manages memory and task scheduling. It also contains the APIs used to define RDDs and other datatypes. It parallelizes a distributed collection of elements across the cluster.
Spark ML	Spark's machine learning library for creating and using machine learning models on large data sets across distributed clusters.
Spark SQL	A Spark module for structured data processing. Users can interact with Spark SQL using SQL queries and the DataFrame API. Spark SQL supports Java, Scala, Python, and R APIs. Spark SQL uses the same execution engine to compute the result independently of the API or language used for computation. Developers can use the API to help express a given transformation. Unlike the basic Spark RDD API, Spark SQL includes a cost-based optimizer, columnar storage, and code generation to perform optimizations that equip Spark with information about the structure of data and the computation in process.

Term	Definition
SQL Procedural code	A set of instructions written in a programming language within an SQL database environment. This code allows users to perform more complex tasks and create custom functions, procedures, and control structures, enabling them to manipulate and manage data in a more controlled and structured manner.
Streaming analytics	Help leverage streams to ingest, analyze, monitor, and correlate data from real-time data sources. They also help to view information and events as they unfold.
Worker node	A unit in a distributed system that performs tasks and processes data according to instructions from a central coordinator.

Author(s)

- Niha Ayaz Sultan