Introduction to Big Data with Spark and Hadoop

Module 5 Glossary: Development and Runtime Environment Options

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: 15 minutes

Term	Definition
AIOps	Implies the use of artificial intelligence to automate or enhance IT operations. It helps collect, aggregate, and work with large volumes of operations data. It also helps to identify events and patterns in large or complex infrastructure systems. It allows quick diagnosis of the root cause of issues so that users can report or fix them automatically.
Apache Hadoop YARN (Yet Another Resource Negotiator)	Cluster manager from the Hadoop project. It's popular in the big data ecosystem and supports many other frameworks besides Spark. YARN clusters have their own dependencies, setup, and configuration requirements, so deploying them is more complex than Spark standalone.
Apache Mesos	General-purpose cluster manager with additional benefits. Using Mesos has some advantages. It can provide dynamic partitioning between Spark and other big data frameworks and scalable partitioning between multiple Spark instances. However, running Spark on Apache Mesos might require additional setup, depending on your configuration requirements.
Apache Spark Architecture	Consists of the driver and the executor processes. The cluster comprises the cluster manager and worker nodes. The Spark Context schedules tasks for the cluster, and the cluster manager manages the cluster's resources.
Apache Spark Cluster Modes	Apache Spark offers various cluster modes for distributed computing, including standalone, YARN (Yet Another Resource Negotiator), and Apache Mesos. Each mode has specific characteristics and setup complexities.
Apache Spark	An open-source, distributed computing framework designed for processing large-scale data and performing data analytics. It provides libraries for various data processing tasks, including batch processing, real-time stream processing, machine learning, and graph processing. Spark is known for its speed and ease of use, making it a popular choice for big data applications.
Bootstrap data set (BSDS)	A Virtual Storage Access Method (VSAM) key-sequenced data set (KSDS) used to store the essential information required by IBM MQ (message queuing). The BSDS typically includes an inventory of all active and archived log data sets known to IBM MQ. IBM MQ uses this inventory to track active and archived log data sets. The BSDS plays a critical role in the proper functioning and management of IBM MQ, ensuring the integrity and availability of log data sets within the messaging system.
Clusters	Refer to groups of servers that are managed collectively and participate in workload management. You can have nodes within a cluster, typically individual physical computer systems with distinct host IP addresses. Each node in a cluster can run one or more application servers. Clusters are a fundamental concept in distributed computing and

Term	Definition
	server management, allowing for the efficient allocation of resources and the scalability of applications and services across multiple server instances.
Containerization	Implies Spark applications are more portable. It makes it easier to manage dependencies and set up the required environment throughout the cluster. It also supports better resource sharing.
Driver program	It can be run in either client or cluster mode. In client mode, the application submitter (such as a user machine terminal) launches the driver outside the cluster. In cluster mode, the driver program is sent to and run on an available worker node inside the cluster. The driver must be able to communicate with the cluster while it is running, whether it is in client or cluster mode.
Dynamic configuration	Refers to a practice employed in software development to avoid hardcoding specific values directly into the application's source code. Instead, critical configuration settings, such as the location of a master server, are stored externally and are adjustable without modifying the application's code.
Environment variables	Spark application configuration method in which environment variables are loaded on each machine, so they can be adjusted on a per-machine basis if hardware or installed software differs between the cluster nodes. A common usage is to ensure each machine in the cluster uses the same Python executable by setting the "PYSPARK_PYTHON" environment variable.
Executor	Utilizes a set portion of local resources as memory and compute cores, running one task per available core. Each executor manages its data caching as dictated by the driver. In general, increasing executors and available cores increases the cluster's parallelism. Tasks run in separate threads until all cores are used. When a task finishes, the executor puts the results in a new RDD partition or transfers them back to the driver. Ideally, limit utilized cores to the total cores available per node.
Hybrid cloud	Unifies and combines public and private cloud and on-premises infrastructure to create a single, cost-optimal, and flexible IT infrastructure.
IBM Analytics Engine	Works with Spark to provide a flexible, scalable analytics solution. It uses an Apache Hadoop cluster framework to separate storage and compute by storing data in object storage such as IBM Cloud Object Storage. This implies users can run compute nodes only when required.
IBM Spectrum Conductor	A multitenant platform for deploying and managing Spark and other frameworks on a cluster with shared resources. This enables multiple Spark applications and versions to be run together on a single large cluster. Cluster resources can be divided up dynamically, avoiding downtime. IBM Spectrum Conductor also provides Spark with enterprise-grade security.
IBM Watson	Creates production-ready environments for AI and machine learning by providing services, support, and holistic workflows. Reducing setup and maintenance saves time so that users can concentrate on training Spark to enhance its machine-learning capabilities. IBM Cloud Pak for Watson AIOps offers solutions with Spark that can correlate data across your operations toolchain to bring insights or identify issues in real time.
Java Archive (JAR)	A standard file format used to package Java classes and related resources into a single compressed file. JAR files are commonly used to bundle Java libraries, classes, and other assets into a single unit for distribution and deployment.
Java	Technology equipped with a programming language and a software platform. To create and develop an application using Java, users are required to download the Java Development Kit (JDK), available for Windows, macOS, and Linux.
Kubernetes (K8s)	A popular framework for running containerized applications on a cluster. It is an open-source system that is highly scalable and provides flexible deployments to the cluster.

Term	Definition
	Spark uses a built-in native Kubernetes scheduler. It is portable, so it can be run in the same way on cloud or on-premises.
Local mode	Runs a Spark application as a single process locally on the machine. Executors are run as separate threads in the main process that calls "spark-submit". Local mode does not connect to any cluster or require configuration outside a basic Spark installation. Local mode can be run on a laptop. That's useful for testing or debugging a Spark application, for example, testing a small data subset to verify correctness before running the application on a cluster. However, being constrained by a single process means local mode is not designed for optimal performance.
Logging configuration	Spark application configuration method in which Spark logging is controlled by the log4j defaults file, which dictates what level of messages, such as info or errors, are logged to the file or output to the driver during application execution.
Properties	Spark application configuration method in which Spark properties are used to adjust and control most application behaviors, including setting properties with the driver and sharing them with the cluster.
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 the same way you would on any other platform.
Scala	General-purpose programming language that supports functional and object-oriented 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.
Spark application	A program or set of computations written using the Apache Spark framework. It consists of a driver program and a set of worker nodes that process data in parallel. Spark applications are designed for distributed data processing, making them suitable for big data analytics and machine learning tasks.
Spark Cluster Manager	Communicates with a cluster to acquire resources for an application to run. It runs as a service outside the application and abstracts the cluster type. While an application is running, the Spark Context creates tasks and communicates to the cluster manager what resources are needed. Then the cluster manager reserves executor cores and memory resources. Once the resources are reserved, tasks can be transferred to the executor processes to run.
Spark Configuration Location	Located under the "conf" directory in the installation. By default, there are no preexisting files after installation; however, Spark provides a template for each configuration type with the filenames shown here. Users can create the appropriate file by removing the '.template' extension. Inside the template files are sample configurations for standard settings. They can be enabled by uncommenting.
Spark Context	Communicates with the Cluster Manager. It is defined in the Driver, with one Spark Context per Spark application.
Spark jobs	Computations that can be executed in parallel. The Spark Context divides jobs into tasks to be executed on the cluster.
Spark logging	Controlled using log4j and the configuration is read through "conf/log4j-properties". Users can adjust a log level to determine which messages (such as debug, info, or errors) are shown in the Spark logs.
Spark Shell environment	When Spark Shell starts, the environment automatically initializes the SparkContext and SparkSession variables. This means you can start working with data immediately. Expressions are entered in the Shell and evaluated in the driver. Entering an action on a Shell DataFrame generates Spark jobs that are sent to the cluster to be scheduled as tasks.

Term	Definition
Spark Shell	Available for Scala and Python, giving you access to Spark APIs for working with data as Spark jobs. Spark Shell can be used in local or cluster mode, with all options available.
Spark Shuffle	Performed when a task requires other data partitions. It marks the boundary between stages.
Spark stages	Represents a set of tasks an executor can complete on the current data partition. Subsequent tasks in later stages must wait for that stage to be completed before beginning execution, creating a dependency from one stage to the next.
Spark standalone cluster	Has two main components: Workers and the master. The workers run on cluster nodes. They start an executor process with one or more reserved cores. There must be one master available which can run on any cluster node. It connects workers to the cluster and keeps track of them with heartbeat polling. However, if the master is together with a worker, do not reserve all the node's cores and memory for the worker.
Spark standalone	Included with the Spark installation. It is best for setting up a simple cluster. There are no additional dependencies required to configure and deploy. Spark standalone is specifically designed to run Spark and is often the fastest way to get a cluster up and running applications.
Spark tasks	Tasks from a given job operate on different data subsets called partitions and can be executed in parallel.
Spark-submit	Spark comes with a unified interface for submitting applications called the "spark-submit" script found in the "bin/" directory. "Spark-submit" can be used for all supported cluster types and accepts many configuration options for the application or cluster. Unified interface means you can switch from running Spark in local mode to cluster by changing a single argument. "Spark-submit" works the same way, irrespective of the application language. For example, a cluster can run Python and Java applications simultaneously by passing in the required files.
Static configuration	Settings that are written programmatically into the application. These settings are not usually changed because they require modifying the application itself. Use static configuration for something that is unlikely to be changed or tweaked between application runs, such as the application name or other properties related to the application only.
Uber-JAR	An Uber-JAR is a single Java Archive (JAR) file that contains not only the application code but also all its dependencies, including transitive ones. The purpose of an Uber-JAR is to create a self-contained package that can be easily transported and executed within a computing cluster or environment.
Worker	Cluster node that can launch executor processes to run tasks.

Author(s)

- Niha Ayaz Sultan

Changelog

Date	Version	Changed by	Change Description
2023-09-05	0.1	Sameeksha Saxena	Initial version created
2023-09-08	0.2	Pornima More	QA pass with edits