TRAINING OFFERING

APACHE SPARK 2 FOR DATA ENGINEERS

SUBJECT MATTER EXPERT

This course introduces the Apache Spark distributed computing engine, and is suitable for developers, data analysts, architects, technical managers, and anyone who needs to use Spark in a hands-on manner. It is based on the Spark 2.x release. The course provides a solid technical introduction to the Spark architecture and how Spark works. It covers the basic building blocks of Spark (e.g. RDDs and the distributed compute engine), as well as higher-level constructs that provide a simpler and more capable interface. It includes in-depth coverage of Spark SQL, DataFrames, and DataSets, which are now the preferred programming API. This includes exploring possible performance issues and strategies for optimization. The course also covers more advanced capabilities such as the use of Spark Streaming to process streaming data, and integrating with the Kafka server.

PREREQUISITES

Students should be familiar with programming principles and have previous experience in software development using Scala. Previous experience with data streaming, SQL, and HDP is also helpful, but not required.

TARGET AUDIENCE

Software engineers that are looking to develop in-memory applications for time sensitive and highly iterative applications in an Enterprise HDP environment.

FORMAT

50% Lecture/Discussion
50% Hands-On Labs

COURSE OBJECTIVES

• Scala Introduction
• Working with: Variables, Data Types Control Flow
• The Scala Interpreter
• Collections and their Standard Methods (e.g. map())
• Working with: Functions, Methods, Function Literals
• Define the Following as they Relate to Scale: Class, Object, Case Class
• Overview, Motivations, Spark Systems
• Spark Ecosystem
• Spark vs. Hadoop
• Acquiring and Installing Spark
The Spark Shell, SparkContext
RDD Concepts, Lifecycle, Lazy Evaluation
RDD Partitioning and Transformations
Working with RDDs Including: Creating and Transforming (map, filter, etc.)
An Overview of RDDs
SparkSession, Loading/Saving Data, Data Formats (JSON, CSV, Parquet, text ...)
Introducing DataFrames and DataSets (Creation and Schema Inference)
Identify Supported Data Formats
Working with the DataFrame (untyped) Query DSL
SQL-based Queries
Working with the DataSet (typed) API
Mapping and Splitting (flatMap(), explode(), and split())
DataSets vs. DataFrames vs. RDDs

COURSE OBJECTIVES CONTINUED
Working with: Grouping, Reducing and Joining
Shuffling, Narrow vs. Wide Dependencies, and Performance Implications
Exploring the Catalyst Query Optimizer (explain(), Query Plans, Issues with lambdas)
The Tungsten Optimizer (Binary Format, Cache Awareness, Whole-Stage Code Gen)
Discuss Caching
Minimizing Shuffling for Increased Performance
Using Broadcast Variables and Accumulators
General Performance Guidelines
Core API, SparkSession.Builder
Configuring and Creating a SparkSession
Building and Running Applications - sbt/build.sbt and spark-submit
Application Lifecycle (Driver, Executors, and Tasks)
Cluster Managers (Standalone, YARN, Mesos)
Logging and Debugging
Introduction and Streaming Basics
Spark Streaming (Spark 1.0+)
  o DStreams, Receivers, Batching
  o Stateless Transformation
  o Windowed Transformation
  o Stateful Transformation
Structured Streaming (Spark 2+)
  Continuous Applications
Continuous Applications
- Table Paradigm, Result Table
- Steps for Structured Streaming
- Sources and Sinks

- Consuming Kafka Data
  - Kafka Overview
  - Structured Streaming - "kafka" Format
  - Processing the Stream

HANDS-ON LABS
- Setting Up the Lab Environment
- Starting the Scala Interpreter
- A First Look at Spark
- A First Look at the Spark Shell
- RDD Basics
  - Operations on Multiple RDDs
- Data Formats
- Spark SQL Basics
- DataFrame Transformations
- The DataSet Typed API
- Splitting Up Data
- Exploring Group Shuffling
- Seeing Catalyst at Work
- Seeing Tungsten at Work
- Working with Caching, Joins, Shuffles, Broadcasts, Accumulators
- Broadcast General Guidelines
- Spark Job Submission
- Additional Spark Capabilities
- Spark Streaming
- Spark Structured Streaming
- Spark Structured Streaming with Kafka