Course Outline

Apache Spark for Developers with Advanced Topics

Course Summary

Description

This course will introduce Apache Spark. The students will learn how Spark fits into the Big Data ecosystem, and how to use Spark for data analysis. This class is taught with either Python language or Scala language. A language primer can be offered if needed.

Objectives

After taking this course, students will learn:

- Spark ecosystem
- Spark Shell
- Spark Data structures (RDD, DataFrame, Dataset)
- Spark SQL
- Modern data formats and Spark
- Spark API
- Spark & Hadoop & Hive
- Spark ML overview
- GraphX
- Spark Streaming

Topics

- Spark Introduction
- The first look at Spark
- Spark Data structures
- Caching
- DataFrames and Datasets
- Spark SQL
- Spark and Hadoop
- Spark API
- Spark ML Overview
- GraphX
- Spark Streaming
- Workshops

Audience

This course is designed for Developers and Architects

Prerequisite

Developer background

Duration

Three Days

Due to the nature of this material, this document refers to numerous hardware and software products by their trade names. References to other companies and their products are for informational purposes only, and all trademarks are the properties of their respective companies. It is not the intent of ProTech Professional Technical Services, Inc. to use any of these names generically.

Apache Spark for Developers with Advanced Topics

Course Outline

I. ***Spark Introduction (2h)***
- A. Big data, Hadoop, Spark
- B. Spark concepts and architecture
- C. Spark components overview
- D. Labs: installing and running Spark

II. ***The first look at Spark (2h)***
- A. Spark shell
- B. Spark web UIs
- C. Analyzing dataset – part 1
- D. Labs: Spark shell exploration

III. ***Spark Data structures (2h)***
- A. Partitions
- B. Distributed execution
- C. Operations: transformations and actions
- D. Labs: Unstructured data analytics using RDDs

IV. ***Caching (1.5h)***
- A. Caching overview
- B. Various caching mechanisms available in Spark
- C. In memory file systems
- D. Caching use cases and best practices
- E. Labs: Benchmark of caching performance

V. ***DataFrames and Datasets (5h)***
- A. DataFrames Intro
- B. Loading structured data (JSON, CSV) using DataFrames
- C. Using schema
- D. Specifying schema for DataFrames
- E. Labs: DataFrames, Datasets, Schema

VI. ***Spark SQL (2h)***
- A. Spark SQL concepts and overview
- B. Defining tables and importing datasets
- C. Querying data using SQL
- D. Handling various storage formats: JSON, Parquet, ORC
- E. Labs: querying structured data using SQL; evaluating data formats

VII. ***Spark and Hadoop (2h)***
- A. Hadoop Primer: HDFS, YARN
- B. Hadoop + Spark architecture
- C. Running Spark on Hadoop YARN
- D. Processing HDFS files using Spark
- E. Spark & Hive

VIII. ***Spark API (2h)***
- A. Overview of Spark APIs in Scala / Python
- B. The lifecycle of a Spark application
- C. Spark APIs
- D. Deploying Spark applications on YARN
- E. Labs: Developing and deploying a Spark application

IX. ***Spark ML Overview (3h)***
- A. Machine Learning primer
- B. Machine Learning in Spark: MLib / ML
- C. Spark ML overview (newer Spark2 version)
- D. Algorithms overview: Clustering, Classifications, Recommendations
- E. Machine Learning and Data Processing Pipelines
- F. Labs: Writing ML applications in Spark

X. ***GraphX (1.5)***
- A. GraphX library overview
- B. GraphX APIs
- C. Create a Graph and navigating it
- D. Shortest distance
- E. Pregel API
- F. Labs: Processing graph data using Spark

Due to the nature of this material, this document refers to numerous hardware and software products by their trade names. References to other companies and their products are for informational purposes only, and all trademarks are the properties of their respective companies. It is not the intent of ProTech Professional Technical Services, Inc. to use any of these names generically.

Apache Spark for Developers with Advanced Topics

Course Outline (cont.)

XI. ***Spark Streaming (3h)***

- A. Streaming concepts
- B. Evaluating Streaming platforms
- C. Spark streaming library overview
- D. Streaming operations
- E. Sliding window operations
- F. Structured Streaming
- G. Continuous streaming
- H. Spark & Kafka streaming
- I. Labs: Writing spark streaming applications

XII. ***Workshops***

- A. These are team workshops
- B. Attendees will work on solving real-world data analysis problems using Spark

XIII. Advanced Topics

- A. Spark Memory Usage (3h)
  1. How much memory your application is using (from Spark UI and Spark logs)
  2. What is Tungsten? How it improves memory use for DataFrames and Datasets
  3. Why it’s important that DataFrames never be partially cached, even if it means spilling the cache to disk
  4. What is co-locating data and what are its benefits?
  5. Tuning JVM garbage collection for Spark
- B. Broadcast Variables (2h)
  1. How broadcast variables can affect performance
  2. Why broadcast joins are useful
  3. How to force Spark to do a broadcast join
  4. When not to force a broadcast join
- C. Catalyst (2)
  1. Avoiding Catalyst anti-patterns, such as Cartesian products and partially cached DataFrames
  2. How to read Catalyst query plan (use of ‘explain’)
  3. Efficient use of the Datasets API within a query plan
  4. How encoders and decoders affect Catalyst optimizations
  5. How and when to write a custom Catalyst optimizer
  6. Tuning shuffling
  7. When does shuffling occur?
  8. Understanding how shuffling influences repartitioning
  9. Understanding shuffling impact on the network I/O
  10. Narrow vs. wide transformations
  11. Spark configuration settings that affect shuffling
- D. Cluster Sizing (1h)
  1. How a lack of memory affects how you should size your disks
  2. The importance of properly defined schemas on memory use
  3. Hardware provisioning
  4. How to decide how much memory to allocate to each machine
  5. Network considerations
  6. How to decide how many CPU cores each machine will need
  7. FIFO scheduler vs. fair scheduler
- E. Apache Spark in production (2h)
  1. How to debug a Spark job
  2. Run a Spark job in IntelliJ, using breakpoints, analyzing values
  3. Job accepted but does not run, what to do?
  4. Dynamic allocation enabled
  5. The number of executors, how to set it up?

Apache Spark for Developers with Advanced Topics

Course Outline (cont.)

F. Spark architecture (2h)
   1. Spark on Kubernetes, on Mesos
   2. How code goes to the data
   3. Block manager
   4. Shuffle work
   5. Zeppelin
   6. From running locally in IDE, testing, deploying
   7. Integration with ML, TensorFlow, PyTorch, etc.