**Apache Spark**

**(Basic and Advanced)**

Chinnasamy

[chinnasamyad@gmail.com](mailto:chinnasamyad@gmail.com)

**Prerequisite:**

Candidate attending the training should have a basic knowledge on Java or scala.

**Big Data Conceptuals**

What is Big Data?

The need for Big Data

Why Big Data now?

Myths of Big Data

Tabular representation of data unit measurement.

Is one petabyte big data ?

Types of Architectures in Big Data

Lambda Architecture

Kappa Architecture

Zeta Architecture

Seda Architecture

NoSQL Store and high throughput messaging system

Illustration about CAP theorem

Problems with large-scale systems

**HDFS**

Why HDFS ?

HDFS Architecture

Using HDFS and hdfs commands

**Spark (The Spark version covered is the latest version of Spark - 1.6)**

**JDK 8 - Quick Introduction**

Functional Programming with Java

Lambda expressions and Functional Interfaces in Java

**Scala - Introduction**

Objects and Classes

val, var, functions, currying, implicits

traits, actors and file manipulations

**Core Spark**

Introduction to Apache Spark

What is Spark ? Explain about the modules in spark

Spark-Shell - scala and python REPL

Spark Internals - The Driver program, master, workers, executors and the tasks

SparkSession- The Umbrella API for all context

Running spark in a standalone mode

Spark UI and monitoring a job

Functional programming with Spark

Map-reduce and Spark advantages over Map-reduce.

**RDD**

What is an RDD ?

Laziness in RDD Evaluation

Different ways of creating an RDD

Types of RDD’s - PairRDD, DoubleRDD

RDD Operations

Partitions - The core of RDD

textFiles, wholeFiles

**Running Spark on a Cluster**

Overview

A Spark Standalone Cluster

The Spark Standalone Web UI

Installing and configuring a cluster

**Operations in Spark**

Spark Configuration and the Spark Context

Configuring spark properties

RDD Operations - Transformation and Actions

map, flatMap, repartition, coalesce, glom, reduce, cartesian, pipe, sample,

distinct, mapPartitions, mapPartitionsWithIndex

Map, filter, distinct, collect, take operations

Joining two RDD’s

Storage levels supported in spark

Programming with a partition and use of custom partitioners

Accumulators and Broadcast variables

Checkpointing an RDD

Spark deployment plans

Spark History Server

**Reading Data from External Sources**

JdbcRdd - Read data from mysql

Connecting and reading data from mongodb

**Caching and Persistence**

RDD Lineage

Caching Overview

Distributed Persistence

**SparkSQL**

The DataFrame Abstraction

Elucidate on SparkSQL

Dataframe manipulation on top of json

The temp table abstraction on top of DataFrame Schema

SQL manipulation on top of parquest files

Dataframes caching

Connecting dataframes to relational database

**Spark Streaming**

Kafka and the need

Basic read from a socket

Spark Streaming from kafka

Windowing operation in streaming

Developing streaming applications

Writing a custom receiver

Spark Structured Streaming

**Advanced Topics**

Spark SQL with Hive

The new Dataset API

Working with nested data

Spark with Alluxio

Custom Accumulators

Writing custom RDD

Writing custom partitioner

Internals of persistence API. How spark manages persistence internally.

(Drilling down the source code)

**Spark Performance Tuning**

Various strategies to adopt to performance tune your spark application.

Introduction to various variables in Spark like shared variables.

Broadcast variables and learning about accumulators.

Common performance issues and troubleshooting the performance problems.

Maven would be used as the build tool to download the dependencies. IntelliJ would be the IDE to develop the applications and examples.

**Notable Corporate Trainings**

|  |  |
| --- | --- |
| **Company Name** | **Trainings** |
| Wipro | Apache Spark, Scala, ELK |
| ITC Infotech | Spark, Scala, Kafka, SBT |
| IBM (Hyderabad, Bangalore) | Apache Spark, Scala |
| HP | Scala |
| Scholastic | Apache Spark |
| HCL | Apache Spark, Hadoop, ELK Stack |
| Big Data Analytics Private Ltd | Hadoop, Spark |

Project: A live project of how each of the API’s are used in the industry.

Use cases covered:

[1] A csv file format of three hundred columns will be used as a dataset.

[2] Consuming and operating two csv files (each of 3 MB) that are produced every second

through spark streaming.

[3] Ten to fifteen transformation on a single job. Efficiently optimize and fine tune on all the

transformations.

[4] Architectural sharing of data between spark jobs.

Hands-on/Lecture Ratio:

The course is 60 % hands-on, 40 % discussion, with the longest discussion segments lasting 20 minutes.

Note to participants:

[\*] All content in this course will be a hands-on session.

[\*] All slides of the course will be given to candidates.

[\*] Source code of all examples tried out in the session will be provided.

Training Developers Environment:

[\*] The training programs would be given as a intellij project.

So I would need in a internet connection for the maven execution. The maven would

download a lot of jars from the internet.

[\*] Download sbt and install it.

http://www.scala-sbt.org/download.html

[\*] Download IntelliJ and install it. <https://www.jetbrains.com/idea/>

[\*] Eclipse Mars, JDK 8, Spark 1.6.0 or Spark 2.0 installation on their respective OS.

[\*] Any linux or unix flavor box is needed for the trainees to do the cluster setup of spark.

The box should have an internet connection, JDK 8 and Spark 2.0 or

Spark 1.6 installed in it.

[\*] If the box is not available, then the VM of Ubuntu Linux is needed. In order to run Linux

VM, Oracle Virtualbox or VMWare is needed.

[\*] JDK\_HOME and PATH variable to the JDK 1.8 should be set.

[\*] The trainer has a MAC laptop, so infrastructure should be provided to

connect MAC laptop to the screen.