Apache Scala and Spark Development

By Dr. Vishwanath Rao

Objectives

- Apache Spark and Scala programming
- Difference between Apache Spark and Hadoop
- Scala and its programming implementation
- Implementing Spark on a cluster
- Writing Spark applications using Python, Java and Scala
- RDD and its operation, along with the implementation of Spark algorithms
- Defining and explaining Spark streaming
- Scala classes concept and executing pattern matching
- Scala–Java interoperability and other Scala operations
- Working on projects using Scala to run on Spark applications

Prerequisites

Knowledge of Java or any related language.

Lab setup

Windows 10 or equivalent OS
16GB RAM
Scala IDE
Jdk 8 or 11
Scala SDK
Other related softwares will be installed under Instructor’s guidance
Open Internet without any download rights

Course Contents

Introducing Scala

Deployment of Scala for Big Data applications and Apache Spark analytics
Scala REPL, lazy values, and control structures in Scala
Directed Acyclic Graph (DAG)
First Spark application using SBT/Eclipse
Spark Web UI
Spark in the Hadoop ecosystem.

The importance of Scala
The concept of REPL (Read Evaluate Print Loop)
Deep dive into Scala pattern matching
Type interface, higher-order function, currying, traits, application space and Scala for data analysis

Learning about the Scala Interpreter
Static object timer in Scala and testing string equality in Scala
Implicit classes in Scala
The concept of currying in Scala
Various classes in Scala

Learning about the Classes concept
Understanding the constructor overloading
Various abstract classes
The hierarchy types in Scala
The concept of object equality
The val and var methods in Scala

Understanding sealed traits, wild, constructor, tuple, variable pattern, and constant pattern

Understanding traits in Scala
The advantages of traits
Linearization of traits
The Java equivalent
Avoiding of boilerplate code

Implementation of traits in Scala and Java
Handling of multiple traits extending

Introduction to Scala collections
Classification of collections
The difference between iterator and iterable in Scala
Example of list sequence in Scala

The two types of collections in Scala
Mutable and immutable collections
Understanding lists and arrays in Scala
The list buffer and array buffer
Queue in Scala
Double-ended queue Deque, Stacks, Sets, Maps, and Tuples in Scala

Introduction to Scala packages and imports
The selective imports
The Scala test classes
Introduction to JUnit test class
JUnit interface via JUnit 3 suite for Scala test
Packaging of Scala applications in the directory structure
Examples of Spark Split and Spark Scala

Introduction to Spark
Spark overcomes the drawbacks of working on MapReduce
Understanding in-memory MapReduce
Interactive operations on MapReduce
Spark stack, fine vs. coarse-grained update, Spark stack, Spark Hadoop YARN, HDFS Revision, and YARN Revision
The overview of Spark and how it is better than Hadoop
Deploying Spark without Hadoop
Spark history server and Cloudera distribution

Spark installation guide
Spark configuration
Memory management
Executor memory vs. driver memory
Working with Spark Shell
The concept of resilient distributed datasets (RDD)
Learning to do functional programming in Spark
The architecture of Spark

Spark RDD
Creating RDDs

RDD partitioning
Operations and transformation in RDD
Deep dive into Spark RDDs
The RDD general operations
Read-only partitioned collection of records
Using the concept of RDD for faster and efficient data processing
RDD action for the collect, count, collects map, save-as-text-files, and pair RDD functions

Understanding the concept of key-value pair in RDDs
Learning how Spark makes MapReduce operations faster
Various operations of RDD
MapReduce interactive operations
Fine and coarse-grained update
Spark stack

Comparing the Spark applications with Spark Shell
Creating a Spark application using Scala or Java
Deploying a Spark application
Scala built application
Creation of the mutable list, set and set operations, list, tuple, and concatenating list
Creating an application using SBT
Deploying an application using Maven
The web user interface of Spark application
A real-world example of Spark
Configuring of Spark

Learning about Spark parallel processing
Deploying on a cluster
Introduction to Spark partitions
File-based partitioning of RDDs
Understanding of HDFS and data locality
Mastering the technique of parallel operations
Comparing repartition and coalesce
RDD actions

The execution flow in Spark
Understanding the RDD persistence overview

Spark execution flow, and Spark terminology
Distribution shared memory vs. RDD
RDD limitations
Spark shell arguments
Distributed persistence
RDD lineage
Key-value pair for sorting implicit conversions like CountByKey, ReduceByKey, SortByKey, and AggregateByKey

Introduction to Machine Learning
Types of Machine Learning
Introduction to MLlib
Various ML algorithms supported by MLlib
Linear regression, logistic regression, decision tree, random forest, and K-means clustering techniques

Why Kafka and what is Kafka?
Kafka architecture
Kafka workflow
Configuring Kafka cluster
Operations
Kafka monitoring tools
Integrating Apache Flume and Apache Kafka

1. Configuring Single Node Single Broker Cluster
2. Configuring Single Node Multi Broker Cluster
3. Producing and consuming messages
4. Integrating Apache Flume and Apache Kafka

Introduction to Spark Streaming
Features of Spark Streaming
Spark Streaming workflow
Initializing StreamingContext, discretized Streams (DStreams), input DStreams and Receivers
Transformations on DStreams, output operations on DStreams, windowed operators and why it is useful
Important windowed operators and stateful operators

Introduction to various variables in Spark like shared variables and broadcast variables
Learning about accumulators
The common performance issues
Troubleshooting the performance problems

Learning about Spark SQL
The context of SQL in Spark for providing structured data processing
JSON support in Spark SQL
Working with XML data
Parquet files
Creating Hive context
Writing data frame to Hive
Reading JDBC files
Understanding the data frames in Spark
Creating Data Frames
Manual inferring of schema
Working with CSV files
Reading JDBC tables
Data frame to JDBC
User-defined functions in Spark SQL
Shared variables and accumulators
Learning to query and transform data in data frames
Data frame provides the benefit of both Spark RDD and Spark SQL
Deploying Hive on Spark as the execution engine

Learning about the scheduling and partitioning in Spark
Hash partition
Range partition
Scheduling within and around applications
Static partitioning, dynamic sharing, and fair scheduling
Map partition with index, the Zip, and GroupByKey
Spark master high availability, standby masters with ZooKeeper, single-node recovery with the local file system and high order functions